ZL UA REST API Reference Guide

ZL UA Version 11.1
Last Updated: February 13, 2026
Document Revision 1.03

Contents

Introduction to REST API

The REST API provides integration of the ZL Unified Archive (ZL UA) High Performance API Kit for compatibility with REST architecture and Web services. This provides several advantages:

- For cloud-based Web services, REST is a standard today.
- Easy adaptation of REST APIs into any programming language.
- Simpler implementation and testing of tools built with REST-based APIs.
- Standardizes ZL UA's API services.

You can call ZL UA's REST API in several ways, and you can create your own client or use existing tools like Postman and Swagger. Examples are provided in the following sections.

Authorization in the REST API

Before making a REST API call, you need to authenticate the application using ZL bearer tokens. You need to send the bearer token in the headers along with the requested URL to the REST API.

To get the ZL bearer token, you need to call the **getToken** API. The **getToken** API requires the following inputs:

1. **certificateBody**: ZL server issued certificate file body.
2. **password**: The password used to encrypt the certificate.
3. **tenantId**: Tenant ID of the ZL server.

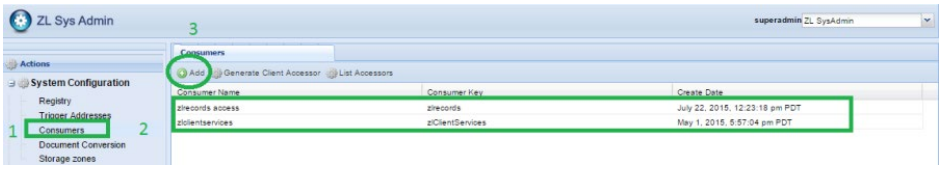
Further details are provided in the following sections.

Generating the Certificate File Body and Password

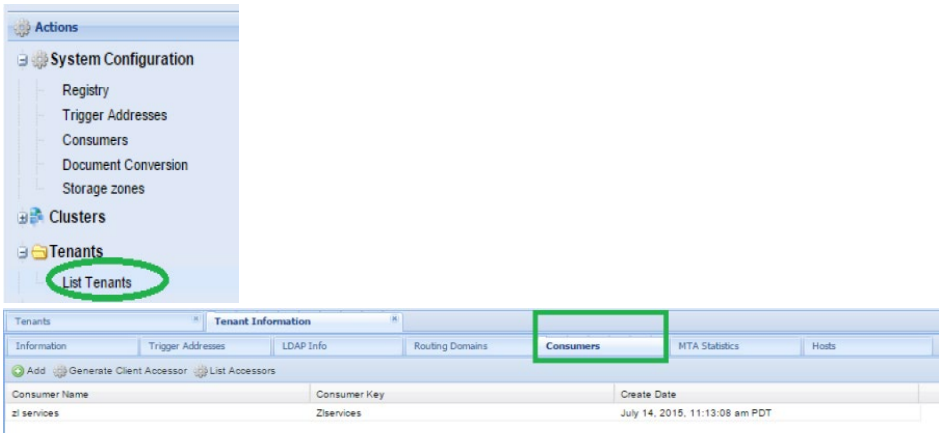
This section describes how to create an OAuth client certificate and password with the ZL UA SysAdmin module. This step must be performed by a System Administrator. For comprehensive details on using the ZL UA SysAdmin module, refer to the *ZL UA System Administrator's Guide.*

To generate the certificate and password, follow these steps:

1. Log into ZL UA and open the SysAdmin module.
2. Go to the appropriate consumer section:
    a. If you are creating a certificate for a user in the default tenant, go to **System Configuration** > **Consumers**.

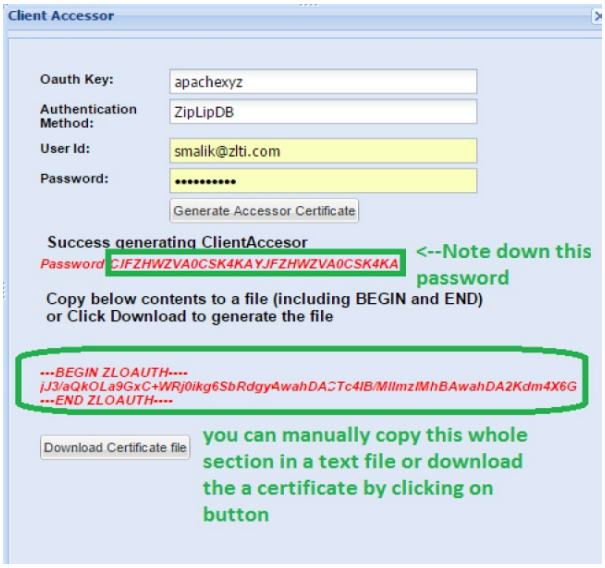


    b. If you are creating a certificate for a user outside the default tenant, go to **Tenants** > **List Tenants**, select the relevant tenant, and then go to the **Consumers** tab.



3. If there is no existing consumer, click **Add** to create one. This can be done once per tenant, and the consumer does not have to be associated with a single user.

4. Click **Generate Client Accessor** to generate a client accessor certificate. The **Oauth Key** and **Authentication Method** fields are automatically populated. For the **User Id** and **Password**, supply the credentials of the relevant user.



5. Record the password and download the certificate file. API functions will need the correct path to the certificate, as demonstrated in the following example.

Client Code Example

A client code example for calling the **getToken** API is shown below. In this example:

- **stCertFile** is the ZL server issued certificate file.
- **stPassword** is the password used to encrypt the certificate.
- **idTenant** is the tenant ID of the ZL server.

```
public class RestAPIClient {
    public static void main(String[] args) {
        Client client = ClientBuilder.newClient();
        WebTarget target = client.target( s: "http://localhost:8080/ps/api/security/getToken");
        String acceptHeader = "*/*";
        String stCert = "wew/1p1XBVQ5/ZcL/N2ebSMVc67XBcX/83IfUd6SeOlTOOD6d02F0+gmPF/" + "KSdtOi8EKODCdOuUscnJHLJ1UcT9a5nLUbYZf0Q3QGFI1/J3
        String stPassword = "GBYQRQPLYYDZZGNCNB4TK0RNWKKM1KRPA";
        int idTenant = -1;
        Form formData = new Form();
        formData.param( name: "certificateBody",stCert);
        formData.param( name: "password",stPassword);
        formData.param( name: "tenant",String.valueOf(idTenant));
        Response response = target.request(MediaType.APPLICATION_FORM_URLENCODED).
                header( s: "Accept",acceptHeader).post(Entity.entity(formData,MediaType.APPLICATION_FORM_URLENCODED_TYPE));
        if (response.getStatus() == 200) {
            // Extract and print the token from the response body
            String token = response.readEntity(String.class);
            System.out.println("Token: " + token);
        } else {
            System.err.println("Error: " + response.getStatus() + ", " + response.readEntity(String.class));
        }
        client.close();
    }
}
```

Figure 1: getTokenAPI

You will get the bearer token after calling this API, and you can send this token in headers to authenticate the REST API calls.

REST API Java Client Examples

This section shows examples of REST API client implementations.

Example 1 - GET Operation: Getting File Server Information

```
public class RestAPIClient {
    public static void main(String[] args) {
        Client client = ClientBuilder.newClient();
        WebTarget target = client.target( s: "http://localhost:8080/ps/api/beta/FileServer/getmailserverusingid/{id}")
                .resolveTemplate( s: "id",  o: "52");
        String token = "CZPQIHHMFRCBOB30RXWRR3HT5UXYQ2UMB";
        String response = target.request(MediaType.APPLICATION_JSON).header(HttpHeaders.AUTHORIZATION,  o: "Bearer " + token).
                get(String.class);
        System.out.println(response);
        client.close();
    }
}
```

Figure 2: Getting File Server Information

Example 2 - Post Operation: Creating File Server

In this example, a client is created and a target URL is built. The request includes ZL bearer tokens in the headers which will hit the ZL API.

```
public class RestAPIClient {
    public static void main(String[] args) {
        Client client = ClientBuilder.newClient();
        String token = "CZPQIHHMFRCBOB30RXWRR3HT5UXYQ2UMB";
        String jsonPayload = "{\"serverName\":\"qdsaqdya111\",\"ip\":\"localhost\",\"port\":\"9975\"}";
        Response response = client.target( s: "http://localhost:8080/ps/api/beta/FileServer/createfileserver")
                .request(MediaType.APPLICATION_JSON).header(HttpHeaders.AUTHORIZATION,  o: "Bearer " + token).post(Entity.json(jsonPayload));
        System.out.println(response.toString());
        client.close();
    }
}
```

Figure 3: Creating File Server

REST API Using Swagger Client

This section shows examples of using the REST API with a Swagger client. Access the Rest API with a Swagger client via the following URL, where “localhost” represents the hostname or IP address of the ZL server:

http://localhost:8080/ps/zlui/app/home/zlswagger.htmlfile

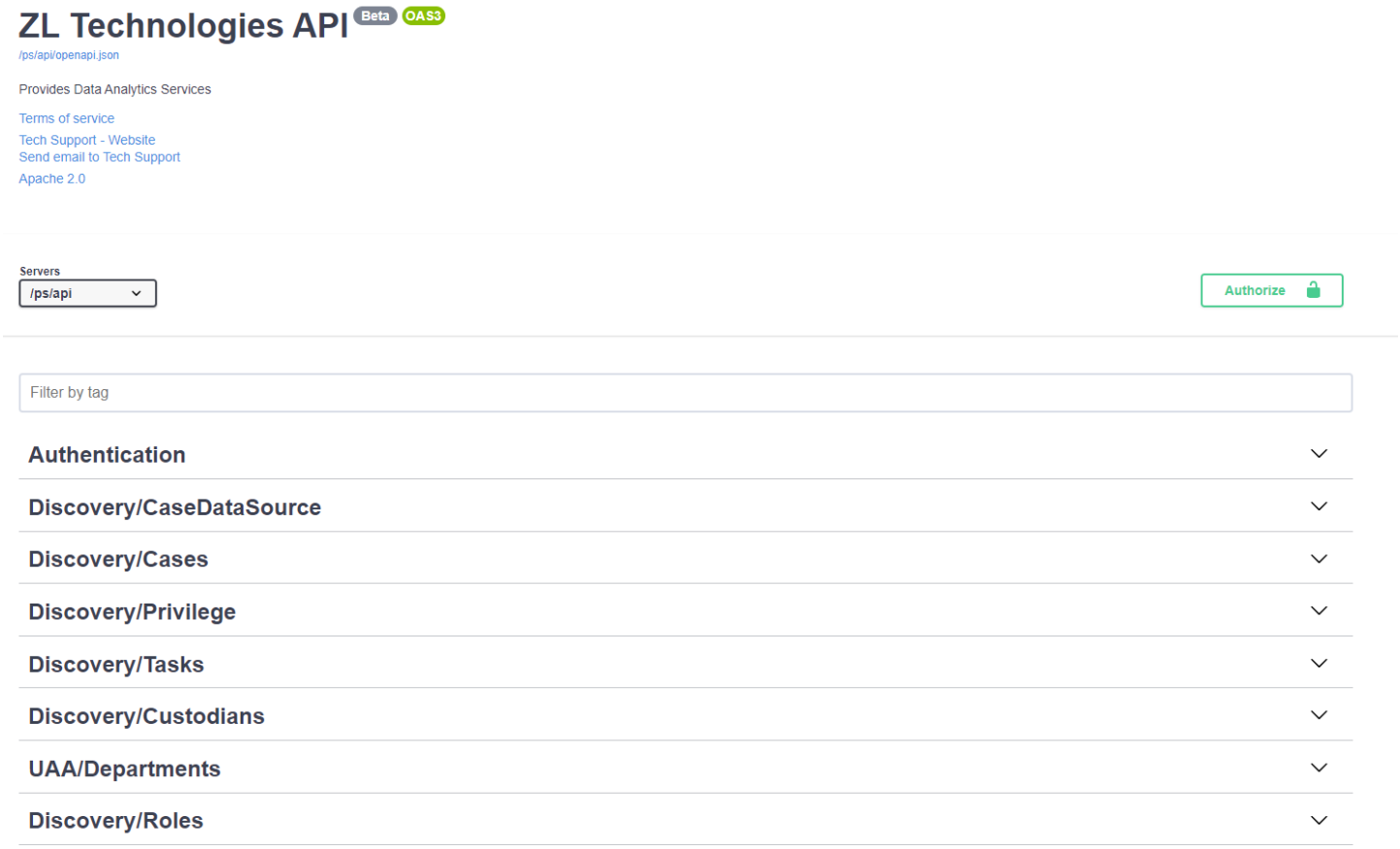


Figure 4: ZL UA Rest API

Authenticating in Swagger

Before using the REST API in Swagger, you must authenticate using the consumer's certificate, password and tenant identification. After authenticating, a token will be generated. You will need to authorize in Swagger using that token. This is shown below.

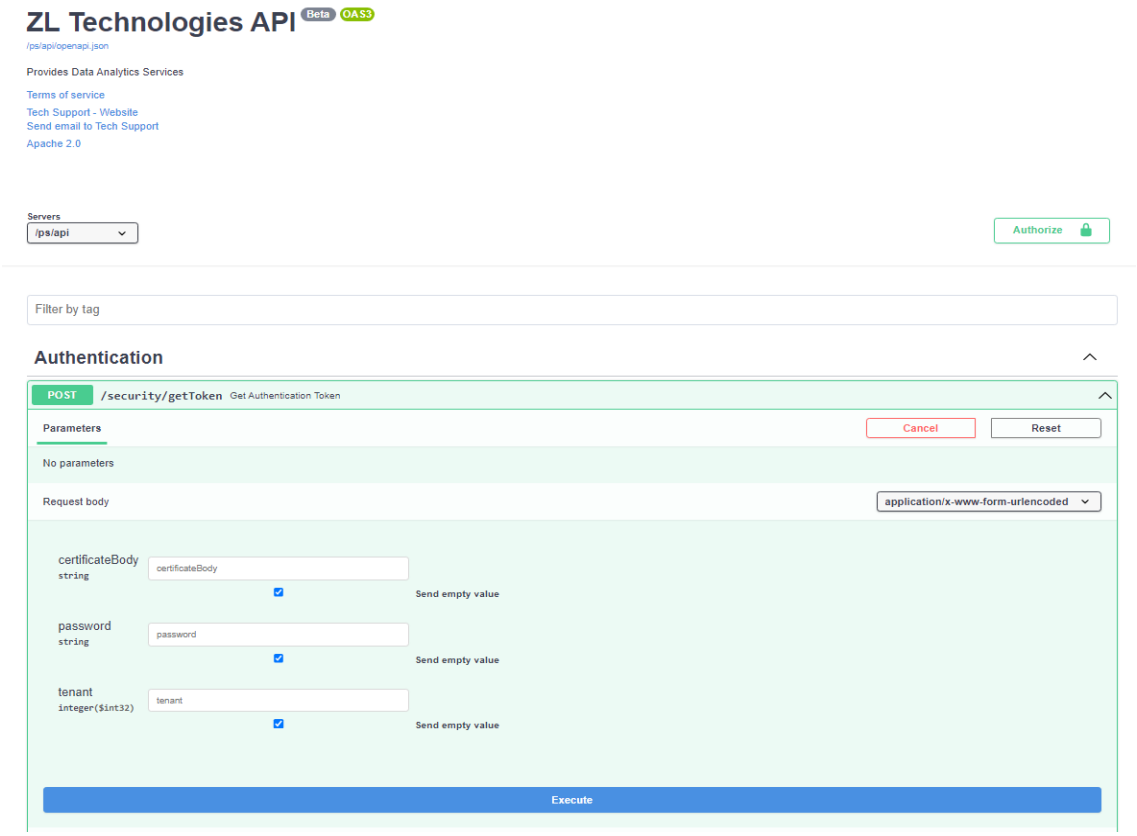


Figure 5: Authentication

Example 1 - GET Operation: Getting File Server Information

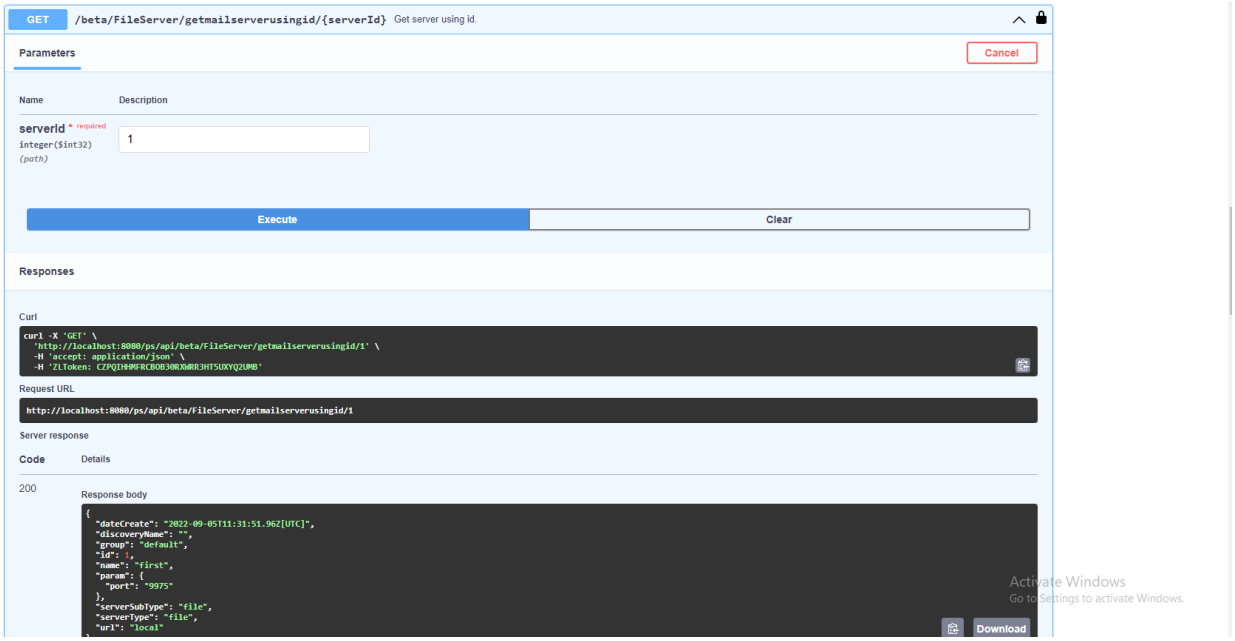


Figure 6: Getting File Sever Information

Example 2 – POST Operation: Creating File Server

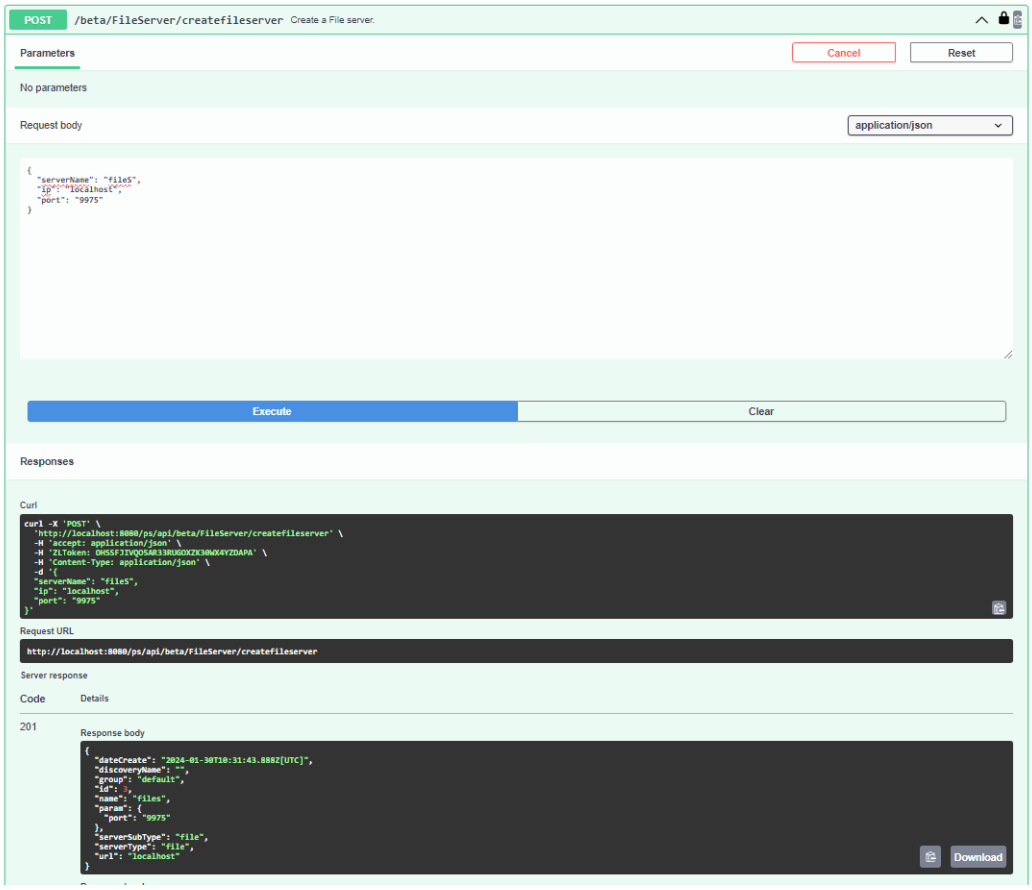


Figure 7: Creating File Server

REST API Using Postman

REST API calls are authorized using ZL bearer tokens which are sent in the bearer token field in the postman along with the HTTP request. This is shown below.

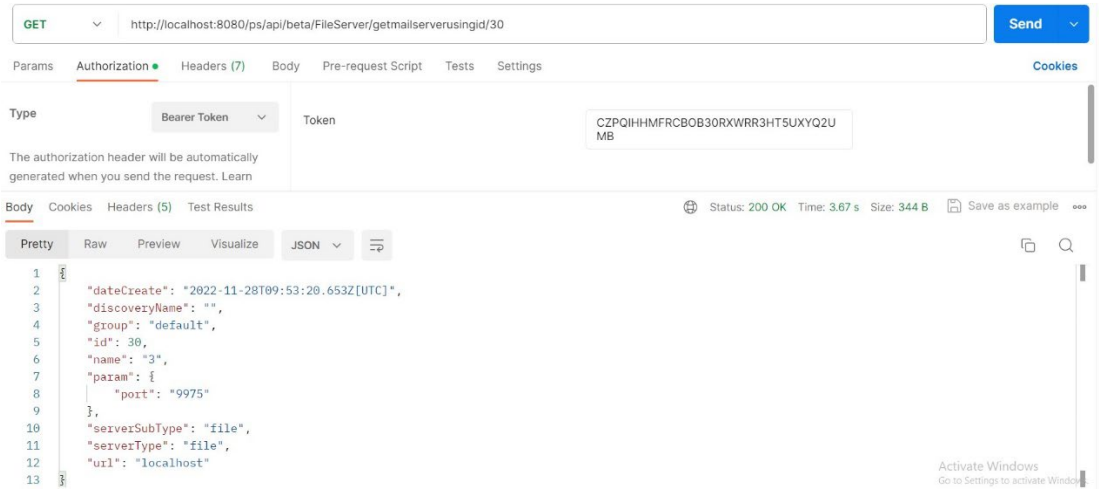


Figure 8: Example 1 - GET Operation: Getting File Server Information

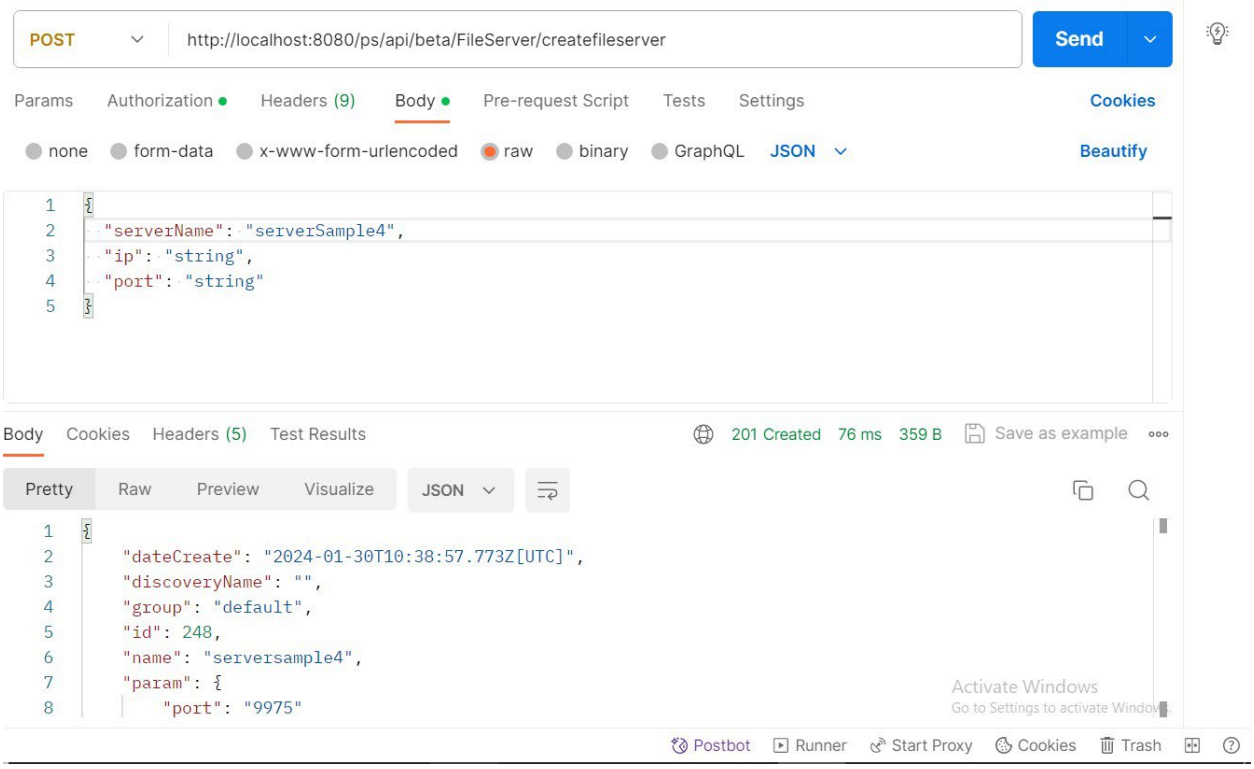


Figure 9: Example 2 – POST Operation: Creating File Server

REST API Endpoints

Each endpoint includes the following inputs and outputs:

- **Request Parameters:** Input defining the object to be retrieved, viewed, created or updated. The request parameters may also include pagination, filter, and sort options for the response data returned by the endpoint.
- **Request Body Schema Fields:** Additional input defining the object to be retrieved, created or updated. For example, if you execute an endpoint to configure a File Share project, these fields would define the intended configuration of the project.
- **Response Schema Fields**: Output defining the object that was retrieved, created or updated. For example, if you execute a call to retrieve the configuration of a content rule, the response schema fields would represent that configuration.
- **Response Codes**: A numeric value indicating the result of the operation. The following response codes are returned by REST API endpoints:

  200: Successful
  201: Created
  202: Accepted in Response Code
  206: Partial Content
  400: Bad Request
  401: Unauthorized
  403: Forbidden/Insufficient Privilege/Unauthorized
  404: Not Found
  406: Not Acceptable
  500: Internal Server Error

What's New In ZL UA 11.1

ZL UA 11.1 includes the following enhancements to the REST API:

- **Updated Workspace/Enterprise Analytics Endpoints:** New endpoints have been added for workspace management operations within the Enterprise Analytics module (ZL UA 11.1.0+).

Endpoint Group	Description
Workspace/Preservation	When running searches to build workspaces, you have the option to save the messages or files returned by the search as a *preservation* and place them on *legal hold*. After the files have been placed on legal hold, they cannot be removed from ZL UA until the legal hold is removed. Use these endpoints to manage preservations within your workspaces. For more information, refer to *Workspace/Preservation* on page 273.
Workspace/Recategorization	Use these endpoints to categorize In-Place files by applying Record Categories to them. For more information, refer to *Workspace/Recategorization* on page 430.
Workspace/Roles	Use these endpoints to grant workspace roles to other users. These roles determine the operations each user will be able to perform within each workspace. For more information, refer to *Workspace/Roles* on page 435.
Workspace/Audit	Use these endpoints to retrieve audit trail data summarizing the operations that have been performed within each workspace. For more information, refer to *Workspace/Audit* on page 440.
Workspace/Export	Use these endpoints to export workspace data. For more information, refer to *Workspace/Export* on page 448.
Workspace/Workspace	Use these endpoints to perform workspace management tasks, such as creating workspaces and adding data sources to them. For more information, refer to *Workspace/Workspace* on page 456

Endpoint Group	Description
Workspace/Search	Use these endpoints to search the data that has been added to a workspace. For more information, refer to *Workspace/Search* on page 490.
Workspace/Stats	Use these endpoints to view cross-tabulation data reflecting the contents of a workspace. These endpoints are applicable to files in full workspaces. For more information, refer to *Workspace/Stats* on page 502.

- **Global Search Endpoints**: Use these new endpoints to conduct Global Searches for In-Place and archived files and mails. For more information, refer to *Global Search* on page 290 (ZL UA 11.1.0+).
- **UAA Licensing Endpoint:** Use this endpoint to retrieve licensing information regarding the mailing lists and users within your ZL UA installation. For more information, refer to *UAA/License* on page 271 (ZL UA 11.1.0+).
- **Updates to Enterprise Files Management (EFM) Endpoints:** Changes and additions to endpoints affecting EFM operations have been made.

Endpoint Group	Description
EFM/Audits	Use these new endpoints to view audit trail report data for an EFM project. The audit trail report data summarizes the actions that have been performed within the project (e.g., tagging, disposition and project management operations). For more information, refer to *EFM/Audits* on page 128 (ZL UA 11.1.0+).

Endpoint Group	Description
EFM/Disposition	These endpoints include the following changes: • **PUT: Disable Disposition***:* Use this new endpoint to disable the Disposition Workflow within a project (ZL UA 11.1.1+). • **PUT: Enable Disposition:** Use this new endpoint to enable the Disposition Workflow within a project (ZL UA 11.1.1+). • **GET: Get Projects Pending Approval:** Use this new endpoint to retrieve a list of projects with Disposition Runs that are pending approval (ZL UA 11.1.1+). • **GET: Get Projects Ready for Disposition:** Use this new endpoint to retrieve a list of projects with Disposition Runs that are ready for disposition (ZL UA 11.1.1+). • **PUT: Re-include all Excluded Items:** Use this new endpoint to re-include all files that had been manually excluded from a Disposition Run (ZL UA 11.1.1+). • **PUT: Disable Exclusion of Items:** Use this new endpoint to disable the exclusion of files from Disposition Runs (ZL UA 11.1.1+). For more information, refer to *EFM/Disposition* on page 133.

Endpoint Group	Description
EFM/Reports	These endpoints include the following changes: • **POST: Generate All Excluded Items Report:** Use this new endpoint to download a report summarizing the files that are eligible for disposition within your EFM projects but have been excluded (ZL UA 11.1.1+). • **POST: Generate All Project User Privileges Report:** Use this new endpoint to download a report summarizing the user privileges that have been granted within your EFM projects (ZL UA 11.1.1+). • **POST: Generate Folder Report:** Use this endpoint to download a Folder Report for an EFM project. This includes information such as the size of the folder, the number of sub-folders and files included within the folder, and the Record Category Policy Rule that has been applied to the folder (ZL UA 11.1.1+). • **POST: Generate File In-Place Manifest Report:** This endpoint has been updated to include a new option to filter the File Manifest Report to include files that were last modified during a specific date range (ZL UA 11.1.0+). • **POST: Generate Out of Sync Report:** Use this endpoint to generate the Out of Sync Report for an EFM project. The Out of Sync Report summarizes the out of sync files and folders in a project. A file or folder will become out of sync if you manually tag it or mark it as a record and it is subsequently re-named or deleted. In this case, the instance of the tag or record created in the database for the file or folder will be orphaned. This is because the database only includes the original path of the file or folder and can't tell if it has been deleted or re-named. (ZL UA 11.1.2+) • **POST: Generate In-Place File Disposition Report:** Use this endpoint to generate a Post Disposition Report that lists the file name, folder name and path of the files that were disposed of during a Disposition Run, as well as other file attributes such as the file name and path, folder name, extension and size, record category, retention code and disposition eligibility date. (ZL UA 11.1.0+) • **POST: Generate Record Category Search:** Use this new endpoint to generate a Record Category Search Report that lists the files that the Record Category has been applied to within the specified group of projects (ZL UA 11.1.0+). For more information, refer to *EFM/Reports* on page 153.
EFM/Tasks	Added the **GET: Recategorization Runs** endpoint, which you can use to view status information for recategorization runs that have been executed within a project. For more information, see *EFM/Tasks* on page 163 (ZL UA 11.1.1+).

Endpoint Group	Description
EFM/Project Privileges	In the EFM module, you can assign project privileges to both users and security groups with the EFM module to determine what operations they will be able to perform within each project. Privileges are granted on a project-by-project basis, so a user or security group can have different privilege levels for different projects. Use these new endpoints to grant, revoke and manage project privileges within the EFM module. For more information, refer to *EFM/Project Privileges* on page 167 (ZL UA 11.1.1+).
EFM/Security Groups	A security group represents a specific group of users in ZL UA. When you assign project privileges to a security group within the EFM module, those privileges are granted to all users in the security group. Use these endpoints to create and manage security groups. For more information, refer to *EFM/Security Groups* on page 188 (ZL UA 11.1.1+).
EFM/Tags	Added the **PUT: Remove Record Category** endpoint, which you can use to remove a Record Category from a file. For more information, see *EFM/Tags* on page 192 (ZL UA 11.1.1+).

ZL UA Operations in the REST API

In ZL UA 11.1, endpoints have been provided for the following purposes:

- **Discovery/Case Data Source:** In the ZL Discovery Manager, you can search message and file data and save the results to a particular eDiscovery case as a *collection* or a *preservation.* These saved searches are referred to as *data sources.*

  Use these endpoints to view information regarding the data sources that have been added to a case and to schedule future data source runs. For more information, see *Discovery/CaseDataSource* on page 21.

- **Discovery/Cases:** All the searches, preservation applications, item review, analysis, exports, and other tasks in ZL Discovery Manager occur within cases. Cases are a matter-specific way of organizing documents and managing permissions in one place. ZL Discovery Manager users may have permissions for multiple cases, and a document can be present in more than one case.

  Use these endpoints to create and manage cases, including the tags that are available within the case and the user privileges assigned within the case. For more information, see *Discovery/Cases* on page 30.

- **Discovery/Privileges:** In the ZL Discovery Manager, privileges can also be assigned within each eDiscovery case to grant users access to operations within the case. A user's role may provide access to operations not allowed by his or her privileges - and vice versa - so it is important to remember that each user will have access to all operations allowed by his assigned role(s) and privilege(s).

  Use these endpoints to grant and manage privileges within a case. For more information, see *Discovery/Privileges* on page 62.

- **Discovery/Custodians:** Custodians are targets of an eDiscovery investigation who are likely to have knowledge about the location of documents that are pertinent to a case. In ZL Discovery Manager, custodians function as a way in which aliases and email addresses can be associated with an individual to facilitate searches.

  Once an individual has been added as a case custodian, case users can use the custodian as a filter when searching for documents to save to the case, and when searching across documents that have already been saved to the case.

  Use these endpoints to manage custodians within a case. For more information, see *Discovery/Custodians* on page 69.

- **UAA/Departments:** A department is defined by ZL UA as a hierarchical grouping of users. Each user in ZL UA is grouped into a department that has been added to the system. The departmental hierarchy configured in ZL UA often resembles the organization's actual departmental structure, but this configuration model is not required.

  Use these endpoints to create and manage departments in ZL UA. For more information, see *UAA/Departments* on page 90.

- **Discovery/Reports**: Use these endpoints to generate various reports containing details on eDiscovery searches, cases and custodians. For more information, see *Discovery/Reports* on page 107.

- **Discovery/Roles:** In the Discovery Manager, a role is an application or department-level set of permissions that determines what users can access and what actions users can perform in the application. All ZL UA roles can be assigned to users by administrators from the Unified Archival Admin (UAA) module, and the eDiscovery-related roles described in this appendix can also be assigned with the Discovery Manager.

  Use these endpoints to grant and manage roles within a case. For more information, see *Discovery/Roles* on page 124.

- **EFM/Audits:** Use these endpoints to view audit trail data summarizing the operations that have been performed within the EFM module. For more information, refer to *EFM/Audits* on page 128.

- **EFM/Disposition:** Manage the Disposition Workflow within EFM projects. The Disposition Workflow is the process by which files whose records management lifecycle has expired are deleted and removed from the ZL UA system.

  Use these endpoints to enable and disable disposition, configure settings affecting disposition will be managed within each project, and view disposition status. For more information, refer to *EFM/Disposition* on page 133.

- **EFM/Reports:** Use these endpoints to generate various reports summarizing the files that are included in the EFM projects that have been added to your installation. For more information, refer to *EFM/Reports* on page 153.

- **EFM/Tasks:** Use these endpoints to monitor the status of tasks involving the application of Record Categories in the EFM module. For more information, refer to *EFM/Tasks* on page 163.

- **EFM/Project Privileges:** In the EFM module, you can assign project privileges to both users and security groups with the EFM module to determine what operations they will be able to perform within each project. Privileges are granted on a project-by-project basis, so a user or security group can have different privilege levels for different projects.

  Use these endpoints to grant, revoke and manage project privileges within the EFM module. For more information, refer to *EFM/Project Privileges* on page 167.

- **EFM/Security Groups:** A security group represents a specific group of users in ZL UA. When you assign project privileges to a security group within the EFM module, those privileges are granted to all users in the security group.

  Use these endpoints to create and manage security groups. For more information, refer to *EFM/Security Groups* on page 188.

- **EFM/Tags:** In ZL UA, a Record Category defines a file's disposition period (i.e., how long it will be retained within ZL UA before being permanently deleted from ZL UA and the source server). When creating Record Categories, you will also need to create retention codes which define the disposition periods in your system.

  Use these endpoints to create and manage Record Categories for use with the EFM module. For more information, see *EFM/Tags* on page 192.

- **EFM/Audits:** Use these endpoints to view audit trail report data for an EFM project. The audit trail report data summarizes the actions that have been performed within the project (e.g., tagging operations, project management operations, etc). For more information, see *EFM/Audits* on page 128.
- **FAM/Roles**: Assign roles to users. A role is the function of an individual, independent of any conditions. Roles are assigned on either system-wide or departmental levels. A ZL user can be assigned any number of system roles.

  Use these endpoints to grant and revoke user roles for use within the FAM module. For more information, see *FAM/Roles* on page 208.
- **UAA/Data Sources**: To archive any type of data, a reference pointing to its server must be created in the ZL Unified Archival Admin (UAA) application, regardless of the server type (mail, file, SharePoint, etc). This reference - referred to as a ZL UA application server or a content server - integrates ZL UA with the physical server on the corporate network.

  Use these endpoints to create, manage and update mail servers in ZL UA. For more information, see *UAA/Data Sources* on page 212.
- **FAM/Disposition**: Disposition is the process by which files whose records management lifecycle has expired are deleted and removed from the ZL UA system.

  Use these endpoints to enable and disable disposition and view disposition status within FAM projects. For more information, see *FAM/Disposition* on page 254.
- **FAM/Tasks**: The ZL FAM module includes background tasks that must be executed after the completion of certain operations. For example, after creating a project, you would need to execute the **Run Crawl**, **Update Content Index**, **Update Index** and **Clear Cache** background tasks to scan the project directories for files and update the project's content index and tag index.

  Use these endpoints to execute background tasks on FAM projects. For more information, see *FAM/Tasks* on page 262.
- **UAA/License**: Use this endpoint to retrieve licensing information regarding the mailing lists and users within your ZL UA installation. For more information, refer to *UAA/License* on page 271.
- **Workspace/Preservation**: When running searches to build workspaces, you have the option to save the messages or files returned by the search as a *preservation* and place them on *legal hold*. Once files have been placed on legal hold, they cannot be removed from ZL UA until the legal hold is removed.

  Use these endpoints to manage preservations within your workspaces. For more information, refer to *Workspace/Preservation* on page 273.
- **Global Search:** Use these endpoints to conduct Global Searches on In-Place and Archived files and mails. For more information, refer to *Global Search* on page 290.
- **FAM/Project Privileges**: Use these endpoints to manage the privileges that have been assigned to users within a FAM project. These privileges determine what operations the users can perform within the project. For more information, see *FAM/Project Privileges* on page 327.
- **UAA/Projects** and **FAM/Projects**: A project is a list of folders or sites that is grouped together to be scanned whenever a server is crawled. Projects are created to determine which system directories or

sites (and, subsequently, which items) in the selected server are to be archived and/or managed in place. You can create projects with either the UAA module or the FAM module.

Use these endpoints to create and manage projects on various server types. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

- **UAA/Roles**: An application or department-level set of permissions that determines what users can access and what actions users can perform in the application.

  Roles can be assigned globally, or for a specific department(s). For example, a Global Discovery Manager role would enable the user's assigned role for all cases. A Discover Manager role for a specific department would restrict the user's role to the cases defined within that department.

  Use these endpoints to grant and revoke roles within the UAA module and ZL UA. For more information, refer to *UAA/Roles* on page 369.

- **FAM/Security Groups**: A security group represents a specific group of users in ZL UA. When you assign project privileges to a security group, those privileges are granted to all users in the security group. Use these endpoints to create and manage security groups. For more information, refer to *FAM/Security Groups* on page 379.

- **UAA/Agents:** Server agents are components that perform various tasks on the server, such as crawling the server for data or performing user synchronization. Generally, each server added to ZL UA must have at least one server agent.

  Use these endpoints to create and manage server agents. For more information, refer to *UAA/Agents* on page 383.

- **FAM/Tags**: Tags are customizable labels that can be applied to documents for various purposes. You could apply tags to the results of a search or file sampling to mark those files for retrieval later or apply tags to mark files that are subject to review, and so on.

  You can also use tags for remediation. When you configure remediation, you assign an action to a tag (e.g., to copy, delete, or move the file). When you execute remediation, that action will be applied to all the files that the tag has been applied to. For example, you could use remediation to move all files that a tag has been applied to from one folder to another.

  You can upload tag definition files and tag specifications into ZL UA to create tags for use in your system. A tag definition file defines and creates tags. These tags can be applied to files manually, or via a tag specification. A tag specification defines a set of rules and conditions, each of which specifies a tag that will be applied to files that meet the terms of the rules and conditions. For example, you could create a tag specification to tag all files that contain the phrase confidential agreement" in the body of an email with the "Privileged" tag. You can upload tag specifications that will tag files based on content, metadata and PII data.

  Use these endpoints to manage tags in REST API and to upload tag definition files and tag specification files. For more information, refer to *FAM/Tags* on page 389.

- **UAA/Users**: A user is a person whose email address(es) and alias(es) are recognized in the ZL UA system. All users registered in ZL UA are associated with one department. Users can inherit policy settings from their department. However, privileged users can configure custom settings to override the

inherited department settings for a particular user. A ZL UA user typically has a primary email address, as well as one or more *alias* email addresses which can be used to locate that user.

Use these endpoints to create and manage users. For more information, see *UAA/Users* on page 406.

- **Workspace/Recategorization:** Use these endpoints to categorize files in Enterprise Analytics workspaces by applying Static Tags and Record Categories to them. For more information, refer to *Workspace/Recategorization* on page 430.
- **Workspace/Roles:** Use these endpoints to grant workspace roles to other users. These roles affect the operations they will be able to perform within each workspace. For more information, refer to *Workspace/Roles* on page 435.
- **Workspace/Audit:** Use these endpoints to retrieve audit trail data summarizing the operations that have been performed within workspaces. For more information, refer to *Workspace/Audit* on page 440.
- **Workspace/Export:** Use these endpoints to export workspace data. For more information, refer to *Workspace/Export* on page 448.
- **Workspace/Workspace**: Use these endpoints to perform workspace management tasks, such as creating workspaces and adding data sources to them. For more information, refer to *Workspace/Workspace* on page 456.
- **Workspace/Search**: Use these endpoints to search workspace data. For more information, refer to *Workspace/Search* on page 490.
- **Workspace/Stats:** Use these endpoints to view tabulation data reflecting the contents of a workspace. These endpoints are applicable to files in full workspaces. For more information, refer to *Workspace/Stats* on page 502

Discovery/CaseDataSource

In the ZL Discovery Manager, you can search archived data and save the results into an eDiscovery case as a *collection* or a *preservation.* These saved searches are referred to as *data sources.*

When you save the search to the case, you can configure scheduling options to automatically re-run the search later. New search results that are returned each time the search is re-executed will be imported into the case after the appropriate background tasks – e.g., the *import task* - are executed. Each future invocation of the search used to create the data source is referred to as a *data source run.*

The following sections describe the Discovery/CaseDataSource endpoints available in the REST API. Use these endpoints to view information regarding the data sources that have been added to a case, the data source runs that have been executed upon a case, and to schedule future data source runs:

- *GET: Get All Case Data Source Runes (getallcasedatasourceruns):* View the data source runs that have been executed for a data source.
- *GET: Get Case Data Source Run Schedule (getcasedatasourcerunschedule):* View the data source run schedule for a data source.
- *GET: Get Latest Case Data Source Run (getlatestcasedatasourcerun):* View the details of the latest data source run executed for a particular data source.
- *POST: Schedule Case Data Source (scheduleDataSource):* Schedule future runs for a data source.
- *DELETE: Unschedule Data Source (unscheduleDataSource):* Remove future run scheduling for a data source.

GET: Get All Case Data Source Runs (getallcasedatasourceruns)

Retrieve information for the data source runs that have been executed for a particular data source.

Path

http://localhost:8080/ps/api/v1/casedatasource/getallcasedatasourcesruns/{dataSourceId}

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
dataSourceId	Integer	Specify the ID of the data source you want to view. You can use the GET: Get All Case Data Sources (getallcasedatasources) endpoint described on page 42 to retrieve the ID values of all data sources that have been added to a particular case.
page **pageSIze**	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.

Response Schema Fields

Schema Field	Type	Description
Data Source Run Entries: The endpoint returns an entry for each run that has been executed for the specified data source. The following information is included within each entry:		
id	Integer	The data source run ID.
caseDataSourceId	Integer	The data source ID.
caseId	Integer	The ID of the case the data source run belongs to.

Schema Field	Type	Description
status	Integer	The status of the data source run: 0: STATUS NONE 10: STATUS SEARCH INITIATED 499: STATUS SEARCH DONE ERROR 500: STATUS SEARCH DONE 502: STATUS INPLACE ITEM RENCONCILE 504: STATUS INPLACE ITEM ARCHIVE 510: STATUS IMPORT INITIATED 999: STATUS IMPORT DONE_ERROR 1000: STATUS IMPORT DONE 1001: STATUS SEARCH ABORT 1002: STATUS IMPORT ABORT 1003: STATUS OTHER ABORT
searchPID	String	The process ID assigned to the data source run.
dateSearchStart	String	The date/times that the data source run started, was last updated and ended. When creating a data source run, the date the search started will march the date it was last updated.
dateSearchUpdate	String	
dateSearchEnd	String	
searchItemCount	String	The number of items returned by the search.
searchStatusMsg	String	The status of the search.
importPID	String	The process ID of the session. This is the hostname of the machine on which the search is running.
importStartDate	String	The date/times that the import task for the case began and ended. New data discovered during the search is added to the case when the import task is executed.
importUpdate	String	
importEndDate	String	
newCaseItemCount	Integer	The number of items included in the case after the data source run was executed and the newly discovered items were added to the case.
newCaseItemReferenceCount	Integer	Total case items that already exist in the case due to other searches/data sources.

Schema Field	Type	Description
previousCount	Integer	The number of items included in the case before the data source run was executed.
errorCount	Integer	The number of errors that occurred during the import task.
importStatusMsg	String	The status of the import task.

GET: Get Case Data Source Run Schedule (getcasedatasourcerunschedule)

Retrieve the scheduling information configured for future re-runs of a data source.

Path

http://localhost:8080/ps/api/v1/casedatasource/getcasedatasourcerunschedule/{dataSourceId}

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
dataSourceId	Integer	Specify the ID of the data source you want to view. Scheduling information for this data source will be retrieved. You can use the GET: Get All Case Data Sources (getallcasedatasources) endpoint described on page 42 to retrieve the ID values of all data sources that have been added to a particular case.

Response Schema Fields

Schema Field	Type	Description
dataSourceId	Integer	The data source ID.
dateStart	String	The date and time that the data source run schedule began.
intervalMs	Integer	The run interval, in milliseconds. The search will be re-run at this interval during the date range specified by the **dataStart** and **dateExpiry.**
dateExpiry	String	The date and time that the data source run schedule ends.

GET: Get Latest Case Data Source Run (getlatestcasedatasourcerun)

Retrieve information for the most recent run that has been executed for a particular data source.

Path

http://localhost:8080/ps/api/v1/casedatasource/getlatestcasedatasourcerun/{dataSourceId}

Request Parameters

Parameter	Type	Description
dataSourceId	Integer	Specify the ID of the data source you want to view. You can use the GET: Get All Case Data Sources (getallcasedatasources) endpoint described on page 42 to retrieve the ID values of all data sources that have been added to a particular case.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
id	Integer	The data source run ID.
caseDataSourceId	Integer	The data source ID.
caseId	Integer	The ID of the case the data source run belongs to.
status	Integer	The status of the data source run.
searchPID	String	The process assigned to the data source run.
dateSearchStart **dateSearchUpdate** **dateSearchEnd**	String	The date/times that the data source run started, was last updated and ended. When creating a data source run, the date the search started will march the date it was last updated.
searchItemCount	String	The number of items returned by the search.
searchStatusMsg	String	The status of the search.
importPID	String	The process ID of the session. This is the hostname of the machine on which the search is running.

Schema Field	Type	Description
importStartDate **importUpdate** **importEndDate**	String	The date/times that the import task for the case began and ended. New data discovered during the search is added to the case when the import task is executed.
newCaseItemCount	Integer	The number of items included in the case after the data source run was executed and the newly discovered items were added to the case.
newCaseItemReferenceCount	Integer	Total case items that already exist in the case due to other searches/data sources.
previousCount	Integer	The number of items included in the case before the data source run was executed.
errorCount	Integer	The number of errors that occurred during the import task.
importStatusMsg	String	The status of the import task.

POST: Schedule Case Data Source (scheduleDataSource)

Configure the scheduling options for future re-runs of a data source.

Path

http://localhost:8080/ps/api/v1/casedatasource/scheduledatasource

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
dataSourceId	Integer	Specify the ID of the data source to be scheduled. You can use the GET: Get All Case Data Sources (getallcasedatasources) endpoint described on page 42 to retrieve the ID values of all data sources that have been added to a particular case.
dateStart	String	The date and time to begin the data source runs.
intervalMs	Integer	The run interval, in milliseconds. The search will be re-run at this interval during the date range specified by the **dataStart** and **dateExpiry.**
dateExpiry	String	The date and time to end the data source runs.

Response Schema Fields

Parameter	Type	Description
dataSourceId	Integer	The ID of the data source that has been scheduled.
dateStart	String	The date and time that the future data source runs will begin.
intervalMs	Integer	The run interval, in milliseconds. The search will be re-run at this interval during the date range specified by the **dataStart** and **dateExpiry.**
dateExpiry	String	The date and time to end the future data source runs.

DELETE: Unschedule Data Source (unscheduleDataSource)

Delete the scheduling for a data source, so that no future re-runs will be executed.

Path

http://localhost:8080/ps/api/v1/casedatasource/unscheduledatasource/{dataSourceId}

Request Parameters

Parameter	Type	Description
dataSourceId	Integer	Specify the ID of the data source to be unscheduled. You can use the GET: Get All Case Data Sources (getallcasedatasources) endpoint described on page 42 to retrieve the ID values of all data sources that have been added to a particular case.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the scheduling was deleted successfully.

Discovery/Cases

All the searches, preservation applications, item review, analysis, exports, and other tasks in the ZL Discovery Manager occur within cases. Cases are a matter-specific way of organizing documents and managing permissions in one place. ZL Discovery Manager users may have permissions for multiple cases, and a document can be present in more than one case.

You can search archived data and save the results to a particular eDiscovery case as a *collection* or a *preservation.* These saved searches are referred to as *data sources* within the case. After a file or email message has been added to a case, you can apply tags to those files. Tags are a way to mark files or emails that include keywords and phrases that are considered particularly important to a particular case.

The following sections describe the Discovery/Cases endpoints available in the REST API. Use these endpoints to create and manage cases, including the tags that are available within the case and the user privileges assigned within the case:

- *POST: Create a Tag (addtag)*
- *PUT: Apply Case Tags (applyCaseTags)*
- *POST: Create a Case (createcase)*
- *POST: Create a Custodian Legal Hold (createCustodianLegalHold)*
- *DELETE: Delete Case (deleteCase)*
- *DELETE: Delete Tag (deleteTag)*
- *GET: Get All Case Data Sources (getallcasedatasources)*
- *GET: Get Case Info (getallcaseusingnamepattern)*
- *GET: Get All Case Tags (getallcasetags)*
- *GET: Get All Child Tags (getallchildtags)*
- *GET: Get All Cases Using Domain ID (getAllCasesUsingDomain)*
- *GET: Get List of All Schema (getAllSchema)*
- *GET: Get Case Data Source Using ID (getcasedatasourceusingsrcid)*
- *GET: Get Case Using ID (getcaseusingid)*
- *GET: Get Case Schema Using Case ID and Field Item Schema (getCaseSchema)*
- *GET: Get Case Schema Metadata (GetCaseSchemaMetaData)*
- *GET: Get Search Tree (getSearchTree)*
- *PUT: Update Case Item Schema (updatecaseitemschema)*
- *PUT: Update Case Schema (updatecaseschema)*
- *PUT: Update Case Schema Metadata (updateCaseSchemaMetaData)*
- *PUT: Update Tag (updatetag)*

POST: Create a Tag (addtag)

You can apply tags to files or emails that include keywords and phrases that are considered particularly important to an eDiscovery case. Use this endpoint to add a tag to a case so that it can be applied to documents within the case.

Path

http://localhost:8080/ps/api/v1/cases/addtag

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
tagName	String	The name of the tag.
tagDescription	String	A description of the tag.
caseId	Integer	The ID of the case the tag should be added to. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
parentTagId	Integer	The ID of the parent tag that the new tag will be grouped under in the tag hierarchy. You can retrieve tag IDs with the GET: Get All Case Tags (getallcasetags) endpoint described on page 45.

Response Schema Fields

Schema Field	Type	Description
parent	String	The name of the tag's parent tag.
id	Integer	The tag ID.
caseId	Integer	The ID of the case the tag belongs to.
parentId	Integer	The ID of the tag's parent tag.
name	String	The internal name of the tag.
displayName	String	The display name of the tag.

Schema Field	Type	Description
tagFlags	Array	An array of Boolean values indicating the status of various tag attributes.
root_node	Boolean	If True, it indicates that the tag is a root-level tag.
read_only	Boolean	If True, it indicates that the tag is read-only.
enduser_tag	Boolean	If True, it indicates that the tag can be applied manually.
auto_tag	Boolean	If True, it indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag	Boolean	If True, it indicates that the tag is a mutually exclusive tag.
tag_32	Boolean	If True, it indicates that the tag is a PII tag.
tag_64	Boolean	If True, it indicates that the tag is a content tag.
description	String	A description of the tag.
createDate	String	The date and time that the tag was created.

PUT: Apply Case Tags (applyCaseTags)

Apply tags to a case item.

Path

http://localhost:8080/ps/api/v1/cases/applyCaseTags

Request Parameters

Parameter	Type	Description
page **pageSize**	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.

Request Body Schema Fields

Schema Field	Type	Description
caseId	Integer	Specify the ID of the case containing the item you want to apply tags to. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
caseItemId	Integer	Specify the ID of the case item you want to apply tags to. You can extract case item IDs from the **CaseItem** database table.
tagIds	Integer	An array of tag IDs to specify the tags you want to apply to the case item. Specify the tag IDs as a comma-separated list. For example: `"tagIds": [ 0,1,2,3 ]` You can retrieve tag IDs with the GET: Get All Case Tags (getallcasetags) endpoint described on page 45.

Response Schema Fields

A string indicating whether the tags were applied successfully.

POST: Create a Case (createcase)

Create a new case.

Path

http://localhost:8080/ps/api/v1/cases/createcase/{idDept}

Request Parameters

Parameter	Type	Description
idDept	Integer	Specify the case department. This field allows you to limit access to the case to privileges with ZL Discovery Manager roles with the proper department scope. If a case's department falls outside of the department scope of a ZL Discovery Manager user's role privileges, the user will not be able to view or access the case (unless granted additional case level privileges).

Request Body Schema Fields

Schema Field	Type	Description
name	String	Enter the name of the case.
docketNumber	String	Enter the case's docket number.
jurisdiction	String	Enter any notes concerning the case's jurisdiction.
category	Integer	Specify a case category to label the type of case. This option does not affect behavior of the application. It is for informational purposes only: • 1: GENERAL LITIGATION • 2: ANTITRUST • 3: LABOR AND EMPLOYMENT • 4: BREACH OF CONTRACT • 5: IP/PATENT • 6: REAL ESTATE • 7: GOVERNMENT INVESTIGATION • 8: INTERNAL MATTER

Schema Field	Type	Description
state	Integer	Specify the current state of the case. This does not affect the behavior of the case. It is for informational purposes only.
fileDate	String	Specify the filing date for the case.
caseDescription	String	Enter a description of the case.
preservation	Boolean	If set to True, documents can be saved to the case through custodian preservation or ECA search without creating a collection for review and analysis.
collection	Boolean	If set to True, documents can be saved to the case for further review and analysis. The items need not be part of a preservation, as items within a collection are also preserved automatically.
enableContentIndex	Boolean	If set to True, a content index for documents collected in the case can be built, enabling content-based searching of collected items.
exportWorkflow	Boolean	Set to True to mandate that any case item export requests must be approved by administrative users before the case items are successfully exported in the specified format. Set to False to allow users with export permissions to export case items immediately without a required approval process.

Response Schema Fields

Schema Field	Type	Description
id	Integer	The case ID.
name	String	The name of the case.
displayName	String	The display name of the case.
createDate	String	The date and time the case was created.
fileDate	String	The file date specified for the case,
lastUpdate	String	The date and time the case was last updated,
state	Integer	The current state of the case.
ownerZlpUserId	Integer	The ZLP user ID of the user who created the case.

Schema Field	Type	Description
journalDomainId	Integer	The journal domain ID of the ZL department associated with the case.
docketNumber	String	The case's docket number.
jurisdiction	String	Any notes concerning the case's jurisdiction.
category	Integer	The category applied to the case.
caseDescription	String	A description of the case.
preservation	Boolean	If set to True, documents can be saved to the case through custodian preservation or ECA search without creating a collection for review and analysis.
collection	Boolean	If set to True, documents can be saved to the case for further review and analysis. The items need not be part of a preservation, as items within a collection are also preserved automatically.
enableContentIndex	Boolean	If set to True, a content index for documents collected in the case can be built, enabling content-based searching of collected items.
exportWorkflow	Boolean	If set to True, any case item export requests must be approved by administrative users before the case items are successfully exported in the specified format.

POST: Create a Custodian Legal Hold (createCustodianLegalHold)

Add a custodian to a case.

Path

http://localhost:8080/ps/api/v1/cases/createCustodianLegalHold

Request Parameters

Parameter	Type	Description
page **pageSize**	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.

Request Schema Fields

Parameter	Type	Description
idCase	Integer	Specify the ID of the case the custodian(s) should be added to. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
custodianIds	Integer	An array of ZLP user IDs identifying the custodians who should be added to the case. Specify the user IDs as a comma-separated list. For example: `"custodianIds": [ 0,1,2,3 ]` The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.

Parameter	Type	Description
srchStoreIds	Integer	An array of search store IDs to specify the search stores which include data for the custodians. Specify the user IDs as a comma-separated list. For example: `"srchStoreIds": [ 0,1,2,3 ]` You can extract search store IDs from the **SearchStore** database table.
datePreservationBegin **datePreservationEnd**	String	These fields specify the date range for the preservation. Messages sent to - or received by - the custodian during this date range will be added to the preservation.
dateFuturePreservationEnd	String	If **isFuturePreservationEnabled** is set to True, specify the end date at which future messages for the custodian will be sent.
isFuturePreservationEnabled	Boolean	If set to True, any messages ingested into ZL UA that are sent to or received by this custodian will be placed on legal hold and saved to the case during processing. Use the **dateFuturePreservationEnd** field to specify the date at which future messages will no longer be placed on legal hold and saved to the case.
isRunSearchImportEnabled	Boolean	For internal use.
Notes	String	Additional information about the preservation.

Response Schema Fields

The response schema includes the following information for each custodian who was added to the case.

Parameter	Type	Description
Custodian Entries: The response includes an entry for each user that was added as a custodian. Each entry includes the following fields:		
id	Integer	The ZLP user ID of the custodian.
caseId	Integer	The ID of the case the custodian(s) was added to.
custodianId	Integer	The custodian ID assigned to the user.

Parameter	Type	Description
dataSourceId	Integer	The ID of the search store used to find data for the custodian.
createDate	String	The date and time the custodian was added to the case.
searchQueryBeginDate **searchQueryEndDate**	String	These fields indicate the date range for the preservation. Messages sent to - or received by - the custodian during this date range are added to the preservation.
isFutureEnabled	Boolean	If set to True, any messages ingested into ZL UA going forward that are sent to – or received by – this custodian will be put on legal hold and saved to the case during processing.
endDate	String	The end date at which future messages for the custodian will no longer be placed on legal hold.
notes	String	Additional information about the custodian.

DELETE: Delete Case (deleteCase)

Delete a case. Specify the case to be deleted by its ID.

Path

http://localhost:8080/ps/api/v1/cases/deleteCase/{caseId}

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case to be deleted. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the case was deleted successfully.

DELETE: Delete Tag (deleteTag)

Delete a tag from a case. Specify the tag to be deleted by its tag ID and the ID of the case it belongs to.

Path

http://localhost:8080/ps/api/v1/cases/deletetag/{tagId}/{caseId}

Request Parameters

Parameter	Type	Description
tagId	Integer	Specify the ID of the tag to be deleted. You can retrieve tag IDs with the GET: Get All Case Tags (getallcasetags) endpoint described on page 45.
caseId	Integer	Specify the ID of the case containing the tag to be deleted. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the case was deleted successfully.

GET: Get All Case Data Sources (getallcasedatasources)

Retrieve information for the data sources that have been added to a particular case. Specify the case by its ID.

Path

http://localhost:8080/ps/api/v1/cases/getallcasedatasources/{caseId}

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case whose data sources you want to view. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
page **pageSize**	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Data Source Entries: The response includes an entry for each data source included in the specified case. Each entry includes the following fields:		
id	Integer	The data source ID.
caseId	Integer	The ID of the case the data source run belongs to.
name	String	The internal name assigned to the data source.
displayName	String	The display name assigned to the data source.
createDate	String	The date and time the data source was created.
parentId	Integer	Reserved. This will be set to -1 for all data sources.

Schema Field	Type	Description
searchStoreId	Integer	The ID of the search store that was used for the search (i.e., the store that was searched).
purpose	String	The purpose of the data source.
dataSourceType	String	The type of data source.
lastUpdate	String	The date/time the data source was last updated.

GET: Get Case Info (getallcaseusingnamepattern)

Retrieve the configuration of a case by searching for its name.

Path

http://localhost:8080/ps/api/v1/cases/getallcaseusingnamepattern/{namePattern}

Request Parameters

Parameter	Type	Description
namePattern	String	Enter the search pattern. The search will return information for cases whose name includes (or is similar to) the search pattern.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
id	Integer	The case ID.
name	String	The name of the case.
displayName	String	The display name of the case.
createDate	String	The date and time the case was created.
fileDate	String	The file date specified for the case,
lastUpdate	String	The date and time the case was last updated,
state	Integer	The current state of the case.
ownerZlpUserId	Integer	The ZLP user ID of the user who created the case.
journalDomainId	Integer	The journal domain ID of the ZL department associated with the case.

GET: Get All Case Tags (getallcasetags)

Retrieve information for the tags that have been added to a case.

Path

http://localhost:8080/ps/api/v1/cases/getallcasetags/{caseId}/{stTagPurpose}

Request Parameters

Parameter	Type	Description
caseId	Integer	The ID of the case containing the tags you want to view. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
stTagPurpose	String	Specify the purpose of the tags you want to view (e.g., **Review**, **AutoProcess**, etc). The endpoint will return information for the selected tag and the child tags that have been added beneath it.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
id	Integer	The tag ID.
caseId	Integer	The ID of the case that the tag belongs to.
name	String	The internal name of the tag.
displayName	String	The display name of the tag.
tagFlags	Array	An array of Boolean values indicating the status of various tag attributes.
root_node	Boolean	If True, it indicates that the tag is a root-level tag.
read_only	Boolean	If True, it indicates that the tag is read-only.
enduser_tag	Boolean	If True, it indicates that the tag can be applied manually.
auto_tag	Boolean	If True, it indicates that the tag can be applied automatically, i.e., via a tag specification file.

Schema Field	Type	Description
max_tag	Boolean	If True, it indicates that the tag is a mutually exclusive tag.
tag_32	Boolean	If True, it indicates that the tag is a PII tag.
tag_64	Boolean	If True, it indicates that the tag is a content tag.
description	String	A description of the tag.
childTags: Includes an entry for each child tag that has been added beneath the specified tag. Each entry includes the following information:		
id	Integer	The child tag ID.
parentId	Integer	The ID of the tag's parent tag.
contextId	Integer	The ID of the case the tag belongs to.
name	String	The internal name of the child tag.
displayName	String	The display name of the child tag.
flags	Array	An array of Boolean values indicating the status of various attributes for the child tag.
root_node	Boolean	Indicates the child tag is the root tag.
read_only	Boolean	Indicates the child tag is read-only.
enduser_tag	Boolean	Indicates that the child tag can be applied manually.
auto_tag	Boolean	Indicates that the child tag can be applied automatically, i.e., via a tag specification file.
max_tag	Boolean	Indicates the child tag is a mutually exclusive tag.
tag_32	Boolean	Indicates the child tag is a PII tag.
tag_64	Boolean	Indicates the child tag is a content tag.
desc	String	A description of the child tag.
createDate	String	The date and time that the child tag was created.

GET: Get All Child Tags (getallchildtags)

Retrieve information for the child tags that have been beneath a parent tag. Specify the parent tag by its tag ID and case ID.

Path

http://localhost:8080/ps/api/v1/cases/getallchildtags/{caseId}/{parentId}

Request Parameters

Parameter	Type	Description
caseId	Integer	The ID of the case that the tags you want to view were added to. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
parentId	Integer	The ID of the parent tag. The endpoint return information for the child tags that have been added beneath this tag.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Child Tag Entries: Includes an entry for each child tag that has been added beneath the specified tag. Each entry includes the following information:		
id	Integer	The tag ID.
parentId	Integer	The ID of the tag's parent tag.
contextId	Integer	The ID of the case the tag belongs to.
name	String	The internal name of the tag.
displayName	String	The display name of the tag.
flags	Array	An array of Boolean values indicating the status of various tag attributes.
root_node	Boolean	If True, it indicates that the tag is a root-level tag.
read_only	Boolean	If True, it indicates that the tag is read-only.

Schema Field	Type	Description
enduser_tag	Boolean	If True, it indicates that the tag can be applied manually.
auto_tag	Boolean	If True, it indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag	Boolean	If True, it indicates that the tag is a mutually exclusive tag.
tag_32	Boolean	If True, it indicates that the tag is a PII tag.
tag_64	Boolean	If True, it indicates that the tag is a content tag.
desc	String	A description of the tag.
createDate	String	The date and time that the tag was created.

GET: Get All Cases Using Domain ID (getAllCasesUsingDomain)

Retrieve the configurations of cases that belong to a specific domain.

Path

http://localhost:8080/ps/api/v1/cases/getAllCasesUsingDomain/{idDomain}

Request Parameters

Parameter	Type	Description
idDomain	Integer	Specify a domain ID. Information for cases which are assigned to departments that belong to this domain will be returned.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Case Entries: Includes an entry for each case that belongs to the specified domain. Each entry includes the following information:		
id	Integer	The case ID.
name	String	The name of the case.
displayName	String	The display name of the case.
createDate	String	The date and time the case was created.
fileDate	String	The file date specified for the case,
lastUpdate	String	The date and time the case was last updated,
state	Integer	The current state of the case.
ownerZlpUserId	Integer	The ZLP user ID of the user who created the case.
journalDomainId	Integer	The journal domain ID of the ZL department associated with the case.

GET: Get List of All Schema (getAllSchema)

Retrieve a list of all metadata schema that has been added to the case. Custom metadata schemas help organizations add and track important information about their cases in the ZL Discovery Manager. By defining company-specific metadata fields within configured schemas, administrative users can allow privileged users to specify additional information for cases and for individual case items. These fields may be constructed for text inputs, dates, select boxes, combo boxes, check boxes, radio buttons, and multi-select boxes.

Path

http://localhost:8080/ps/api/v1/cases/getAllSchema

Request Parameters

Parameter	Type	Description
page **pageSize**	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
SchemaName	String	The name of the schema.
FieldsListData: The response includes an entry for each schema field that has been added. Each entry includes the following fields:		
FieldName	String	The name of the field.
FieldType	String	The data type of the field.
Description	String	A description of the field.
InputType	String	The field's input type (**Text**, **Password**, **Radiobox**, **Checkbox**, **Textarea**, etc).
Mandatory	Boolean	Indicates whether the field is mandatory (True) or not.

GET: Get Case Data Source Using ID (getcasedatasourceusingsrcid)

Retrieve information for a data source that has been added to a particular case. Specify the data source by its ID.

Path

http://localhost:8080/ps/api/v1/cases/getcasedatasourcesusingsrcid/{srcId}

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
srcId	Integer	Specify the ID of the data sources you want to view. You can retrieve data source IDs with the GET: Get All Case Data Sources (getallcasedatasources) endpoint described on page 42.

Response Schema Fields

Schema Field	Type	Description
id	Integer	The data source ID.
caseId	Integer	The ID of the case the data source run belongs to.
name	String	The internal name assigned to the data source.
displayName	String	The display name assigned to the data source.
createDate	String	The date and time the data source was created.
parentId	Integer	Reserved. This will be set to -1 for all data sources.
searchStoreId	Integer	The ID of the search store that was used for the search (i.e., the store that was searched).
purpose	String	The purpose of the data source.
dataSourceType	String	The type of data source.
lastUpdate	String	The date/time the data source was last updated.

GET: Get Case Using ID (getcaseusingid)

Retrieve the configuration of a case. Specify the case by its ID.

Path

http://localhost:8080/ps/api/v1/cases/getcaseusingid/{caseId}

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
id	Integer	The case ID.
name	String	The name of the case.
displayName	String	The display name of the case.
createDate	String	The date and time the case was created.
fileDate	String	The file date specified for the case.
lastUpdate	String	The date and time the case was last updated.
state	Integer	The current state of the case.
ownerZlpUserId	Integer	The ZLP user ID of the user who created the case.
journalDomainId	Integer	The journal domain ID of the ZL department associated with the case.

GET: Get Case Schema Using Case ID and Field Item Schema (getCaseSchema)

Retrieve schema information that has been added to a case. Specify the schema by its case ID value.

Path

http://localhost:8080/ps/api/v1/cases/getCaseSchema/{caseId}/{fItemSchema}

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
fItemSchema	Boolean	Select True to return the case item metadata schema that has been configured, or False to return the case schema.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
schemaName	String	The name of the file schema.
caseMetaDataField Entries: Includes an entry for each field that belongs to the specified file schema. Each entry includes the following information:		
FieldName	String	The name of the schema field.
FieldType	String	The type of the schema field.
Description	String	The description of the schema field.
InputType	String	The field's input type (**Text**, **Password**, **Radiobox**, **Checkbox**, **Textarea**, etc.).
Mandatory	Boolean	Indicates whether the field is mandatory (True) or not (False).

GET: Get Case Schema Metadata (GetCaseSchemaMetaData)

Retrieve schema information that has been added to a case. Specify the schema by its case ID value.

Path

http://localhost:8080/ps/api/v1/cases/getCaseSchemaMetaData/{caseId}

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
CaseSchemaMetaData Entries: Includes an entry for each field that belongs to the specified file schema. Each entry includes the following information:		
additionalProp1	String	Each of these fields contains key-value pairs. In this case, the key will be the schema field name of the metadata schema for the particular case, and the value will represent the value assigned to that field inside the case.
additionalProp2	String	
additionalProp3	String	

GET: Get Search Tree

Retrieve search tree information for a case.

Path

http://localhost:8080/ps/api/v1/cases/getSearchTree/{caseId}

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.

Request Body Schema Fields

None.

Response Schema Fields

The response schema includes an array that includes a `li_attr` entry for each data source that has been added to the case. Each of these entries includes a `children` array, which includes a `li_attr` child entry for each preservation search that has been saved within the data source. These entries indicate the search type, display name, search name, and status of the preservation search. For example, the following entry is for a saved preservation search named `ws_gov_res_20251027_031906`:

```
"li_attr": {
   "searchType": "DataSource",
   "displayName": "ws_gov_res_20251027_031906",
   "searchName": "126",
   "nodeType": "savedSearch"
 },
 "searchStatus": 1000,
 "text": "ws_gov_res_20251027_031906",
 "type": "DoneStatus"
}
```

PUT: Update Case Item Schema (updatecaseitemschema)

Update the case item-level schema that has been applied to a case. Once a document has been saved to a case, permitted case users can add custom metadata to the document using the case's applied item-level schema fields.

Path

http://localhost:8080/ps/api/v1/cases/updatecaseitemschema

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
schemaName	String	Enter the name of the schema.

Request Body Schema Fields

None.

Response Schema Fields

Parameter	Type	Description
caseId	Integer	The ID of the case the schema belongs to.
schemaName	String	The name of the schema.

PUT: Update Case Schema (updatecaseschema)

Update the case schema that has been applied to a case. Case schemas help organizations add and track important information about the cases created and the items saved to cases in ZL Discovery Manager. By defining company-specific metadata fields within configured schemas, administrative users can allow privileged users to specify additional information for cases and for individual case items. These fields may be constructed for text inputs, dates, select boxes, combo boxes, check boxes, radio buttons, and multi-select boxes.

Path

http://localhost:8080/ps/api/v1/cases/updatecaseschema

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern*)* endpoint described on page 44.
schemaName	String	Enter the name of the schema. You can retrieve schema names from the **CaseSchema** database table.

Request Body Schema Fields

None.

Response Schema Fields

Parameter	Type	Description
caseId	Integer	The ID of the case the schema belongs to.
schemaName	String	The name of the schema.

PUT: Update Case Schema Metadata (updateCaseSchemaMetaData)

Update metadata schema that has been applied to a case. Custom metadata schemas help organizations add and track important information about their cases in the ZL Discovery Manager. By defining company-specific metadata fields within configured schemas, administrative users can allow privileged users to specify additional information for cases and for individual case items. These fields may be constructed for text inputs, dates, select boxes, combo boxes, check boxes, radio buttons, and multiselect boxes.

Path

http://localhost:8080/ps/api/v1/cases/updateCaseSchemaMetaData

Request Parameters

Schema Field	Type	Description
idCase	**Integer**	Enter the case ID. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
FieldData Entries: Each entry includes the following information:		

additionalProp1	String	Use these fields to specify key-value pairs identifying the metadata fields and values. In this case, the key will be the schema field name of the metadata schema for the particular case, and the value will represent the value assigned to that field inside the case. For example:
		`{`
		`"idCase": 20,`
		`"FieldData": {`
		`"name1": "test",`
		`"name2": "123"`
		`}`

additionalProp2	String	
additionalProp3	String	

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
CaseSchemaMetaData Entries: Includes an entry for each field that belongs to the specified file schema. Each entry includes the following information:		
additionalProp1	String	Each of these fields contains key-value pairs. In this case, the key will be the schema field name of the metadata schema for the particular case, and the value will represent the value assigned to that field inside the case.
additionalProp2	String	
additionalProp3	String	

PUT: Update Tag (updatetag)

Update a tag's display name and description.

Path

http://localhost:8080/ps/api/v1/cases/updatetag

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
tagDisplayName	String	Specify the tag's display name.
tagDescription	String	Specify a description of the tag.
caseId	Integer	The ID of the case the tag to be updated belongs to. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
tagId	Integer	The ID of the tag to be updated. You can retrieve tag IDs with the GET: Get All Case Tags (getallcasetags) endpoint described on page 45.

Response Schema Fields

Schema Field	Type	Description
parent	String	The name of the tag's parent tag.
id	Integer	The tag ID.
caseId	Integer	The ID of the case the tag belongs to.
parentId	Integer	The ID of the tag's parent tag.
name	String	The internal name of the tag.
displayName	String	The display name of the tag.
tagFlags	Array	An array of Boolean values indicating the status of various tag attributes.

Schema Field	Type	Description
root_node	Boolean	If True, it indicates that the tag is a root-level tag.
read_only	Boolean	If True, it indicates that the tag is read-only.
enduser_tag	Boolean	If True, it indicates that the tag can be applied manually.
auto_tag	Boolean	If True, it indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag	Boolean	If True, it indicates that the tag is a mutually exclusive tag.
tag_32	Boolean	If True, it indicates that the tag is a PII tag.
tag_64	Boolean	If True, it indicates that the tag is a content tag.
description	String	A description of the tag.
createDate	String	The date and time that the tag was created.

Discovery/Privileges

In the ZL Discovery Manager, privileges can also be assigned within each eDiscovery case to grant users access to perform operations within the case. A user's role may provide access to operations not allowed by their privileges - and vice versa - so it is important to remember that each user will have access to all operations allowed by his assigned role(s) and privilege(s).

There are four possible privileges:

- **Case Manager**: Permission to administer cases including managing case information, custom metadata, tags, privileges, custodians, and case indexes. Permission to browse, search, and take action - e.g. tag or export - on documents collected or preserved in the case.
- **Case Administrator:** Permission to administer cases including managing case information, custom metadata, tags, privileges, custodians, and case indexes.
- **Reviewer**: Permission to browse, search, and take action - e.g. tag, or export - on documents collected or preserved in the case.
- **Searches**: Permission to search the items that have been added to the case.

The following sections describe the Discovery/Privileges endpoints available in the REST API. Use these endpoints to grant and manage privileges within a case:

- *DELTE: Delete All Privileges Within a Case (deleteallprivileges)*
- *DELTE: Delete a User's Privileges Within a Case (deleteprivilege)*
- *GET: Get User Privileges (getuserprivilege)*
- *PUT: Update User Privileges (updateuserprivilege)*

DELTE: Delete All Privileges Within a Case (deleteallprivileges)

Delete all user privileges that have been assigned within a case.

Path

http://localhost:8080/ps/api/v1/caseprivileges/deleteallprivileges/{caseId}

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case you want to update. All user privileges that have been assigned within this case will be revoked. You can retrieve case IDs with the GET: Get Case Info Using Name Pattern endpoint. For information, *GET: Get Case Info (getallcaseusingnamepattern)* on page 44.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the user privileges were deleted successfully.

DELTE: Delete a User's Privileges Within a Case (deleteprivilege)

Delete the privileges that have been assigned to a specific user within a case.

Path

http://localhost:8080/ps/api/v1/caseprivileges/deleteprivilege/{idZlpUser}/{caseId}

Request Parameters

Parameter	Type	Description
idZlpUser	Integer	Specify the ID of the user whose privileges are to be revoked.
caseId	Integer	Specify the ID of the case you want to update. You can retrieve case IDs with the GET: Get Case Info Using Name Pattern endpoint. For information, *GET: Get Case Info (getallcaseusingnamepattern)* on page 44.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the user's privileges were deleted successfully.

GET: Get User Privileges (getuserprivilege)

Retrieve information regarding the privileges that have been assigned to a user within a particular case. Specify the case by its ID and the user by their ZLP user ID.

Path

http://localhost:8080/ps/api/v1/caseprivileges/getuserprivilege/{idZlpUser}/{caseId}

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
idZlpUser	Integer	Specify the ZLP user ID of the user you want to view.
caseId	Integer	Specify the ID of the case you want to view. You can retrieve case IDs with the GET: Get Case Info Using Name Pattern endpoint. For information, *GET: Get Case Info (getallcaseusingnamepattern)* on page 44.

Response Schema Fields

Schema Field	Type	Description
zlpUserID	Integer	The ZLP ID of the user.
fullName	String	The name of the user.
emailAddress	String	The user's primary email address.
privileges	Array	An array of Boolean values indicating which privileges the user has been assigned.
reviewPrivileges	Boolean	If True, the user has permission to browse, search, and take action - e.g., tag, export, remove legal hold - on documents collected or preserved in the case.
searchesPrivileges	Boolean	If True, the user has permission to search the contents of the case.

Schema Field	Type	Description
caseManagerPrivileges	Boolean	If True, the user has permission to administer cases including managing case information, custom metadata, tags, privileges, custodians, and case indexes. This also includes permission to browse, search, and take action - e.g., tag, export, remove legal hold- on documents collected or preserved in the case.
administratorPrivileges	Boolean	If True, the user has permission to administer cases including managing case information, custom metadata, tags, privileges, custodians, and case indexes.

PUT: Update User Privileges (updateuserprivilege)

Update the user privileges that have been assigned to a user within a particular case. Specify the case by its ID and the user by their ZLP user ID.

Path

http://localhost:8080/ps/api/v1/caseprivileges/updateuserprivilege

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
reviewerPrivileges	Boolean	Set to True to grant the user permission to browse, search, and take action - e.g., tag, export, remove legal hold - on documents collected or preserved in the case.
searchesPrivileges	Boolean	Set to True to grant the user permission to search the contents of the case.
caseManagerPrivileges	Boolean	Set to True to grant the user permission to administer cases including managing case information, custom metadata, tags, privileges, custodians, and case indexes. This also includes permission to browse, search, and take action - e.g., tag, export, remove legal hold- on documents collected or preserved in the case.
administratorPrivileges	Boolean	Set to True to grant the user permission to administer cases including managing case information, custom metadata, tags, privileges, custodians, and case indexes.
zlpUserId	Integer	Specify the ZLP user ID of the user you want to view.
caseId	Integer	Specify the ID of the case you want to view. You can retrieve case IDs with the GET: Get Case Info Using Name Pattern endpoint. For information, *GET: Get Case Info (getallcaseusingnamepattern)* on page 44.

Response Schema Fields

Schema Field	Type	Description
zlpUserID	Integer	The ZLP ID of the user.
fullName	String	The name of the user.
emailAddress	String	The user's primary email address.
privileges	Array	An array of Boolean values indicating which privileges the user has been assigned.
reviewPrivileges	Boolean	If True, the user has permission to browse, search, and take action - e.g., tag, export, remove legal hold - on documents collected or preserved in the case.
searchesPrivileges	Boolean	If True, the user has permission to search the contents of the case.
caseManagerPrivileges	Boolean	If True, the user has permission to administer cases including managing case information, custom metadata, tags, privileges, custodians, and case indexes. This also includes permission to browse, search, and take action - e.g., tag, export, remove legal hold- on documents collected or preserved in the case.
administratorPrivileges	Boolean	If True, the user has Permission to administer cases including managing case information, custom metadata, tags, privileges, custodians, and case indexes.

Discovery/Custodians

Custodians are targets of an eDiscovery investigation who might have knowledge about the location of documents that are pertinent to a case. In ZL Discovery Manager, custodians function as a way in which aliases and email addresses can be associated with an individual to facilitate searches.

Once an individual has been added as a case custodian, case users can use the custodian as a filter when searching for documents to save to the case, and when searching across documents that have already been saved to the case.

It is also possible to create a custodian preservation: an automatic preservation placed on all mail documents related to a custodian (i.e., messages sent to the custodian or received by the custodian) over a specified time frame. Custodian preservations can be enacted within a case during case creation and/or after the case has been created.

The following sections describe the Discovery/Custodians endpoints available in the REST API. Use these endpoints to manage custodians within a case:

- *POST: Add Custodian (addcustodian)*
- *PUT: Add Custodian Alias (addalias)*
- *GET: Get All Aliases (getallaliases)*
- *GET: Get All Custodians (getallcustodians)*
- *GET: Get All Custodians Using ZLP User ID (getallcutodiansusingzlpuserid)*
- *GET: Get Custodian Preservation Using Custodian ID (getcustodianpreservationusingid)*
- *GET: Get All Custodian Preservations (getAllCustodianPreservation)*
- *GET: Get Custodian Using Address (getusingaddress)*
- *GET: Get Custodian Using ID (getusingid)*
- *DELETE: Delete Custodian (delete)*
- *DELETE: Delete Custodian Alias (deletealias)*
- *PUT: Update Custodian (updatecustodian)*

POST: Add Custodian (addcustodian)

Add a custodian to a case. You can add a user that has already been added to ZL UA, or you can add a new user.

Path

http://localhost:8080/ps/api/v1/custodian/addcustodian

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case the custodian should be added to. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
type	Integer	Specify any integer value.
zlpUserId	Integer	If the custodian already exists in ZL UA, specify their ZLP user ID.
Configure the remaining fields if the custodian has **not been** added to ZL UA. If the user has already been added to ZL UA, you can leave these fields empty.		
fTerminated	Boolean	Set to True if the custodian should be set to terminated status.
fullName	String	Specify the full name of the custodian.
address	String	Specify the primary email address of the custodian.
externalReference	String	Use this optional field to store any unique IDs used outside of ZL UA.
miscellaneous	String	Enter any miscellaneous information about the custodian.
fAddAlias	Boolean	Set to True to create a new alias for the custodian.

Response Schema Fields

Parameter	Type	Description
id	Integer	The custodian ID assigned to the user.
caseId	Integer	Specify the ID of the case the custodian should be added to.

Parameter	Type	Description
zlpUserId	Integer	The ZLP user ID of the new custodian.
isTerminated	Boolean	If True, the user has been terminated.
address	String	Specify the primary email address of the custodian.
externalReference	String	Use this optional field to store any unique IDs used outside of ZL UA.
fullName	String	Specify the full name of the custodian.
createDate	String	The date the custodian was created.
misc1	String	Miscellaneous information about the custodian.
custodianType	Integer	The custodian type.

PUT: Add Custodian Alias (addalias)

Add an alias to a custodian.

Path

http://localhost:8080/ps/api/v1/custodian/addalias

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodian to be updated. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
custodianId	Integer	Specify the ID of the custodian to be updated. You can retrieve custodian IDs with the GET: Get All Custodians (getallcustodians) endpoint described on page 76.
aliasType	Integer	Specify the alias type: • 0: Default (Email) • 1: X500 DN • 2: Address • 3: Manual • 4: Exchange Legacy DN • 5: Transformed Lotus DN • 6: NetBios User Name • 100: IM • 200: Bloomberg • 300: Parlano
alias	String	Specify the new alias.
createDate	String	Not required.

Response Schema Fields

Parameter	Type	Description
caseId	Integer	The ID of the case containing the custodian that has been updated. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
custodianId	Integer	The ID of the custodian that has been updated.
aliasType	Integer	The alias type: • 0: Default (Email) • 1: X500 DN • 2: Address • 3: Manual • 4: Exchange Legacy DN • 5: Transformed Lotus DN • 6: NetBios User Name • 100: IM • 200: Bloomberg • 300: Parlano
alias	String	The new alias.
createDate	String	The alias creation date.

GET: Get All Aliases (getallaliases)

Retrieve all aliases that have been added to a case for a particular custodian.

Path

http://localhost:8080/ps/api/v1/custodian/getallaliases/{caseId}/{custodianId}

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodian to be viewed. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
custodianId	Integer	Specify the ID of the custodian to be viewed. For information on retrieving custodian IDs, refer to *GET: Get All Custodians (getallcustodians)* on page 76.
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Response Schema Fields

Parameter	Type	Description
Custodian Alias Entries: The response includes an entry for each alias that has been added to the case for this custodian. Each entry includes the following information:		
caseId	Integer	The case ID.
custodianId	Integer	The custodian ID.

Parameter	Type	Description
aliasType	Integer	The alias type: • 0: Default (Email) • 1: X500 DN • 2: Address • 3: Manual • 4: Exchange Legacy DN • 5: Transformed Lotus DN • 6: NetBios User Name • 100: IM • 200: Bloomberg • 300: Parlano
alias	String	The new alias.
createDate	String	The alias creation date.

GET: Get All Custodians (getallcustodians)

Retrieve information for all custodians that have been added to a case.

Path

http://localhost:8080/ps/api/v1/custodian/getallcustodians/{caseId}

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodians to be viewed. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Schema Fields

None.

Response Schema Fields

Parameter	Type	Description
Custodian Alias Entries: The response includes an entry for each custodian that has been added to the case. Each entry includes the following fields:		
id	Integer	The custodian ID assigned to the user.
caseId	Integer	The ID of the case the custodian has been added to.
zlpUserId	Integer	The ZLP user ID of the custodian.
isTerminated	Boolean	If True, the user has been terminated.
address	String	The primary email address of the custodian.
externalReference	String	This optional field is used to store any unique IDs used outside of ZL UA.

Parameter	Type	Description
fullName	String	The full name of the custodian.
createDate	String	The date the custodian was created.
misc1	String	Miscellaneous information about the custodian.
custodianType	Integer	The custodian type.

GET: Get All Custodians Using ZLP User ID (getallcutodiansusingzlpuserid)

Retrieve information for a custodian. Specify the custodian by their user ID.

Path

http://localhost:8080/ps/api/v1/custodian/getallcustodianusingzlpuserid/{zlpUserId}

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to grant roles to. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Schema Fields

None.

Response Schema Fields

Parameter	Type	Description
id	Integer	The custodian ID assigned to the user.
caseId	Integer	The ID of the case the custodian has been added to.
zlpUserId	Integer	The ZLP user ID of the custodian.
isTerminated	Boolean	If True, the user has been terminated.
address	String	Specify the primary email address of the custodian.
externalReference	String	Use this optional field to store any unique IDs used outside of ZL UA.
fullName	String	Specify the full name of the custodian.

Parameter	Type	Description
createDate	String	The date the custodian was created.
misc1	String	Miscellaneous information about the custodian.
custodianType	Integer	The custodian type.

GET: Get Custodian Preservation Using Custodian ID (getcustodianpreservationusingid)

A custodian preservation in ZL UA is an automatic preservation placed on all mail documents related to a custodian (i.e., messages sent to the custodian or received by the custodian) over a specified time frame. Custodian preservations can be enacted within a case during case creation and/or after the case has been created. Use this endpoint to view a custodian preservation.

Path

http://localhost:8080/ps/api//v1/custodian/getcustodianpreservationusingid/{idCustodianPreservation}

Request Parameters

Parameter	Type	Description
idCustodianPreservation	Integer	Specify the preservation ID. You can retrieve preservation IDs with the GET: Get All Custodian Preservations (getallcustodianpreservation) endpoint described on page 82.

Request Schema Fields

None.

Response Schema Fields

Parameter	Type	Description
id	Integer	The preservation ID.
caseId	Integer	The ID of the case the preservation has been added to.
custodianId	Integer	The ID of the custodian the preservation belongs to.
zlpUserId	Integer	The ZLP user ID of the custodian.
dataSourceId	Integer	The data source ID of the preservation.
searchStoreId	Integer	The ID of the search store that was used for the preservation (i.e., the store that was searched).
createDate	String	The date and time that the preservation was created.
searchQueryBeginDate **searchQueryEndDate**	String	These fields specify the date range for the preservation. Messages sent to - or received by - the custodian during this date range are added to the preservation.

Parameter	Type	Description
isFutureEnabled	Boolean	If set to True, any messages ingested into ZL UA going forward that were sent to – or received by – this custodian will be put on legal hold and saved to the case during processing.
endDate	String	If **isFutureEnabled** is set to True, this indicates the date and time that custodian-related messages for this custodian will no longer be saved to the case.
isMarkedForDeletion	Boolean	If True, it indicates that the preservation is marked for deletion.
notes	String	Additional information about the preservation.

GET: Get All Custodian Preservations (getallcustodianpreservation)

A custodian preservation in ZL UA is an automatic preservation placed on all mail documents related to a custodian (i.e., messages sent to the custodian or received by the custodian) over a specified time frame. Custodian preservations can be enacted within a case during case creation and/or after the case has been created. Use this endpoint to view the custodian preservations that have been added to a case.

Path

http://localhost:8080/ps/api/v1/custodian/getallcustodianpreservation/{idCase}

Request Parameters

Parameter	Type	Description
idCase	Integer	Specify the ID of the case whose custodian preservations you want to view. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Schema Fields

None.

Response Schema Fields

Parameter	Type	Description
Custodian Preservation Entries: The response includes an entry for each custodian preservation that has been added to the case. Each entry includes the following fields:		
id	Integer	The preservation ID.
caseId	Integer	The ID of the case the preservation has been added to.
custodianId	Integer	The ID of the custodian the preservation belongs to.
zlpUserId	Integer	The ZLP user ID of the custodian.

Parameter	Type	Description
dataSourceId	Integer	The data source ID of the preservation.
searchStoreId	Integer	The ID of the search store that was used for the preservation (i.e., the store that was searched).
createDate	String	The date and time that the preservation was created.
searchQueryBeginDate **searchQueryEndDate**	String	These fields specify the date range for the preservation. Messages sent to - or received by - the custodian during this date range are added to the preservation.
isFutureEnabled	Boolean	If set to True, any messages ingested into ZL UA going forward that were sent to – or received by – this custodian will be put on legal hold and saved to the case during processing.
endDate	String	If **isFutureEnabled** is set to True, this indicates the date and time that custodian-related messages for this custodian will no longer be saved to the case.
isMarkedForDeletion	Boolean	If True, it indicates that the preservation is marked for deletion.
notes	String	Additional information about the preservation.

GET: Get Custodian Using Address (getusingaddress)

Retrieve information for a custodian that has been added to a case. Specify the custodian to be viewed by their email address.

Path

http://localhost:8080/ps/api/v1/custodian/getusingaddress/{caseId}/{address}

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodian to be viewed. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
address	String	Specify the custodian's primary email address.

Request Schema Fields

None.

Response Schema Fields

Parameter	Type	Description
id	Integer	The custodian ID assigned to the user.
caseId	Integer	The ID of the case the custodian should be added to.
zlpUserId	Integer	The ZLP user ID of the custodian.
isTerminated	Boolean	If True, the user has been terminated.
address	String	Specify the primary email address of the custodian.
externalReference	String	An optional field to store any unique IDs used outside of ZL UA.
fullName	String	The full name of the custodian.
createDate	String	The date the custodian was created.
misc1	String	Miscellaneous information about the custodian.
custodianType	Integer	The custodian type.

GET: Get Custodian Using ID (getusingid)

Retrieve information for a custodian that has been added to a case. Specify the custodian to be viewed by their custodian ID.

Path

http://localhost:8080/ps/api/v1/custodian/getusingid/{caseId}/{custodianId}

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodian to be viewed. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
custodianId	Integer	Specify the custodian ID. You can retrieve custodian IDs with the GET: Get All Custodians (getallcustodians) endpoint described on page 76.

Request Schema Fields

None.

Response Schema Fields

Parameter	Type	Description
id	Integer	The custodian ID assigned to the user.
caseId	Integer	The ID of the case the custodian was added to.
zlpUserId	Integer	The ZLP user ID of the custodian.
isTerminated	Boolean	If True, the user has been terminated.
address	String	The primary email address of the custodian.
externalReference	String	An optional field to store any unique IDs used outside of ZL UA.
fullName	String	The full name of the custodian.
createDate	String	The date the custodian was created.
misc1	String	Miscellaneous information about the custodian.
custodianType	Integer	The custodian type.

DELETE: Delete Custodian (delete)

Remove a custodian from a case.

Path

http://localhost:8080/ps/api/v1/custodian/delete/{caseId}/{custodianId}

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodian to be removed. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
custodianId	Integer	Specify the ID of the custodian to be removed. You can retrieve custodian IDs with the GET: Get All Custodians (getallcustodians) endpoint described on page 76.

Request Schema Fields

None.

Response Schema Fields

A string indicating whether the custodian was removed successfully.

DELETE: Delete Custodian Alias (deletealias)

Delete a custodian alias.

Path

http://localhost:8080/ps/api/v1/custodian/deletealias

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodian to be updated. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
custodianId	Integer	Specify the ID of the custodian to be updated. You can retrieve custodian IDs with the GET: Get All Custodians (getallcustodians) endpoint described on page 76.
aliasType	Integer	Specify the alias type that is to be deleted: • 0: Default (Email) • 1: X500 DN • 2: Address • 3: Manual • 4: Exchange Legacy DN • 5: Transformed Lotus DN • 6: NetBios User Name • 100: IM • 200: Bloomberg • 300: Parlano
alias	String	Specify the alias.
createDate	String	Not required.

Response Schema Fields

A string indicating whether the custodian alias was deleted successfully.

PUT: Update Custodian (updatecustodian)

Update a custodian's basic configuration, such as their full name.

Path

http://localhost:8080/ps/api/v1/custodian/updatecustodian

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case the custodian belongs to. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
custodianId	Integer	Specify the custodian ID of the custodian to be updated. You can retrieve custodian IDs with the GET: Get All Custodians (getallcustodians) endpoint described on page 76.
fullName	String	Specify the full name of the custodian.
externalReference	String	Use this optional field to store any unique IDs used outside of ZL UA.
miscellaneous	String	Enter any miscellaneous information about the custodian.

Response Schema Fields

Parameter	Type	Description
id	Integer	The custodian ID assigned to the user.
caseId	Integer	The ID of the case the custodian has been added to.
zlpUserId	Integer	The ZLP user ID of the new custodian.
isTerminated	Boolean	If True, the user has been terminated.
address	String	The primary email address of the custodian.
externalReference	String	An optional field to store any unique IDs used outside of ZL UA.
fullName	String	The full name of the custodian.

Parameter	Type	Description
createDate	String	The date the custodian was created.
misc1	String	Miscellaneous information about the custodian.
custodianType	Integer	The custodian type.

UAA/Departments

In ZL UA, a department is defined as a hierarchical grouping of users. Each user created in ZL UA is added into a department that has been defined within ZL UA. The structure of the departmental configured in ZL UA often resembles the organization's actual departmental structure, but this configuration model is not required.

The highest level of this department hierarchy is the ROOT department, beneath which all other departments are created. Every department created beneath the ROOT department will inherit the compliance settings of the ROOT department by default.

A department can have one or more sub-departments beneath it in the department hierarchy. These are referred to as the department's *child departments.* Each child department inherits the compliance settings of its parent department by default. However, an administrative user with the proper permissions can configure a department's compliance settings to diverge from those of its parent department. Each user recognized in the system must be associated with one (and only one) department.

The following sections describe the UAA/Departments endpoints available in the REST API. Use these endpoints to create, manage and update departments:

- *POST: Create Child Department (createchilddepartment):* Create a department.
- *DELETE: Delete Department (deleteDepartment)*: Remove a department.
- *GET: Get Child Departments (getallchilddepartment):* Retrieve the configuration of a child department.
- *GET: Get All Departments Using Name (getalldeptsusingnamepattern):* Retrieve the configuration of a department(s) whose name matches a search pattern.
- *GET: Get Users of a Department (getallusers):* Retrieve a list of the users that have been added to a department.
- *GET: Get Department Using ID (getdepartmentusingid):* Retrieve the configuration of a department specified by its ID.
- *GET: Get Department By Name (getdepartmentusingname):* Retrieve the configuration of a department specified by its name.
- *GET: Get Department Policy Information Using ID (getpolicy):* Retrieve information regarding the default compliance policies that have been assigned to a department.
- *PUT: Update Department (updatedepartment):* Update the configuration of a department.
- *PUT: Update Parent Department (updateparentdepartment):* Update the parent department assigned to a department.

POST: Create Child Department (createchilddepartment)

Create a child department.

Path

http://localhost:8080/ps/api/v1/Department/createchilddepartment

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
name	String	**MANDATORY**. Type the internal name of the new department. Internal department names cannot be re-used, even after a department has been deleted. If you delete a department and need to re-create it later, you could set the new department's display name to match the deleted department's display name. This should make the new department's purpose clear to users who are browsing the departmental hierarchy in the UAA module.
displayName	String	**MANDATORY**. Type an identifying name for the new department. This name will be displayed whenever the departmental hierarchy is displayed in the UAA module.
domainName	String	Enter a domain name here to create a new domain. Leave this field blank if the domain name is the same as the department name.
parentServerName	String	Enter the name of the department's parent department. A child department inherits the compliance settings of its parent department, unless those settings are overwritten manually.
classifierName	String	**MANDATORY**. Enter the name of the lexicon to be assigned to the new department. Each department can have only one lexicon assigned to it. Lexicons must be created with the Compliance Manager module.
externalReference	String	Use this optional field to store any unique IDs used outside of ZL UA.
deptTags	String	Input to this field is not required.

Schema Field	Type	Description
reviewEscalationDepts	String	Specify the name(s) of the Review Escalation Department that should be used when reviewers assigned to this department are unable to make a review decision on a particular message. These reviewers will have the option to escalate the message, effectively passing it on to the Review Escalation Department for a review decision. If there is more than one Review Escalation Department, enter the names as a comma-separated list.
miscField1 **miscField2**	String	Use these optional fields to type any additional notes for the department being created.

Response Schema Fields

Schema Field	Type	Description
id	String	The department ID.
name	String	The department internal name.
displayName	String	The department display name.
parentDepartmentId	String	The ID of the parent department.
domain	String	The department domain name.
parentServerName	String	The name of the parent department.
domainId	String	The ID of the department's domain.
journalDomain	String	The journal domain assigned to the department.
journalDomainId	String	The ID of the journal domain assigned to the department.
createDate	String	The date and time the department was created.
lastUpdate	String	The date and time the department's configuration was last updated.
externalReference	String	An optional field used to store any unique IDs used outside of ZL UA.
departmentTags	String	List any department tags that have been applied to the department.
miscField1 **miscField2**	String	Any miscellaneous information entered when the department was created.

Schema Field	Type	Description
classifierName	String	The name of the lexicon assigned to the department.
reviewEscalationDepts	String	The name of the Review Escalation Departments assigned to the department.
isProcessingCenter	Boolean	Indicates whether the department is configured as a processing center.

DELETE: Delete Department (deleteDepartment)

Delete a department.

Path

http://localhost:8080/ps/api/v1/Department/deletedepartment/{DeptId}

Request Parameters

Parameter	Type	Description
deptId	Integer	Specify the ID of the department to be deleted. The REST API includes several endpoints you can use to retrieve department IDs and configuration data. For example, you can use the GET: Get All Departments Using Name (getalldeptsusingnamepattern) endpoint described on page 96 to retrieve a department's configuration by searching for its name.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the department was deleted successfully.

GET: Get Child Departments (getallchilddepartment)

Retrieve the configurations of a department's child departments. Identify the department by its ID.

Path

http://localhost:8080/ps/api/v1/Department/getallchilddepartment/{deptId}

Request Parameters

Parameter	Type	Description
deptId	Integer	Specify the ID of the department whose child department configurations you want to view. The REST API includes several endpoints you can use to retrieve department IDs and configuration data. For example, you can use the GET: Get All Departments Using Name (getalldeptsusingnamepattern) endpoint described on page 96 to retrieve a department's configuration by searching for its name.
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

The response schema includes an entry for each child department that is found. The fields included in each child department entry are the same as those included in the response returned after creating a department. They represent the configuration of the child department. For descriptions of these fields, refer to *POST: Create Child Department* on page 91.

GET: Get All Departments Using Name (getalldeptsusingnamepattern)

Retrieve the configuration of a department. Identify the department by its name.

Path

http://localhost:8080/ps/api/v1/Department/getalldeptsusingnamepattern/{deptNamePattern}

Request Parameters

Parameter	Type	Description
deptNamePattern	Integer	Enter the search pattern. The search will return information for departments whose name includes (or is similar to) the search pattern.
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

The response schema includes an entry for each department that is found. The fields included in each department entry are the same as those included in the response returned after creating a department. They represent the configuration of the child department. For descriptions of these fields, refer to *POST: Create Child Department* on page 91.

GET: Get Users of a Department (getallusers)

Retrieve a list of the users that have been added to a department.

Path

http://localhost:8080/ps/api/v1/Department/getallusers/{departmentName}

Request Parameters

Parameter	Type	Description
departmentName	Integer	Specify the name of the department whose users you want to view. The REST API includes several endpoints you can use to retrieve department names and other configuration data. For example, you can use the GET: Get All Departments Using Name (getalldeptsusingnamepattern) endpoint described on page 96 to search for department information by specifying a search pattern. The endpoint will return information for departments whose names match (or are similar to) the search pattern.
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Department Users: Includes an entry for each user included in the specified department. The schema fields included in each entry are described below.		
idZlpUser	Integer	The ZL user ID assigned to the user.
type	Integer	The user type: • 0: User • 100: User group/mailing list
address	String	The user's primary email address.

Schema Field	Type	Description
owner	String	The user's owner. This is meant to identify the user's manager or the user's creator.
extReference	String	Unique ID or information used outside of ZL UA that is relevant to the user.
userTags	String	User tags applied to the user.
retags	String	Retention tags applied to the user.
altReviewDepts	String	The Alternate Review Department(s) the user is assigned to.
deptName	String	The department the user is assigned to.
reviewDeptName	String	The Review Department the user is assigned to.
mailServerName	String	The mail server for this user.
mailStoreInfo	String	Mail store information for this user.
syncExclude	Boolean	Indicates whether the user should be excluded (True) from User Synchronizations or not (False).
archive **journal**	Boolean	Set to True if the user is available for archiving and journaling, respectively.
fullName	String	The first and last name of the user.
dateCreate	String	The date and time the user was created.
dateLastUpdate	String	The date and time the user last updated.
connectUserId	String	The user ID used to connect to the user's mail server.
dateHire	String	The date and time the user was hired.
dateTerminated	String	The date and time the user was terminated, if applicable.
terminated	Boolean	Indicates whether the user has been Terminated (True) or not.

Schema Field	Type	Description
dateIterStart	String	These fields indicate that date and time that the last user synchronization process started and ended, and the date and time that the user's information was updated during synchronization.
dateIterEnd	String	
dateIterUpdate	String	
dateFullScanStart	String	The date and time that the last full scan of the user's mailbox began.
dateFullScanEnd	String	The date that the last full scan of the user's mailbox ended.
dateArchiveBegin	String	The date and time that archiving of the user's date began.
miscField1 **miscField2**	String	Additional information entered for the user.

GET: Get Department Using ID (getdepartmentusingid)

Retrieve the configuration of a department. Identify the department by its ID.

Path

http://localhost:8080/ps/api/v1/Department/getdepartmentusingid/{deptId}

Request Parameters

Parameter	Type	Description
deptId	Integer	Enter the ID of the department to view. The REST API includes several endpoints you can use to retrieve department IDs and configuration data. For example, you can use the GET: Get All Departments Using Name (getalldeptsusingnamepattern) endpoint described on page 96 to retrieve a department's configuration by searching for its name.

Request Body Schema Fields

None.

Response Schema Fields

The response schema fields returned for the specified department are the same as those included in the response returned after creating a department. They represent the configuration of the department. For descriptions of these fields, refer to *POST: Create Child Department* on page 91.

GET: Get Department By Name (getdepartmentusingname)

Retrieve the configuration of a department. Identify the department by its name.

Path

http://localhost:8080/ps/api/v1/Department/getdepartmentusingname/{deptName}

Request Parameters

Parameter	Type	Description
deptName	String	Enter the name of the department whose configuration is to be retrieved.

Request Body Schema Fields

None.

Response Schema Fields

The response schema fields returned for the specified department are the same as those included in the response returned after creating a department. They represent the configuration of the department. For descriptions of these fields, refer to *POST: Create Child Department* on page 91.

GET: Get Department Policy Information Using ID (getpolicy)

View the policies that have been assigned to a department. Identify the department by its ID.

Path

http://localhost:8080/ps/api/v1/Department/getpolicy/{deptId}/{stPolicyType}

Request Parameters

Parameter	Type	Description
deptId	Integer	Enter the ID of the department to view. The REST API includes several endpoints you can use to retrieve department IDs and configuration data. For example, you can use the GET: Get All Departments Using Name (getalldeptsusingnamepattern) endpoint described on page 96 to retrieve a department's configuration by searching for its name.
stPolicyType	String	Select the policy you want to view (e.g., Archiving, Stubbing, etc).

Request Body Schema Fields

None.

Response Schema Fields

The following fields are returned for each policy specified in the input.

Parameter	Type	Description
policyName	String	The policy name.
policyDesc	String	A description of the policy.
policyType	String	The policy type.
id	Integer	The ID assigned to the policy.
createDate	String	The date and time the policy was created.
policiesRules: Includes an entry for each policy rule that has been added to the policy. Each entry includes the following fields:		
policyId	Integer	The policy ID.

Parameter	Type	Description
policyRuleId	Integer	The policy rule ID.
policyRuleName	String	The name of the policy rule (required).
policyRuleDesc	String	A description of the policy rule.
policyRuleActionCode	Integer	The action code for the policy rule. This identifies the action to be taken upon files the policy is applied to.
listPolicyRuleConditions: Includes an entry for each condition that has been added to the policy rule. Each file or message is checked against these conditions to determine if the policy rule and the associated action will be applied. Each entry includes the following fields: • **Field:** The content field to check against the condition. • **Subfield:** The content subfield to check against the condition. • **Operator:** The operator defines how the content field and sub-field will be checked against the condition. • **Pattern:** The pattern to check against the content field and sub-field. For example, if the **Field** is set to **Has Attachment** and the **Operator** to is true, the condition would check for messages or files that have attachments.		
policyRuleConditionDesc	String	A concise description of all the conditions of a policy rule. For example, a policy rule written to apply to text files that were last accessed later than October 8, 2024 would appear as follows: `"IF File Last Accessed Date Later Than '2024/10/08 ' AND File Type Equals (Match case) '.txt' "`
policyRulesActionDisplay	String	A description of the action type assigned to the policy rule.

policyActionParams	String	This includes the action parameters configured for the policy rule. For example:
		`"policyActionParams": {`
		`"Archive Action": "0",`
		`}`

PUT: Update Department (updatedepartment)

Update a department's configuration.

Path

http://localhost:8080/ps/api/v1/Department/updatedepartment/{deptId}

Request Parameters

Parameter	Type	Description
deptId	Integer	Specify the ID of the department to be updated. The REST API includes several endpoints you can use to retrieve department IDs and configuration data. For example, you can use the GET: Get All Departments Using Name (getalldeptsusingnamepattern) endpoint described on page 96 to retrieve a department's configuration by searching for its name.

Request Body Schema Fields

Schema Field	Type	Description
name	String	Enter the internal name of the department. Internal department names cannot be re-used, even after a department has been deleted. If you delete a department and need to re-create it later, you could set the new department's display name to match the deleted department's display name. This should make the new department's purpose clear to users who are browsing the departmental hierarchy in the UAA module.
displayName	String	**MANDATORY**. Type the identifying name for the department. This name will be displayed whenever the departmental hierarchy is displayed in the UAA module.
domainName	String	Enter the name of the department's domain.
parentServerName	String	Enter the name of the department's parent department. A child department inherits the compliance settings of its parent department, unless those settings are overwritten manually.
classifierName	String	Enter the name of the lexicon to be assigned to the department. Each department can have only one lexicon assigned to it. Lexicons must be created with the Compliance Manager module.

Schema Field	Type	Description
externalReference	String	Use this optional field to store any unique IDs used outside of ZL UA.
deptTags	String	Input to this field is not required.
reviewEscalationDepts	String	Specify the name of the Review Escalation Department that should be used when reviewers assigned to this department are unable to make a review decision on a particular message. These reviewers will have the option to escalate the message, effectively passing it on to the Review Escalation Department for a review decision.
miscField1 **miscField2**	String	Use these optional fields to type any additional notes for the department being created.

Response Schema Fields

The **Response Schema Fields** returned represent the updated configuration of the department. This is the same set of fields as the **Request Schema Fields** described earlier.

PUT: Update Parent Department (updateparentdepartment)

Update the parent department assigned to a department.

Path

http://localhost:8080/ps/api/v1/Department/updateparentdepartment/{deptId}/{parentId}

Request Parameters

Parameter	Type	Description
deptId **parentId**	Integer	Specify the ID of the department to be updated as the **deptId** value, and the ID of the department to be assigned as its parent as the **parentId** value. The REST API includes several endpoints you can use to retrieve department IDs and configuration data. For example, you can use the GET: Get All Departments Using Name (getalldeptsusingnamepattern) endpoint described on page 96 to retrieve a department's configuration by searching for its name.

Request Body Schema Fields

None.

Response Schema Fields

The response schema fields returned for the department are the same as those included in the response returned after creating a department. They represent the configuration of the department. For descriptions of these fields, refer to *POST: Create Child Department* on page 91.

Discovery/Reports

The following sections describe the Discovery/Reports endpoints available in the REST API. Use these endpoints to download the following reports:

- **Download Audit Trail Report**: Lists the actions performed within a case and indicates which user performed each action.
- **Download Custodian Preservation Hits Report:** Indicates how many messages were returned for each custodian in the preservations included in a particular case and search store.
- **Download Case Custodian Report:** Presents details of on the custodians that have been added to a case.
- **Download Export Task Report:** Presents detailed information on every export action taken within the case.
- **Download Global Case Report:** Presents details of all the cases within your assigned departmental scope. It includes information such as the case name, case ID, case mode, case filing date, docket number, and more.
- **Download Global Custodian Report:** Presents details of all the custodians that have been added to cases within your assigned departmental scope. It includes information such as the custodian's name and email address and the names and IDs of the cases the custodian has been added to.
- **Download User Entitlement Report:** Summarizes the roles and role scopes that have been granted to users in the ZL Discovery Manager application.
- **Download Consolidated Reports of Collection, Preservation and Analysis:** Summarizes the collections, preservations and analysis searches that have been added to a case.
- **Download Custodian Preservation Report:** Summarizes the custodian preservations that have been added to cases.
- **Download Global Survey Report:** A consolidated report of all the surveys across all the cases accessible to you in ZL Discovery Manager.
- **Download Keyword Expansion Report:** Indicates how many messages within the search store contain variations of the keyword(s) that have been entered as search criteria into the ECA search, providing early visibility into search results.
- **Download Preservation Notification Status Report:** This report summarizes the status of each preservation notification. It includes information such as the notification name and ID, the case name, and the creation date for each preservation notification.
- **Download Search Term Report:** Summarizes the search terms used to conduct a search.
- **Download Workflow Audit Term Report:** Presents details on either Investigation Request Workflow actions or on Case Survey-related actions taken in the ZL Discovery Manager application.

POST: Download Audit Report (downloadAuditReport)

Download an audit report for a case. This lists the actions performed within the case, and indicates which user performed each action. The report will be downloaded to the local Downloads directory.

Path

http://localhost:8080/ps/api/v1/discoveryreports/downloadAuditReport

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the case ID. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
category **action**	String	Specify the category and action type you want to view. This includes the following options: • **View All Case Actions:** Set both fields to **ACTION_ANY** to return all available actions that have been performed within the case. • **View A Subset of Case Actions:** Configure each field to specify a specific category or action you want to view. Refer to the following section, *Categories and Actions,* for a list of the strings you can apply to the **category** and **action** fields. For example, you could set **category** to **tag** and **action** to **ACTION_TAG_CREATED** to view audit trail data for tag creation operations. Or you could set **category** to **tag** and **action** to **ACTION_ANY** to view audit trail data for all tagging operations. You can use the remaining fields to filter the report by the dates on which the actions occurred.
dateMode	String	Set this to any of the following values: • **1day**: Actions performed within the last day. • **1week**: Actions performed within the last week. • **1month**: Actions performed within the last 30 days. • **custom**: Actions performed within a specific date range.

Parameter	Type	Description
startDate **endDate**	String	Use these fields to specify the start date and end date for the report (inclusive) when the **dateMode** is set to **custom**. Audit trail data for this date range will be included in the report.

Categories and Actions

Category Field	Available Action Field Values **Set to ACTION_ANY to view all available action types for a specific category, or choose a specific action type below.**
case: View case management actions.	ACTION_CASE_CREATED ACTION_CASE_UPDATED ACTION_CASE_DELETED ACTION_CASE_SUBSCRIBE ACTION_CASE_UNSUBSCRIBE ACTION_CASE_VIEWED
item: View actions performed upon items within a case.	ACTION_ITEM_CREATED ACTION_ITEM_VIEWED ACTION_ITEM_UPDATED ACTION_ITEM_DELETED ACTION_ITEMS_EXPORTED ACTION_ITEM_REVIEWED ACTION_ITEM_NOT_REVIEWED ACTION_ITEM_IMPORTED ACTION_ITEM_LEGAL_HOLD_APPLY ACTION_ITEM_LEGAL_HOLD_REMOVE ACTION_ITEM_TAG_APPLY ACTION_ITEM_TAG_REMOVE ACTION_ITEM_REDACTION_ADDED ACTION_ITEM_REDACTION_REMOVED ACTION_ITEM_INLINE_ANNOTATED ACTION_ITEM_DOWNLOADED_REDACTED ACTION_ITEM_PDF_PREVIEW ACTION_ITEM_DOWNLOADED ACTION_ITEM_ASSIGN_REVIEWER ACTION_ITEM_UNASSIGN_REVIEWER
tag: View tagging actions	ACTION_TAG_CREATED ACTION_TAG_DELETED

Category Field	Available Action Field Values Set to ACTION_ANY to view all available action types for a specific category, or choose a specific action type below.
custodian: View actions related to custodian management	ACTION_CUSTODIAN_CREATED ACTION_CUSTODIAN_UPDATED ACTION_CUSTODIAN_DELETED ACTION_CUSTODIAN_ALIAS_CREATED ACTION_CUSTODIAN_ALIAS_DELETED ACTION_CUSTODIAN_LEGALHOLD_CREATED ACTION_CUSTODIAN_LEGALHOLD_DELETED ACTION_CUSTODIAN_LEGALHOLD_UPDATED
member: View actions related to the members of a case	ACTION_CASE_MEMBER_ADDED ACTION_CASE_MEMBER_UPDATED ACTION_CASE_MEMBER_DELETED
search: View actions related to search activity.	ACTION_SEARCH_DATA ACTION_CASE_SEARCH ACTION_SEARCH_CASESEARCH_UPDATE ACTION_SEARCH_DATASOURCE_CREATED ACTION_SEARCH_DATASOURCE_UPDATED ACTION_SEARCH_DATASOURCE_DELETED ACTION_SEARCH_BACKGROUND_TASK

Response Schema Fields

None.

POST: Download Custodian Preservation Hit Report (downloadcustodianpreservationhitreport)

Download a Custodian Preservation Hit Report. This report indicates how many messages were returned for each custodian in the preservations included in a particular case and search store. The report will be downloaded to the local Downloads directory.

Path

http://localhost:8080/ps/api/v1/discoveryreports/downloadcustodianpreservationhitreport/{caseId}/{storeId}

Request Parameters

Parameter	Type	Description
caseId	Integer	The ID of the case the preservation belongs to. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
storeId	Integer	The ID of the search store that was used to create the preservation (i.e., the store that was searched). You can extract search store IDs from the **SearchStore** database table.

Request Body Schema Fields

None.

Response Schema Fields

None.

POST: Download Case Custodian Report (downloadCustodianReport)

Download a case custodian report. The case custodian report is a consolidated report that presents details of all the custodians added to the case within their assigned departmental scope. The report will be downloaded to the local **Downloads** directory.

Path

http://localhost:8080/ps/api/v1/discoveryreports/downloadcustodianreport/{caseId}

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the case ID. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.

Request Body Schema Fields

None.

Response Schema Fields

None.

POST: Download Export Task Report (downloadExportTaskReport)

Download an export task report for a case. This report presents detailed information on the export actions taken within the case. The report will be downloaded to the local Downloads directory.

Path

http://localhost:8080/ps/api/v1/discoveryreports/downloadExportTaskReport

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
idCase	Integer	Specify the case ID.
dateStart **dateEnd**	String	The start date and end date for the report (inclusive). Export task data for this date range will be included in the report.

Response Schema Fields

None.

POST: Download Global Case Report (downloadGlobalCaseReport)

Download the global case report: a consolidated report that presents details of all the cases within your assigned departmental scope. It includes information such as the case name, case ID, case mode, case filing date, docket number, and more. The report will be downloaded to the local **Downloads** directory.

Path

http://localhost:8080/ps/api/v1/discoveryreports/downloadglobalcasereport

Request Parameters

None.

Request Body Schema Fields

None.

Response Schema Fields

None.

POST: Download Global Custodian Report (downloadGlobalCustodianReport)

Download the global custodian report: a consolidated report that presents details of all the custodians that have been added to cases within your assigned departmental scope. It includes information such as the custodian's name and email address and the names and IDs of the cases the custodian has been added to. The report will be downloaded to the local Downloads directory.

Path

http://localhost:8080/ps/api/v1/discoveryreports/downloadglobalcustodianreport

Request Parameters

None.

Request Body Schema Fields

None.

Response Schema Fields

None.

POST: Download User Entitlement Report (downloadUserEntitlementReport)

Download the user entitlement report, which summarizes the roles and role scopes that have been granted to users in the ZL Discovery Manager application. The report will be downloaded to the local Downloads directory.

Path

http://localhost:8080/ps/api/v1/discoveryreports/downloaduserentitlementreport

Request Parameters

None.

Request Body Schema Fields

None.

Response Schema Fields

None.

POST: Download Consolidated Reports of Collections, Preservations and Analysis (downloadConsolidatedReports)

Download a report summarizing the collections, preservations and/or analysis searches that have been added to a case. The report will be downloaded to the local Downloads directory.

Path

http://localhost:8080/ps/api/v1/discoveryreports/downloadconsolidatedreports/{caseId}/{reportPurpose}

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the case ID. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
reportPurpose	String	Specify the data sources you want to view in the report. Select **Collections**, **Preservations**, **Analysis** or **All.**

Request Body Schema Fields

None.

Response Schema Fields

None.

POST: Download Custodian Preservation Report (downloadCustodianPreservationReport)

Download a custodian preservation report, which summarizes the custodian preservations that have been added to cases. The report will be downloaded to the local Downloads directory.

Path

http://localhost:8080/ps/api/v1/discoveryreports/downloadcustodianpreservationreport

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
caseMode	Integer	Specify the case mode. Enter **any** to search any case. Enter **only** and specify a case ID(s) with the **caseIds** field to search a specific case.
caseIds	Integer	An array of case IDs to specify the cases for which you want to view custodian preservation data. Specify the case IDs as a comma-separated list. For example: `"caseIds": [ 0,1,2,3 ]` You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
srchField	String	Use this field to specify how the search will be performed: • **fullName:** Search by user name. • **alias:** Search by user alias. • **owner:** Search by owner.
srchPattern	String	Enter a search pattern to locate the custodians of interest.
dateEnd	String	Specify the end date for the report.

Response Schema Fields

None.

POST: Download Global Survey Report (downloadGlobalSurveyReport)

Download a global survey report, a consolidated report of all the surveys or preservation notifications across a specific case, or across all accessible cases. The report will be downloaded to the local Downloads directory.

Path

http://localhost:8080/ps/api/v1/v1/discoveryreports/downloadglobalsurveyreport

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
caseMode	Integer	Specify the case mode. Enter **any** to search any case. Enter **only** and specify a case ID with the **caseId** field to search a specific case.
type	String	Enter **caseSurveys** to view the details of the case surveys, or **preservation** to view the details of preservation notifications.
caseId	Integer	Specify the case ID. This is applicable when the **caseMode** is set to **only.** You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
surveyIds	Integer	An array of survey IDs you want to view. Specify the case IDs as a comma-separated list. For example: `"surveyIds": [ 0,1,2,3 ]`

Response Schema Fields

None.

POST: Download Keyword Expansion Report (downloadKeywordExpansionReport)

Download a keyword expansion report. This report indicates how many messages within the search store contain variations of the keyword(s) that have been entered as search criteria into the ECA search, providing early visibility into search results. You can choose which terms should be included in the ECA search from the results. The report will be downloaded to the local Downloads directory.

Path

http://localhost:8080/ps/api/v1/discoveryreports/downloadkeywordexpansionreport

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
caseId	Integer	The ID of the case that you want to search. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
searchStoreId	Integer	The ID of the search store. You can extract search store IDs from the **SearchStore** database table.
fields	String	An array specifying the message components to search. You can use the following values: **msgSubject**, **msgBody**, **msgAttachment**, **msgAttachmentNames**, and **msgAttachmentChildPath**. Specify the field names as a comma-separated list. For example, to search the message subject and message body, you would enter: `"fields": [ msgSubject, msgBody ]` Leave this blank to view all survey data for the specified case.
pattern	String	Enter the pattern to search for. The report will indicate how many items within the specified search store contain this pattern in the specified fields.

Response Schema Fields

None.

POST: Download Preservation Notification Status Report (downloadPreservationNotificationStatusReport)

Download a preservation notification status report, which summarizes the status of each preservation notification. It includes information such as the notification name and ID, the case name, and the creation date for each preservation notification. The report will be downloaded to the local Downloads directory.

Path

http://localhost:8080/ps/api/v1v1/discoveryreports/downloadpreservationnotificationstatusreport

Request Parameters

None.

Request Body Schema Fields

None.

Response Schema Fields

None.

POST: Download Search Terms Report (downloadSearchTermReport)

Download a search term report, which summarizes the search terms used to conduct a search. The report will be downloaded to the local Downloads directory.

Path

http://localhost:8080/ps/api/v1/discoveryreports/downloadsearchtermreport/{caseID}/{searchID}

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the case ID. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
searchID	Integer	Specify the search (data source) ID. You can use the GET: Get All Case Data Sources (getallcasedatasources) endpoint described on page 42 to retrieve the ID values of all data sources that have been added to a particular case.

Response Schema Fields

None.

POST: Download Workflow Audit Report (downloadWorkflowAuditReport)

Download a workflow audit report for a case. This presents details on either Investigation Request Workflow actions or on Case Survey-related actions taken in the ZL Discovery Manager application. The report will be downloaded to the local Downloads directory.

Path

http://localhost:8080/ps/api/v1/discoveryreports/downloadworkflowauditreport

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
taskType **subTaskType**	String	Use the **taskType** field to specify the task type: • **investigation:** Audit trail data for the Investigation Request Workflow. • **survey:** Audit trail data for Case Surveys. Use the **subTaskType** field to specify the action type: • **created:** Audit trail data for items being created. • **updated:** Audit trail data for items being updated.
idCase	Integer	Specify the case ID. You can retrieve case IDs with the GET: Get Case Info (getallcaseusingnamepattern) endpoint described on page 44.
dateStart **dateEnd**	String	The start date and end date for the report (inclusive). Audit trail data for this date range will be included in the report.

Response Schema Fields

None.

Discovery/Roles

In the Discovery Manager, a role is an application or department-level set of permissions that determines what actions users can perform in the application. Roles can be assigned globally, or for a specific department(s). For example, a Global Discovery Manager role would enable the user's assigned role for all cases. A Discover Manager role for a specific department would restrict the user's role to the cases defined within that department. The department a case belongs to is defined during case setup.

The following sections describe the Discovery/Roles endpoints available in the REST API. Use these endpoints to grant and manage roles within a case:

- *PUT: Grant User Roles (grantroles)*
- *PUT: Revoke User Roles (revokeroles)*

PUT: Grant User Roles (grantroles)

Grant roles to a user.

Path

http://localhost:8080/ps/api/v1/discoveryroles/grantroles

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to grant roles to. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
User Roles: The remaining input is an array of fields defining the roles to be granted. Specify the following for each **userRole** entry:		
roleId	Integer	The role ID. For information on retrieving role IDs, refer to the following sections: • *GET: Get All Custom Roles* on page 370 • *GET: Get All System Roles* on page 372
scope	String	Specify whether the role should be granted globally so that it is applicable to all departments, or if it should be granted to specific departments only: • **Global**: Global • **InclRecur**: On selected departments recursively • **Incl**: On selected departments only
allScopeDomainIds	Integer	An array of domain IDs to specify the departments the role is applicable to for roles that are only granted on selected departments. These can be retrieved from the **ArchiveServer** database table. Specify the domain IDs as a comma-separated list. For example: `"AllScopeDomainIds": [ 0,1,2,3 ]`

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each role specified in the request.		
success	Boolean	Indicates whether the role was granted successfully (True) or not (False).
result	String	The result of the request. The string will indicate how the role has been applied (on which departments, scope, role ID, etc).
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

PUT: Revoke User Roles (revokeroles)

Revoke roles that have been previously assigned to a user.

Path

http://localhost:8080/ps/api/v1/discoveryroles/revokeroles

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to modify. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
roleIds	Integer	An array of role IDs to specify the roles to be revoked. Specify the role IDs as a comma-separated list. For example: `"Roles": [ 0,1,2,3 ]` For information on retrieving role IDs, refer to the following sections: • *GET: Get All Custom Roles* on page 370 • *GET: Get All System Roles* on page 372

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each role specified in the request.		
success	Boolean	Indicates whether the role was revoked successfully (True) or not (False).
result	String	The result of the request. If successful, a message will display indicating that the role has been revoked.
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

EFM/Audits

This section describes endpoints you can use to view audit trail report data for EFM projects. This data summarizes the actions that have been performed within the project (e.g., tagging and project management operations):

- *GET: Get Audit Action Params (getAuditActionParams)*
- *POST: Get Project Audit Trail Data (getProjectAuditTrail)*

GET: Get Audit Action Params (getAuditActionParams)

Use this endpoint to retrieve EFM action codes. The action codes are required as input when you use the POST: Get Project Audit Trail Data endpoint described in the next section to view audit trail data. They determine what type of actions and operations will be included in the audit trail data.

Path

http://localhost:8080/ps/api/v1/efm/audits/getauditactionparams/{auditType}

Request Parameters

Parameter	Type	Description
auditType	String	Specify the type of action you want to view the action codes for: • **project:** Actions related to project management (e.g., project created or project viewed). • **tagconfig:** Tagging actions (e.g., Record Category or Tag applied). • **disposition:** Actions related to the Disposition Workflow (e.g., Disposition Run Created). • **audit:** Audit operations (e.g., audit trail viewed).

Request Body Schema Fields

None.

Response Schema Fields

Parameter	Type	Description
actionCode	Integer	The action code for the sub-type.
actionDesc	String	A description of the action sub-type.

POST: Get Project Audit Trail Data (getProjectAuditTrail)

View audit trail report data for an EFM project. The audit trail data summarizes the actions that have been performed within the project. You can customize the request data to create an audit trail report that includes all actions performed within the project, or a report that includes specific types of actions (e.g., tagging and project management operations). You can also filter the request to display actions that occurred within a specific date range.

Path

http://localhost:8080/ps/api/v1/efm/audits/getprojectaudittrail

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter for this endpoint defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

Parameter	Type	Description
projectId	Integer	Specify the ID of the project for which you want to view audit trail data. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Parameter	Type	Description
actionType	String	Specify the action type for the report: • **any:** All actions performed within the project. • **project:** Actions related to project management (e.g., project created or project viewed). • **tagconfig:** Tagging actions (e.g., Record Category or Static Tag applied). • **disposition:** Actions related to the Disposition Workflow (e.g., Disposition Run Created). • **audit:** Audit operations (e.g., audit trail viewed). You can specify an action sub-type with the **actionCode** parameter to view a more specific set of operations.
actionCode	String	Specify an action code to identify the action sub-type to be included in the report. This is used in conjunction with the **actionType** field to allow you to generate audit trail data for specific sets of operations. For example, if **actionType** is set to **tagconfig**, you could specify an action code to view audit trail data for Record Categories or Static Tags being applied to files. Use the GET: Get Audit Action Params endpoint described in the previous section to view the valid action codes for each action type. You can also use the default value of **-1** as the action code to view all operations for the selected **actionType**. Use **-1** when the **actionType** is set to **any.**
dateStart **dateEnd**	String	These fields are applicable when the **mode** field is set to **dates,** meaning that the report will include actions performed within a specific date range. Use them to specify the start date and end date (inclusive) for the report.
mode	String	Set this to any of the following values to determine the date range for the report: • **1day**: Actions performed within the last day. • **1week**: Actions performed within the last week. • **1month**: Actions performed within the last 30 days. • **dates**: Actions performed within a specific date range (i.e., the range specified by the **dateStart** and **dateEnd** parameters).

Response Schema Fields

The response includes an entry for each action taken upon the project that matches the request schema. Each entry includes the following information describing the action.

Parameter	Type	Description
projectName	String	The name of the project upon which the action was taken.
action	String	The action taken upon the project.
datePerformed	String	The date and time the action was performed.
user	String	The user who performed the action.
itemName **displayName**	Integer	The internal and display names of the object upon which the action was taken. For example, if a Record Category was applied to the file, this would specify the file name.
sourceIP	String	The IP address of the machine from which the user performed the action.
destinationIP	String	The IP address of the ZL server running the EFM module.
details	String	Additional information about the action. This varies depending on the action. For example, if a Record Category was applied to the file, this would specify the Record Category ID.

EFM/Disposition

Disposition is the process by which files whose records management lifecycle has expired are removed from ZL UA. The Disposition Workflow includes the following steps:

- Disposition is enabled in a project. When enabling the Disposition Workflow, you will also schedule Disposition Runs. During the Disposition Run, files whose lifecycle have expired are permanently deleted from ZL UA and the source server.

  You can also create manual (ad-hoc) Disposition Runs for projects that do not have scheduling enabled for the Disposition Workflow (or if you want to add an additional Disposition Run to a project with scheduling enabled).

- Before each scheduled Disposition Run, the system determines which files in the project are eligible for disposition and creates a Disposition Approval Requests for those files. The request may require administrative approval before the Disposition Run can start (depending on the project configuration). The approval process includes an option to exclude specific files from a Disposition Run, so that the excluded files will not be removed from ZL UA and the source server.

- After you approve the Disposition Approval Request, you can execute the Disposition Run.

The following sections describe the EFM/Disposition endpoints available in the REST API. Use these endpoints to manage the Disposition Workflow*:*

- *PUT: Approve Disposition Run (approveDisposition)*
- *PUT: Cancel Disposition Run (cancelDispositionRun)*
- *POST: Create Manual Disposition Run (createAdHocDisposition)*
- *POST: Create Disposition Rollover Run (createDispositionRolloverRun)*
- *PUT: Disable Disposition (disableDisposition)*
- *PUT: Enable Disposition (enableDisposition)*
- *PUT: Exclude Items from Disposition Run (excludeItemsFromDisposition)*
- *POST: Get Disposition Run Information (getDispositionRunsUsingProjectId)*
- *GET: Get Disposition Run Status (getDispositionTaskStatus)*
- *POST: Get Expired Files List (getExpiryFileItems)*
- *GET: Get Projects Pending Approval (getprojectspendingforapproval)*
- *GET: Get Projects Ready For Disposition (getprojectsreadyfordisposition)*
- *PUT: Re-include all Excluded Items (reIncludeAllExcludedItems)*
- *PUT: Disable Exclusion of Items (disableExclusion)*
- *POST: Start Disposition Run (runDisposition)*

PUT: Approve Disposition Run (approveDisposition)

Approve a Disposition Run.

Path

http://localhost:8080/ps/api/v1/efm/disposition/approvedisposition

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project containing the Disposition Run to be approved. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
runId	Integer	The ID of the Disposition Run to be approved. You can use the POST: Get Disposition Run Information (getDispositionRunsUsingProjectId) endpoint described on page 142 to retrieve Disposition Run IDs. For manual (ad-hoc) Disposition Runs, enter -1.
approvalComment	String	Enter a comment to be logged with the approval.

Response Schema Fields

The response schema includes a string indicating whether the Disposition Run was approved successfully.

PUT: Cancel Disposition Run (cancelDispositionRun)

Cancel a Disposition Run.

Path

http://localhost:8080/ps/api/v1/efm/disposition/canceldispositionrun/{projectId}/{dispositionRunId}

Request Parameters

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project containing the Disposition Run to be aborted. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
dispositionRunId	Integer	The ID of the Disposition Run to be aborted. You can use the POST: Get Disposition Run Information (getDispositionRunsUsingProjectId) endpoint described on page 142 to retrieve Disposition Run IDs. For manual (ad-hoc) Disposition Runs, enter -1.

Request Body Schema Fields

None.

Response Schema Fields

The response schema includes a string indicating whether the Disposition Run was aborted successfully.

POST: Create Manual Disposition Run (createAdHocDisposition)

Create a manual (ad-hoc) Disposition Run.

Path

http://localhost:8080/ps/api/v1/efm/disposition/createadhocdisposition

Request Parameters

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project you want to add the Disposition Run to. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
dateStart **dateEnd**	String	Specify the start date and end date for the Disposition Run. Files whose lifecycle expired during this date range will be included in the Disposition Run.

Request Body Schema Fields

None.

Response Schema Fields

The response includes a string indicating whether the manual Disposition Run was created successfully.

POST: Create Disposition Rollover Run (createDispositionRolloverRun)

Create a rollover Disposition Run. Rollover runs may be required when the number of files to be removed in a Disposition Run exceeds the maximum of 1 million items.

Path

http://localhost:8080/ps/api/v1/efm/disposition/createdispositionrolloverrun/{projectId}/{dispositionRunId}

Request Parameters

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project you want to add the Disposition Run to. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
dispositionRunId	String	The ID of the Disposition Run (for scheduled Disposition Runs). You can use the POST: Get Disposition Run Information (getDispositionRunsUsingProjectId) endpoint described on page 142 to retrieve Disposition Run IDs.

Request Body Schema Fields

None.

Response Schema Fields

The response includes a string indicating whether the rollover Disposition Run was created successfully.

PUT: Disable Disposition

Disable the Disposition Workflow on a project.

Path

http://localhost:8080/ps/api/v1/efm/disposition/disabledisposition/{projectId}

Request Parameters

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project you want to modify. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

The response includes a string indicating whether the Disposition Workflow was disabled successfully.

PUT: Enable Disposition

Enable the Disposition Workflow on a project.

Path

http://localhost:8080/ps/api/v1/efm/disposition/enabledisposition/{projectId}

Request Parameters

	Type	Description
projectId	Integer	Specify the ID of the project you want to modify. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

The response includes a string indicating whether the Disposition Workflow was enabled successfully.

PUT: Exclude Items from Disposition Run (excludeItemsFromDisposition)

Exclude items from a Disposition Run for a project, so that they will not be removed when the next Disposition Run is executed.

Path

http://localhost:8080/ps/api/v1/v1/efm/disposition/excludeitemfromdispostion/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project modify. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
fExclude	Boolean	Set to True to exclude the specified files from the Disposition Run, or False to include them. Please note that all files whose lifecycle expired during the date range specified for a Disposition Run will be included by default.

Request Schema Fields

Schema Field	Type	Description
List of File IDs	Array	A comma-separated list of IDs for the files you want to exclude. You can use the POST: Get Expired Files List endpoint described on page 146 to retrieve file IDs of files that whose lifecycle has expired within a project.

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each role specified in the request.		
success	Boolean	Indicates whether the role was granted successfully (True) or not (False).
result	String	The result of the request. The string will indicate how the role has been applied (on which departments, scope, role ID, etc.).
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

POST: Get Disposition Run Information (getDispositionRunsUsingProjectId)

Retrieve information for the Disposition Runs for a project. The response to the call includes information such as the name of the user who approved the Disposition Run and any comments entered when approving the request.

Path

http://localhost:8080/ps/api/v1/v1/efm/disposition/getdispositionrunsusingprojectid

Request Parameters

Parameter	Type	Description
page **pageSize**	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
filter	String	Enter a string to identify the type of the Disposition Run you want to retrieve information for. This defaults to **pending**, which means that information for Disposition Runs within the project that are either pending approval or have been approved and are ready to be executed. Other values include: • **expiredRequests**: Expired Disposition Runs. These are Disposition Runs which expired before they were executed. • **done**: Disposition Runs that were executed previously. These are also referred to as past runs. • **all**: All the Disposition Runs for the project.

Request Body Schema Fields

Schema Field	Type	Description
items	Integer	Enter a comma-separated list of project IDs to specify the projects you want to view. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Response Schema Fields

The response schema includes the following information for each Disposition Run. The Disposition Runs will be returned in ascending order and grouped by project ID, with the most recent Disposition Runs for each project being listed first.

Schema Field	Type	Description
projectId	Integer	The project ID.
runId	Integer	The Disposition Run ID.
approvalComment	String	Comments entered by the user who approved the Disposition Run.
statusMsg	String	A string describing the current status of the Disposition Run.
approverFullName	String	The name of the user who approved the Disposition Run.
approverAddress	String	The email address of the user who approved the Disposition Run.
dateStart **dateEnd**	String	The start date and end date for the Disposition Run. Files whose lifecycle expired during this date range are included in the Disposition Run.
approved	Boolean	Indicates whether the Disposition Run has been approved.

GET: Get Disposition Run Status (getDispositionTaskStatus)

Obtain the status of a Disposition Run.

Path

http://localhost:8080/ps/api//efm/disposition/getDispositionTaskStatus/{projectId}/{idDisposition}

Request Parameters

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project containing the Disposition Run to be viewed. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
idDisposition	Integer	The ID of the Disposition Run whose status is to be checked. You can use the POST: Get Disposition Run Information (getDispositionRunsUsingProjectId) endpoint described on page 142 to retrieve Disposition Run IDs.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
idDisposition	Integer	The ID of the Disposition Run whose status is being viewed.
idProj	Integer	The ID of the project containing the Disposition Run whose status is being viewed.
approvalRequestId	Integer	The ID value assigned to the approval request for the Disposition Run.
periodEnd	String	The date that the retention period for the Disposition Run ended. This is typically the last day of the month in which the Disposition Run was created.
idRetBdb	Integer	The retention ID assigned to the Disposition Run.

Schema Field	Type	Description
dateStart **dateUpdate** **dateEnd**	String	The date and time that the Disposition Run started and ended, and the date and time that its status was last updated.
idStatus	Integer	The ID code assigned to the Disposition Run's current status.
statusMsg	String	A message indicating the status of the Disposition Run. For example, it will report "Approval Request Pending" if the Disposition Run has not been approved.
pid	String	The process ID assigned to the Disposition Run.

POST: Get Expired Files List (getExpiryFileItems)

Retrieve the list of expired files for a project. These are the files whose lifecycle has expired and will be included in the next Disposition Run.

Path

http://localhost:8080/ps/api//v1/efm/disposition/getexpiryfileitems

Request Parameters

Parameter	Type	Description
page **pageSize**	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project from which you want to retrieve the information. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
runId	String	For scheduled Disposition Runs, enter the ID of the Disposition Run ID. You can use the POST: Get Disposition Run Information (getDispositionRunsUsingProjectId) endpoint described on page 142 to retrieve Disposition Run IDs. For manual (ad-hoc) Disposition Runs, enter -1.
startDate **endDate**	String	Specify a start date and end date. Information for files whose lifecycle expired during this date range will be returned. This is not required for scheduled Disposition Runs.

Response Schema Fields

Schema Field	Type	Description
itemName	String	The file name.
itemId	String	The ID assigned to the file.
projectId	Integer	The ID assigned to the project the file belongs to.
expiryDate	String	The date and time that the file's lifecycle expired.
isfExclude	Boolean	Indicates whether the file has been marked for exclusion (True) from the Disposition Run.
recordCategoryId	Integer	The ID assigned to the file's Record Category.

GET: Get Projects Pending Approval (getprojectspendingforapproval)

Retrieve the project IDs of projects that have pending Disposition Approval Requests.

Path

http://localhost:8080/ps/api/v1/efm/getprojectspendingforapproval

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
currentPage	Integer	The page being displayed.
itemsInCurrent Page	Integer	The number of items displayed on the current page.
pageSIze	Integer	The maximum number of items that can be displayed on each page.
totalItems	Integer	The number of items returned by the search.
totalPages	Integer	The number of pages required to display the search results.
Response: An array of Integer values that includes the project ID for each project with a pending Disposition Approval Request. The project ID values are returned as a comma-separated list.		

GET: Get Projects Ready For Disposition (getprojectsreadyfordisposition)

Retrieve the project IDs of projects that are ready for disposition, meaning that a Disposition Run has been approved.

Path

http://localhost:8080/ps/api/v1/efm/getprojectsreadyfordisposition

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
currentPage	Integer	The page being displayed.
itemsInCurrent Page	Integer	The number of items displayed on the current page.
pageSIze	Integer	The maximum number of items that can be displayed on each page.
totalItems	Integer	The number of items returned by the search.
totalPages	Integer	The number of pages required to display the search results.
Response: An array of Integer values that includes the project ID for each project that is ready for disposition. The project ID values are returned as a comma-separated list.		

PUT: Re-include all Excluded Items (reincludeallexcludeditems)

Administrators have the option to exclude eligible files from a Disposition Run, so that they **will not** be removed from ZL UA and the source server when the Disposition Run is executed. Use this endpoint to re-include all files that were previously excluded from a Disposition Run, so that they **will** be removed when the Disposition Run is executed.

Path

http://localhost:8080/ps/api/v1/efm/disposition/reIncludeAllExcludedItems/{projectId}

Request Parameters

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project containing the Disposition Run to be modified. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

The response schema includes a string indicating whether the items were re-included successfully.

PUT: Disable Exclusion of Items (disableexclusion)

Administrators have the option to exclude eligible files from a Disposition Run, so that they **will not** be removed from ZL UA and the source server when the Disposition Run is executed. Use this endpoint to disable the exclusion feature within a project.

Path

http://localhost:8080/ps/api/v1/efm/disposition/disableexclusion/{projectId}

Request Parameters

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project to be modified. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

The response schema includes a string indicating whether the project configuration was updated successfully.

POST: Start Disposition Run (runDisposition)

Start a Disposition Run.

Path

http://localhost:8080/ps/api/v1/efm/disposition/runDisposition/{projectId}/{idDisposition}

Request Parameters

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project containing the Disposition Run to be started. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
idDisposition	Integer	The ID of the Disposition Run to be started. You can use the POST: Get Disposition Run Information (getDispositionRunsUsingProjectId) endpoint described on page 142 to retrieve Disposition Run IDs. For manual (ad-hoc) Disposition Runs, enter -1.

Request Body Schema Fields

None.

Response Schema Fields

The response schema includes a string indicating whether the Disposition Run was started successfully. It also includes the task ID assigned to the Disposition Run.

EFM/Reports

The following sections describe the EFM/Reports endpoints available in the REST API. Use these endpoints to generate reports that include information regarding the files that are included in the EFM projects within your installation:

- *POST: Generate All Excluded Items Report (generateAllExcludedReport)*
- *POST: Generate All Project User Privileges Report (generateAllProjectUserPrivilegesReport)*
- *POST: Generate Folder Report (generateFolderReport)*
- *POST: Generate File Manifest Report (generateInPlaceFileManifestReport)*
- *POST: Generate Out of Sync Report (generateOutOfSyncReport)*
- *POST: Generate Post-Disposition Report (generateInPlaceFilePostDispositionReport)*
- *POST: Generate Record Category Search Report (generateRecordCategorySearchReport)*

POST: Generate All Excluded Items Report (generateAllExcludedReport)

Download a report summarizing the files that are eligible for disposition within your EFM projects, but were excluded by the administrator approving the Disposition Run. This report includes the following details for each excluded file:

- **File Name**
- **Project ID**
- **Folder ID**
- **File Type**
- **File Size**
- **Creation**, **Last Modified** and **Last Accessed Dates**
- **Record Category**
- **Disposition Trigger Date** and **Disposition Expiry Date** (i.e., the start and end dates of the file's retention period)
- **Disposition Approval Status**
- **User Name** (i.e., the user who excluded the file)

Path

http://localhost:8080/ps/api/v1/efm/reports/generateallexcludedreport

Request Parameters

Parameter	Type	Description
renderer	String	Select the report format (**excel** or **csv**).

Request Body Schema Fields

Schema Field	Type	Description
outputDirectory	String	Specify the directory in which the report file should be generated. Make sure to specify a valid path. For example: `C:\EFMReports`
projIds	Integer	Specify the IDs of the projects for which you want to generate the report as a comma-separated list. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Response Schema Fields

The response includes a string indicating whether the specified report was generated successfully.

POST: Generate All Project User Privileges Report (generateAllProjectUserPrivilegesReport)

Download a report summarizing the user privileges that have been granted within your EFM projects. The report will include an entry for each user who has been granted privileges within the specified projects. Each entry includes the following details for each user:

- **Project ID ad Project Name**: The ID and name assigned to the project.
- **User**: The ZLP user ID.
- **User Name**: The user name.
- **User Email**: The primary email address associated with the user.
- **Privilege List:** The privileges assigned to the user.

Path

http://localhost:8080/ps/api/v1/efm/reports/generateallprojectuserprivilegesreport

Request Parameters

Parameter	Type	Description
renderer	String	Select the report format (**excel** or **csv**).

Request Body Schema Fields

Schema Field	Type	Description
projectIds	Integer	Specify the project IDs for which you want to generate the report as a comma-separated list. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
outputDirectory	String	Specify the directory in which the report file should be generated. Make sure to specify a valid path. For example: `C:\EFMReports`

Response Schema Fields

The response includes a string indicating whether the specified report was generated successfully.

POST: Generate Folder Report (generateFolderReport)

Generate a Folder Report for an EFM project. The Folder Report includes information such as the size of the folder, the number of sub-folders and files it includes, and the Record Category Declaration Policy that has been applied to it.

Path

http://localhost:8080/ps/api/v1/efm/reports/generatefolderreport

Request Parameters

Parameter	Type	Description
renderer	String	Set to **excel** to generate the Disposition Report as an Excel spreadsheet, or **csv** to generate it as a CSV file.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project for which you want to generate the report. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
folderId	Integer	Specify the ID of the folder for which you want to generate the report. You can use the GET: Get Sub-Folders endpoint (http://localhost:8080/ps/api/v1/storage/project/{projectId}/subFolders) to retrieve information on the folders and sub-folders within a project. The **dirId** output represents the folder ID for each folder. The report will include information for the specified folder and all sub-folders.

Response Schema Fields

The response includes a link you can use to download the generated report.

POST: Generate File Manifest Report (generateInPlaceFileManifestReport)

Download reports summarizing a project's files. Each report lists the file name, folder name and path of a specific group of files within the project, as well as other file attributes such as the file type, creation and last modification date, size, record category, and disposition eligibility date. The following report types are available:

- **Manifest Report**: List all files included in the project, with an option to only include files that were last modified during a specific date range.
- **Excluded Files Report**: Lists all files that are eligible for disposition within the project but have been excluded.
- **Approved Files Report**: Lists all files that have been approved for disposition within the project.
- **Pre-Disposition Report**: Lists all files that are eligible for disposition within the project.

Path

http://localhost:8080/ps/api/v1/efm/reports/generateinplacefilemanifestreport

Request Parameters

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project for which you want to generate the report. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
reportType	Array	Select the report you want to generate.
renderer	Array	Select the report format (**excel** or **csv**).
outputDirectory	String	Specify the directory in which the report file should be generated. Make sure to specify a valid path. For example: `C:\EFMReports`

Schema Field	Type	Description
dateStart **dateEnd**	String	If the **reportType** field is set to **Manifest Report**, you can optionally use these fields to create a date range for the report, so that it would only include files that were last modified during the date range. Specify each date using the following format: YYYY/MM/DD The date range is inclusive, meaning that files that were last modified on the dates specified as the start and end dates would be included in the report.
includeACL	Boolean	Set to True to include ACL details for each file in the report.

Request Body Schema Fields

None.

Response Schema Fields

The response includes a string indicating whether the specified report was generated successfully.

POST: Generate Out of Sync Report (generateOutOfSyncReport)

Download a report summarizing the out of sync files and folders in a project. A file or folder will become out of sync if you manually tag it or mark it as a record and it is subsequently re-named or deleted. In this case, the instance of the tag or record created in the database for the file or folder will be orphaned. This is because the database only includes the original path of the file or folder, and can't tell if it has been deleted or re-named.

The Out of Sync report includes general information about each out of sync file or folder, such as its path when it was tagged or marked as a record, its internal directory ID, any retention codes or tags that were applied to it, and any record schema field values for files and folders that have been marked as records.

Path

http://localhost:8080/ps/api/v1/efm/reports/generateoutofsyncreport

Request Body Schema Fields

Schema Field	Type	Description
renderer	String	Specify the report format (**excel** or **csv**).
projId	Integer	Specify the ID of the project for which you want to generate the report. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
fManualOnly	Boolean	Set this field to True to only include information for Out of Sync files and folders that were manually tagged in the report. This means that if a folder policy rule was used to apply a Record Category to a folder and it became Out of Sync, the report would only include information for the folder. It would not include information for the files within the folder because they were not manually tagged. Set this field to False to include information for all Out of Sync files and folders, regardless of whether they were manually tagged or tagged via folder policy rule.

Response Schema Fields

The response includes a link you can use to download the generated report.

POST: Generate Post-Disposition Report (generateInPlaceFilePostDispositionReport)

Download a Post-Disposition Report. The Post Disposition Report lists the file name, folder name and path of the files that were disposed of during a Disposition Run, as well as other file attributes such as the file name and path, folder name, extension and size, record category, retention code and disposition eligibility date.

Path

http://localhost:8080/ps/api/v1/efm/reports/generateinplacefilepostdispositionreport

Request Parameters

Schema Field	Type	Description
projectId	Integer	Specify the ID of the project for which you want to generate the report. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
idDispRun	Integer	The ID of the Disposition Run for which the report should be generated. You can use the POST: Get Disposition Run Information (getDispositionRunsUsingProjectId) endpoint described on page 142 to retrieve Disposition Run IDs.
renderer	Array	Select the report format (**excel** or **csv**).
outputDirectory	String	Specify the directory in which the report file should be generated. Make sure to specify a valid path. For example: `C:\EFMReports`

Request Body Schema Fields

None.

Response Schema Fields

The response includes a string indicating whether the specified report was generated successfully.

POST: Generate Record Category Search Report (generateRecordCategorySearchReport)

Generate a Record Category Search Report. You will specify a list of project IDs and a Record Category in the input to this endpoint (alternatively, you can choose to generate a report for all projects). The Record Category Search Report will be generated as an Excel spreadsheet that lists the files that the Record Category has been applied to within the specified group of projects.

Each file entry within the report includes the following details:

- **File Name:** The name of the file.
- **Project ID:** The ID of the project the file belongs to.
- **Folder ID:** The ID of the folder the file is located in.
- **File Type:** The file type (e.g., OFFICE or IMAGE).
- **File Size:** The size of the file, in KB.
- **Create Date, Last Modified Date, Last Accessed Date:** The dates the file was created, last modified and last accessed. The dates are displayed in GMT.
- **Record Category:** The Record Category assigned to the file.
- **Disposition Trigger Date, Disposition Expiry Date:** The Disposition Expiry Date also known as the file's retention date. This is the date at which the file's retention period expires and it will become eligible for disposition. The Disposition Trigger Date is the date at which the file's retention period began.
- **Disposition Flags:** A description of the file's disposition status (e.g., "Disposition Ready").

Path

http://localhost:8080/ps/api/v1/efm/reports/generaterecordcategorysearchreport

Request Parameters

Parameter	Type	Description
renderer	String	Set to **excel** to generate the Disposition Report as an Excel spreadsheet, or **csv** to generate it as a CSV file.

Request Body Schema Fields

Schema Field	Type	Description
projectIds	String	Specify the ID of the projects for which you want to generate the report as a comma-separated list. If you do not specify any project IDs, the report will include information for all projects. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
recordCategory	String	The name of the Record Category you want to generate the report for. You can retrieve Record Category names with the POST: Get Record Categories (getRecordCategories) endpoint described on page 199.
outputDir	String	Specify the directory in which the report file should be generated. Make sure to specify a valid path. For example: `C:\\EFMReports`

Response Schema Fields

The response includes a string indicating whether the specified report was generated successfully.

EFM/Tasks

This section describes endpoints you can use to view the status of EFM tasks to apply or remove Record Categores:

- *POST: Get Recategorization Runs (getRecategorizationRuns)*
- *POST: Get Recategorization Task Status (getRecategorizationTaskStatus)*

POST: Get Recategorization Runs (getRecategorizationRuns)

View the status of all recategorization runs (i.e., tasks to apply or remove Record Categories) that have been executed within a specific project.

Path

http://localhost:8080/ps/api/v1/efm/tasks/getrecategorizationruns/{projectId}

Request Parameters

Schema Field	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	
projectId	Integer	Specify the ID of the project you want to view recategorization run status for. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

The endpoint returns the following fields for each recategorization run that was executed within the specified project.

Schema Field	Type	Description
taskId	Integer	The task ID assigned to the recategorization run.
appId	Integer	The application ID assigned to the recategorization run.
projectId	Integer	The ID of the project.
status	Integer	The status ID value for the recategorization run.
totalItems	Integer	The total number of items included in the recategorization run.

Schema Field	Type	Description
loadedItems	Integer	Not used.
updatedItems	Integer	The number of files that were updated.
failedItems	Integer	The number of files where the update failed.
ignoredItems	Integer	The number of files that were ignored.
transcriptVaultId	String	The transcript vault ID assigned to the recategorization run.
taskStatus	String	The status of the recategorization run (e.g., **Success**).
rtStartDate **rtEndDate**	String	The start and end dates of the recategorization run.
rtTaskAction	String	A description of the recategorization task (e.g., **Record Category applied**).
rtUserName	String	The name of the ZL UA use who performed the recategorization run.

POST: Get Recategorization Task Status (getRecategorizationTaskStatus)

Obtain the status of a specific recategorization task that was performed upon files (i.e., the application or removal of Record Categories).

Path

http://localhost:8080/ps/api/v1/efm/tasks/getrecategorizationtaskstatus/{iTaskId}

Request Parameters

Parameter	Type	Description
iTaskId	Integer	The ID of the task to be viewed. Task IDs are returned in the response schema when you use the POST: Manually Apply Record Category endpoint to apply or remove Record Categories. For more information on these endpoints, refer to *EFM/Tags* on page 192. You can also view the task IDs for all recategorization tasks that have been performed within a given project with the GET: Get Recategorization Runs endpoint described in the previous section.

Request Body Schema Fields

None.

Response Schema Fields

The endpoint returns the following fields for the specified export task.

Schema Field	Type	Description
taskId	Integer	The task ID assigned to the operation.
taskStatus	String	The result of the operation (e.g., **Success**).
countTotal	Integer	The total number of files specified in the input to the task.
countUpdated	Integer	The number of files that were successfully updated.
countNotProcessed	Integer	The number of files that were not processed.
countFailed	Integer	The number of files where the update failed (e.g., the Record Category could not be applied or removed to the file).

EFM/Project Privileges

In the EFM module, you can assign project privileges to users and security groups with the EFM module to determine what operations they will be able to perform within each project. Privileges are granted on a project-by-project basis, so a user or security group can have different privilege level within different projects. Use these endpoints to grant, revoke and manage project privileges within the EFM module:

- *PUT: Add Group Project Privileges (addgroupprojectprivileges)*
- *PUT: Add User Project Privileges (adduserprojectprivileges)*
- *GET: Get Column Configuration for all Projects (getAllProjectColumnConfig)****
- *GET: Get Project Column Configuration for a Project (getProjectColumnConfig)****
- *GET: Get Project Entity IDs (getProjectEntityIds)*
- *GET: Get Project Entity Types (getProjectEntityTypes)*
- *GET: Get Project Privileges Using Project ID (getprojectprivileges)*
- *GET: Get User Privileges of Projects (getuserprivilegesofprojects)*
- *PUT: Grant Group Project Privileges (grantgroupprojectprivileges)*
- *PUT: Grant User Project Privileges (grantuserprojectprivileges)*
- *PUT: Revoke All Project Privileges (revokeallprojectprivileges)*
- *PUT: Revoke Group Project Privileges (revokegroupprojectprivileges)*
- *PUT: Revoke User Project Privileges (revokeuserprojectprivileges)*
- *PUT: Set Project Column Configuration for a Project (setProjectColumnConfig)****

***These endpoints are not related to project privileges. Use them to view and manage the default display options for the EFM module's Project View.

PUT: Add Group Project Privileges (addgroupprojectprivileges)

Assign project privileges to a **single** security group that has been created within a project. You can use the PUT: Grant Group Project Privileges endpoint described on page 178 to grant privileges to **multiple** security groups simultaneously.

Path

http://localhost:8080/ps/api/v1/efm/privileges/addgroupprojectprivileges

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
zlpUserIds	Integer	Specify the ZLP user ID of the security group to grant the privileges to. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
The remaining fields are Boolean values. Set to True to grant the privilege to the security groups specified in the request. For more extensive details on the operations that each privilege provides access to, refer to the *Granting Project Privileges* section of the *ZL UA Enterprise Files Management Administrator's Guide.*		
fProjectAdminPrivilege	Boolean	Has access to all EFM functionality for the project: project configurations, administrative tasks, tag management, searches, reports, audit trails and granting project roles to other users.
fReadPrivilege	Boolean	Can view the contents of the project and run reports.
fAnalyticsReview	Boolean	Can view the contents of the project, tag items, and run searches and reports
fAnalyticsSearch	Boolean	Can view the contents of the project and run searches and reports.

Schema Field	Type	Description
fAnalyticsAudits	Boolean	Can view audit trails, view the contents of the project, and run reports.
fRetentionApproval	Boolean	Can approve the destruction of files that are eligible for disposition because their records management lifecycle has expired.

Response Schema Fields

Schema Field	Type	Description
entityId	Integer	The ID assigned to the project entity created by the request. Each project entity represents a security group that has been assigned privileges within the project, and you can use the ID to retrieve information regarding this with other endpoints.
entityType	Integer	The entity type.
projectPrivileges	String	An array of strings identifying the privileges assigned to the security group.

PUT: Add User Project Privileges (adduserprojectprivileges)

Assign project privileges to a **single** user. You can use the PUT: Grant User Project Privileges endpoint described on page 180 to grant privileges to **multiple** users simultaneously.

Path

http://localhost:8080/ps/api/v1/efm/privileges/adduserprojectprivileges

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
zlpUserIds	Integer	Specify the ZLP user IDs of the user to grant the privileges to. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
The remaining fields are Boolean values. Set to True to grant the privilege to the users specified in the request. For more extensive details on the operations that each privilege provides access to, refer to the *Granting Project Privileges* section of the *ZL UA Enterprise Files Management Administrator's Guide.*		
fProjectAdminPrivilege	Boolean	Has access to all EFM functionality for the project: project configurations, administrative tasks, tag management, searches, reports, audit trails and granting project roles to other users.
fReadPrivilege	Boolean	Can view the contents of the project and run reports.
fAnalyticsReview	Boolean	Can view the contents of the project, tag items, and run searches and reports
fAnalyticsSearch	Boolean	Can view the contents of the project and run searches and reports.
fAnalyticsAudits	Boolean	Can view audit trails, view the contents of the project, and run reports.

Schema Field	Type	Description
fRetentionApproval	Boolean	Can approve the destruction of files that are eligible for disposition because their records management lifecycle has expired.

Response Schema Fields

Schema Field	Type	Description
entityId	Integer	The ID assigned to the project entity created by the request. Each project entity represents a user that has been assigned privileges within the project, and you can use the ID to retrieve information regarding this with other endpoints.
entityType	Integer	The entity type.
projectPrivileges	String	An array of strings identifying the privileges assigned to the user.

GET: Get Column Configuration for all Projects (getAllProjectColumnConfig)

Retrieve a list of the columns that are available to be displayed for each file in the EFM module's **File Space** tab. You can configure the columns to be displayed for a project with the PUT: Set Project Column Configuration endpoint described on page 187, and you can view the columns that have been configured for display for a project with the GET: Get Project Column Configuration endpoint described on page 172.

Path

http://localhost:8080/ps/api/v1/efm/privileges/getallprojectcolumnconfig

Request Parameters

None.

Request Body Schema Fields

None.

Response Schema Fields

The response schema includes a series of strings listing the columns that are available for display for each project in the EFM module's **File Space** tab.

GET: Get Project Column Configuration for a Project (getProjectColumnConfig)

Use this endpoint to view the columns that have been configured to displayed in the EFM module's **File Space** tab for a project.

Path

http://localhost:8080/ps/api/v1/efm/privileges/getprojectcolumnconfig/{projectId}/

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

The response schema includes a series of strings listing the columns that have been configured for display in the EFM module's **File Space** tab for the project.

GET: Get Project Entity IDs (getProjectEntityIds)

A project privilege entity represents the privileges that have been assigned to a user or security group within a project. Use this endpoint to retrieve the entity IDs for the project privileges that have been assigned within a project.

Path

http://localhost:8080/ps/api/v1/efm/privileges/getprojectentityids/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project from which you want to retrieve the information. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
Page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

The endpoint returns the following information for each use or security group that has been granted privileges within the specified project.

Schema Field	Type	Description
stUserName	String	The name of the user or security group.
entityId	Integer	The ID assigned to the project privilege entity created for the user or security group.

GET: Get Project Entity Types (getProjectEntityTypes)

A project privilege entity represents the privileges that have been assigned to a user or security group within a project. Use this endpoint to retrieve the entity types that have been assigned within a project.

Path

http://localhost:8080/ps/api/v1/efm/privileges/getprojectentitytypes/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project from which you want to retrieve the information. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
Page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

The endpoint returns the following information for each project entity that has been created within the project.

Schema Field	Type	Description
stUserName	String	The name of the user or security group.
entityType	Integer	The type of project privilege entity that was created for the user or security group: • 0: Users • 3: Security Groups

GET: Get Project Privileges Using Project ID (getprojectprivileges)

Get the project privileges that have been assigned within a project.

Path

http://localhost:8080/ps/api/v1/efm/privileges/getprojectprivileges/{projectId}

Request Parameters

Parameter	Type	Description
Page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	
projectId	Integer	Specify the ID of the project from which you want to retrieve the information. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
User/Security Group Project Privileges: Includes an entry that includes the following fields for each user or security group that has been assigned privileges within the project.		
username	String	The name of the user or security group.
privileges	String	The privileges assigned to the user or security group.

GET: Get User Privileges of Projects (getuserprivilegesofprojects)

View the project privileges that have been assigned to a user. This endpoint cannot be used to view the project privileges that have been granted to security groups.

Path

http://localhost:8080/ps/api/v1/efm/privileges/getuserprivilegesofprojects/{userId}

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	
userId	Integer	Specify the ZLP user ID of the user whose privileges you want to view. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
User Project Privileges: Includes an entry that includes the following fields for each project that the user has been assigned privileges within.		
projectId	Integer	The ID of the project.
projectName	String	The name of the project.
privileges	String	The privileges assigned to the user within the project.

PUT: Grant Group Project Privileges (grantgroupprojectprivileges)

Grant project privileges to security groups that have been created within a project.

Path

http://localhost:8080/ps/api/v1/efm/privileges/grantgroupprojectprivileges

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

zlpUserIds	Integer	Specify a list of ZLP user IDs to identify the security groups to grant the privileges to. Specify the group IDs as a comma-separated list. For example:
		`"zlpUserIds": [`
		`0,1,2,3`
		`]`
		The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.

The remaining fields are Boolean values. Set to True to grant the privilege to the security groups specified in the request. For more extensive details on the operations that each privilege provides access to, refer to the *Granting Project Privileges* section of the *ZL UA Enterprise Files Management Administrator's Guide.*		
fProjectAdminPrivilege	Boolean	Has access to all EFM functionality for the project: project configurations, administrative tasks, tag management, searches, reports, audit trails and granting project roles to other users.
fReadPrivilege	Boolean	Can view the contents of the project and run reports.
fAnalyticsReview	Boolean	Can view the contents of the project, tag items, and run searches and reports

Schema Field	Type	Description
fAnalyticsSearch	Boolean	Can view the contents of the project and run searches and reports.
fAnalyticsAudits	Boolean	Can view audit trails, view the contents of the project, and run reports.
fRetentionApproval	Boolean	Can approve the destruction of files that are eligible for disposition because their records management lifecycle has expired.

Response Schema Fields

Schema Field	Type	Description
entityId	Integer	The ID assigned to the project entity created by the request. Each project entity represents a security group that has been assigned privileges within the project, and you can use the ID to retrieve information regarding this with other endpoints.
entityType	Integer	The entity type. This is set to 3 for security groups.
projectPrivileges	String	An array of strings identifying the privileges assigned to the security group.

PUT: Grant User Project Privileges (grantuserprojectprivileges)

Grant project privileges to a user (or group of users).

Path

http://localhost:8080/ps/api/v1/efm/privileges/grantuserprojectprivileges

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

zlpUserIds	Integer	Specify a list of ZLP user IDs to identify the users to grant the privileges to. Specify the group IDs as a comma-separated list. For example:
		`"zlpUserIds": [`
		`0,1,2,3`
		`]`
		The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.

The remaining fields are Boolean values. Set to True to grant the privilege to the users specified in the request. For more extensive details on the operations that each privilege provides access to, refer to the *Granting Project Privileges* section of the *ZL UA Enterprise Files Management Administrator's Guide.*		
fProjectAdminPrivilege	Boolean	Has access to all EFM functionality for the project: project configurations, administrative tasks, tag management, searches, reports, audit trails and granting project roles to other users.
fReadPrivilege	Boolean	Can view the contents of the project and run reports.
fAnalyticsReview	Boolean	Can view the contents of the project, tag items, and run searches and reports

Schema Field	Type	Description
fAnalyticsSearch	Boolean	Can view the contents of the project and run searches and reports.
fAnalyticsAudits	Boolean	Can view audit trails, view the contents of the project, and run reports.
fRetentionApproval	Boolean	Can approve the destruction of files that are eligible for disposition because their records management lifecycle has expired.

Response Schema Fields

Schema Field	Type	Description
entityId	Integer	The ID assigned to the project entity created by the request. Each project entity represents a user that has been assigned privileges within the project, and you can use the ID to retrieve information regarding this with other endpoints.
entityType	Integer	The entity type. This is set to 0 for users.
projectPrivileges	String	An array of strings identifying the privileges assigned to the user.

PUT: Revoke All Project Privileges (revokeallprojectprivileges)

Revoke all privileges that have been assigned within a project.

Path

http://localhost:8080/ps/api/v1/efm/privileges/revokeallprojectprivileges/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project from which you want to revoke privileges. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

A string message indicating whether the privileges were successfully revoked or not.

PUT: Revoke Group Project Privileges (revokegroupprojectprivileges)

Revoke privileges that have been assigned to a specific list of security groups within a project.

Path

http://localhost:8080/v1/efm/privileges/revokegroupprojectprivileges

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

zlpUserIds	Integer	An array of group IDs to identify the security groups from which the project privileges should be revoked. Specify the group IDs as a comma-separated list. For example:
		`"zlpUserIds": [`
		`0,1,2,3`
		`]`
		The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.

Response Schema Fields

Schema Field	Type	Description
Additional Prop: Includes the following fields for each security group specified in the request.		
success	Boolean	Indicates whether privileges were revoked from the security group successfully (True) or not (False).
result	String	The result of the request. If **Success** is set to True, a message will display indicating that privileges have been revoked.
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

PUT: Revoke User Project Privileges (revokeuserprojectprivileges)

Revoke privileges that have been assigned to users within a project.

Path

http://localhost:8080/ps/api/v1/efm/privileges/revokeuserprojectprivileges

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

zlpUserIds	Integer	An array of user IDs to specify the users from whom the project privileges should be revoked. Specify the user IDs as a comma-separated list. For example:
		`"zlpUserIds": [`
		`0,1,2,3`
		`]`
		The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.

Response Schema Fields

Schema Field	Type	Description
Additional Prop: Includes the following fields for each user specified in the request.		
success	Boolean	Indicates whether privileges were revoked from the group successfully (True) or not (False).
result	String	The result of the request. If successful, a message will be displayed that privileges have been revoked.
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

PUT: Set Project Column Configuration for a Project (setProjectColumnConfig)

Use this endpoint to view the columns that have been configured to be displayed for each file in the EFM module's **File Space** tab in a project.

Path

http://localhost:8080/ps/api/v1/efm/privileges/setprojectcolumnconfig/

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
columns	Array	Select the columns that should be displayed for the project: **Record Category, Tags**, **Size**, **Last Modified**, **Trigger Date**, **Disposition Date**

Request Body Schema Fields

None.

Response Schema Fields

The response schema returns a string indicating that the project configuration has been successfully updated.

EFM/Security Groups

A security group represents a specific group of users in ZL UA. When you assign project privileges to a security group within the EFM module, those privileges are granted to all users in the security group. The following sections describe the EFM/Security Groups endpoints available in the REST API. You can use these endpoints to create and manage security groups:

- *PUT: Ensure Security Group (ensuresecuritygroup)*
- *GET: Get All Security Groups (getallsecuritygroups)*
- *PUT: Remove Security Group (removesecuritygroup)*

PUT: Ensure Security Group (ensuresecuritygroup)

Create a security group.

Path

http://localhost:8080/ps/api/v1/efm/security/ensuresecuritygroup

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description

aIdZlpUser	Integer	An array of ZLP user IDs to specify the users who should be added to the new security group. Specify the user IDs as a comma-separated list. For example:
		`"aIdzlpUser": [`
		`0,1,2,3`
		`]`
		The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each user specified in the request.		
success	Boolean	Indicates whether the users were added successfully (True) or not (False).
result	String	The result of the request. If **Success** is True, then this contains information regarding the security group.
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

GET: Get All Security Groups (getallsecuritygroups)

Retrieve a list of all security groups.

Path

http://localhost:8080/ps/api/v1/efm/security/getallsecuritygroups

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Security Groups: Includes an entry for each user added to the security group. The response schema fields included in each entry are the same as those included in the response returned after creating the user. For descriptions of these fields, refer to *POST: Create User* on page 409.		

PUT: Remove Security Group (removesecuritygroup)

Remove a security group.

Path

http://localhost:8080/ps/api/v1/efm/security/removesecuritygroup

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description

aIdZlpUser	Integer	An array of ZLP user IDs to specify the security groups to be removed. Specify the security group IDs as a comma-separated list. For example:
		`"aIdzlpUser": [`
		`0,1,2,3`
		`]`
		The REST API includes several endpoints you can use to retrieve user and security group configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each security group specified in the request.		
success	Boolean	Indicates whether the security group was removed successfully (True) or not (False).
result	String	The result of the request. If **Success** is set to True, this will indicate that the security group was removed successfully.
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

EFM/Tags

In ZL UA, a Record Category defines a file's disposition period (i.e., how long it will be retained within ZL UA before being permanently deleted from ZL UA and the source server). When creating Record Categories, you will also need to create retention codes to define the disposition periods in your system.

The following sections describe the EFM/Tags endpoints available in the REST API. Use these endpoints to create and manage Record Categories and retention codes:

- *POST: Create Record Category (createRecordCategory)*
- *POST: Create Retention Code (createRetentionCode)*
- *DELETE: Delete Retention Code (deleteRetentionCode)*
- *POST: Get Record Categories (getRecordCategories)*
- *POST: Get Retention Code (getRetentionCode)*
- *POST: Manually Apply Record Category (manuallyApplyRecordCategory)*
- *PUT: Remove Record Category (removeRecordCategory)*
- *PUT: Update Record Category (updateRecordCategory)*
- *PUT: Update Retention Code (updateRetentionCode)*

POST: Create Record Category (createRecordCategory)

Use this endpoint to create a Record Category.

Path

http://localhost:8080/ps/api/v1/efm/tags/createrecordcategory

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
recordCategoryId	Integer	Ignore this field when creating a Record Category.
recordCategoryName	String	Enter the name of the Record Category.
recordCategoryDescription	String	Enter a description of the Record Category.
recordCategoryPrefix	String	Enter a unique prefix identifier for the Record Category.
recordCategoryTriggerFormula	String	Use any of the following values to specify the start date for the file's retention period: • @fileDate@ • @publishDate@ • @recordDate@ • A value identifying a specific date that is not more than 100 years old or 100 years into the future. Enter the date in MM/DD/YYYY format. For example, May 26th 2025 then should be entered as: 05/26/2025 The file date refers to the date that the file was processed into ZL UA. Please note that for In-Place files, the publish date and record date are set to the date the file was last modified due to the way that ZL UA manages In-Place files internally.

Schema Field	Type	Description
recordCategoryRetentionCodeId	Integer	The ID of the retention code created for the Record Category. You can use the POST: Get Retention Codes endpoint described on page 200 to retrieve retention code ID values. For more information on creating retention codes, refer to *POST: Create Retention Code* on page 196.
archive	Boolean	Set this to True so that files this Record Category is applied to when they are ingested into ZL UA will be marked for Selective Archiving. This is not applicable to files that the Record Category is manually applied to.
filingEnabled	Boolean	Set this to True so that the Record Category can be applied to files. Set this to False to restrict the Record Category, so it cannot be applied to files.

Response Schema Fields

Schema Field	Type	Description
recordCategoryId	Integer	The ID assigned to the Record Category.
recordCategoryName	String	The name of the Record Category.
recordCategoryDescription	String	The description of the Record Category.
recordCategoryPrefix	String	A unique prefix identifier for the Record Category.
recordCategoryTriggerFormula	String	The start date for the file's retention period: • @fileDate@ • @publishDate@ • @recordDate@ • A value identifying a specific date. The file date refers to the date that the file was processed into ZL UA. Please note that for In-Place files, the publish date and record date are set to the date the file was last modified due to the way that ZL UA manages In-Place files internally.

Schema Field	Type	Description
recordCategoryRetentionCodeId	Integer	The ID of the retention code created for the Record Category. For more information on creating retention codes, refer to *POST: Create Retention Code* on page 196.
archive	Boolean	If True, then files this Record Category is applied to when they are ingested into ZL UA will be marked for Selective Archiving. This is not applicable to files that the Record Category is manually applied to.
recordCategoryCreateDate **recordCategoryUpdateDate**	String	The dates that the Record Category was created or last updated.
recordCategoryRetentionCodeName	String	The retention code assigned to the Record Category.
The next group of fields returned by this endpoint include internal information regarding the retention code assigned to the Record Category that are used as inputs when creating or updating the Record Category. For more information on creating retention codes, refer to POST: Create Retention Code on page 196.		
filingEnabled	Boolean	If True, the Record Category can be applied to files. If False, the Record Category is restricted and cannot be applied to files.

POST: Create Retention Code (createRetentionCode)

Use this endpoint to create a retention code to apply to a Record Category.

Path

http://localhost:8080/ps/api/v1/custodian/v1/efm/tags/createretentioncode

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
retentionCodeId	Integer	Not required when creating a retention code.
recordStoreId	Integer	The record store ID to assign to the retention code. You can retrieve record store IDs from the **RetentionCode** database table. In addition, you can check the **RecordStore** database table to see the record store IDs assigned to different tenants.
retentionCodeName	String	The name of the retention code.
beginSuperseded	Boolean	Set to True to designate that each file the tag is applied to begin its disposition lifecycle only after the record and/or its category/ folder has been superseded (even when the trigger date has already passed). Records that are not superseded or are in an un-superseded category/folder will remain active records until the record and/or its entire category/folder is superseded, at which point the record begins its lifecycle in accordance with the defined disposition properties of the code.
beginDispositionAuthority	Boolean	Set to True to add Disposition Authority to the code. Disposition Authority tags define the legal authority that allows for disposition to be carried out upon files. They are used for tracking and reporting purposes.
retentionPeriod	Integer	Specify the retention period for the retention code. Use the **retentionUnit** field described below to specify the units for the retention period.

Schema Field	Type	Description
retentionUnit	String	Specify the retention unit for the retention code: **YEAR**, **WEEK**, **MONTH** or **DAY** For example, if **retentionPeriod** is set to 7 and this field is set to **MONTH**, the retention period would be 7 months.

Response Schema Fields

The fields included in the response schema are the same set of fields required in the request schema, as well as the ID that has been assigned to the retention code. These fields define the retention code's configuration.

DELETE: Delete Retention Code (deleteRetentionCode)

Use this endpoint to delete a retention code.

Path

http://localhost:8080/ps/api/v1/efm/tags/deleteretentioncode/{retentionCodeId}

Request Parameters

Parameter	Type	Description
retentionCodeId	Integer	The ID of the retention code to be deleted. You can retrieve retention code IDs with the POST: Get Retention Code endpoint described on page 200.

Request Body Schema Fields

None.

Response Schema Fields

The endpoint returns a string indicating whether the retention code was successfully deleted.

POST: Get Record Categories (getRecordCategories)

Use this endpoint to retrieve the configuration of a Record Category.

Path

http://localhost:8080/ps/api/v1/efm/tags/getrecordcategories

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

Schema Field	Type	Description
items	Integer	Enter a comma-separated list of ID values identifying the Record Categories whose configurations are to be retrieved. If you specify an ID that has not been assigned to a Record Category, it will skip that ID. If all the specified IDs are invalid, an error will be generated. If you do not provide any input, then the configurations of all the Record Categories will be retrieved and displayed in a list format. You can retrieve Record Category configuration and ID values with the POST: Get Record Category endpoint described on page 199.

Response Schema Fields

This endpoint returns an entry for each Record Category specified in the input. Each entry includes a series of fields that represent the configuration of the Record Category. For descriptions of these fields, refer to *POST: Create Record Category* on page 193.

POST: Get Retention Code (getRetentionCode)

Use this endpoint to retrieve the configuration of a retention code.

Path

http://localhost:8080/ps/api//v1/efm/tags/getretentioncode

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

Schema Field	Type	Description
items	Integer	Enter a comma-separated list of IDs to specify the retention codes whose configurations are to be retrieved. If you specify an ID that has not been assigned to a retention code, it will skip that ID. If all the specified IDs are invalid, an error will be generated. If you do not provide any input, then the configurations of all the retention codes will be retrieved and displayed in a list format. You can use the POST: Get Retention Code endpoint to retrieve retention code ID values. For more information, refer to *POST: Get Retention Code* on page 200.

Response Schema Fields

This endpoint returns an entry for each retention code specified in the input. Each entry includes a series of fields that represent the configuration of the retention code. For descriptions of these fields, refer to *POST: Create Retention Code* on page 196.

POST: Manually Apply Record Category (manuallyApplyRecordCategory)

Use this endpoint to manually apply a Record Category to a file or group of files.

Path

http://localhost:8080/ps/api/v1/efm/tags/manuallyapplyrecordcategory

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
listUnids	String	Enter a comma-separated list of file IDs to specify the files to be modified. You can retrieve file IDs from the **InPlaceFileItem** database table.
recordCategoryId	Integer	Enter the ID of the Record Category to be applied to the files. You can retrieve Record Category IDs with the POST: Get Record Category endpoint described on page 199.
idProj	Integer	Specify the ID of the project containing the files to be modified. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Response Schema Fields

A string indicating whether the operation was successful, as well as the task ID assigned to the operation. You can use the task ID to check the task status with the POST: Get Recategorization Task Status endpoint described on page 166.

PUT: Remove Record Category (removeRecordCategory)

Use this endpoint to remove the Record Category that has been assigned to a file or group of files. After the Record Category has been removed, the file(s) will no longer be subject to removal via the Disposition Workflow.

Path

http://localhost:8080/ps/api/v1/efm/tags/removerecordcategory

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
listUnids	String	Enter a comma-separated list of file IDs to specify the files to be modified. You can retrieve file IDs from the **InPlaceFileItem** database table.
idProj	Integer	Specify the ID of the project containing the files to be modified. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Response Schema Fields

A string indicating whether the operation was successful, as well as the task ID assigned to the operation. For example:

```
Task Started for remove Record Category, Task Id: 40
```

You can use the task ID to check the task status with the POST: Get Recategorization Task Status endpoint described on page 166.

PUT: Update Record Category (updateRecordCategory)

Use this endpoint to update a Record Category.

Path

http://localhost:8080/ps/api/v1/efm/tags/updaterecordcategory

Request Body Schema Fields

Schema Field	Type	Description
recordCategoryId	Integer	Specify the ID of the Record Category being updated. You can retrieve Record Category IDs with the POST: Get Record Category endpoint described on page 199.
recordCategoryName	String	Enter a name of the Record Category.
recordCategoryDescription	String	Enter a description of the Record Category.
recordCategoryPrefix	String	Enter a unique prefix identifier for the Record Category.
recordCategoryTriggerFormula	String	Use any of the following values to specify the start date for the file's retention period: • @fileDate@ • @publishDate@ • @recordDate@ • A value identifying a specific date that is not more than 100 years old or 100 years into the future. Enter the date in MM/DD/YYYY format. For example, May 26th 2025 then should be written as 05/26/2025. The file date refers to the date that the file was processed into ZL UA. Please note that for In-Place files, the publish date and record date are set to the date the file was last modified due to the way that ZL UA manages In-Place files internally.

Schema Field	Type	Description
recordCategoryRetentionCodeId	Integer	The ID of the retention code created for the Record Category. This determines the retention period for each file the Record Category is applied to. You can use the POST: Get Retention Codes endpoint described on page 200 to retrieve retention code ID values. For more information on creating retention codes, refer to *POST: Create Retention Code* on page 196.
archive	Boolean	Set this to True so that files this Record Category is applied to when they are ingested into ZL UA will be marked for selective archiving (meaning that they will be copied into the ZL Vault). This is not applicable to files that the Record Category is manually applied to.
filingEnabled	Boolean	Set this to True so that the Record Category can be applied to files. Set this to False to restrict the Record Category, so it cannot be applied to files.

Response Schema Fields

Schema Field	Type	Description
recordCategoryId	Integer	The ID assigned to the Record Category.
recordCategoryName	String	The name of the Record Category.
recordCategoryDescription	String	The description of the Record Category.
recordCategoryPrefix	String	A unique prefix identifier for the Record Category.

Schema Field	Type	Description
recordCategoryTriggerFormula	String	The start date for the file's retention period: • @fileDate@ • @publishDate@ • @recordDate@ • A value identifying a specific date. The file date refers to the date that the file was processed into ZL UA. Please note that for In-Place files, the publish date and record date are set to the date the file was last modified due to the way that ZL UA manages In-Place files internally.
recordCategoryRetentionCodeId	Integer	The ID of the retention code created for the Record Category.
archive	Boolean	If True, then files this Record Category is applied to when they are ingested into ZL UA will be marked for Selective Archiving. This is not applicable to files that the Record Category is manually applied to.
filingEnabled	Boolean	If True, the Record Category can be applied to files. If False, the Record Category is restricted and cannot be applied to files.

PUT: Update Retention Code (updateRetentionCode)

Use this endpoint to update a retention code's configuration.

Path

http://localhost:8080/ps/api//v1/efm/tags/updateretentioncode

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
retentionCodeId	Integer	The ID assigned to the retention code being updated. You can retrieve retention code IDs with the POST: Get Retention Code endpoint described on page 200.
recordStoreId	Integer	The record store ID to assign to the retention code. You can retrieve record store IDs from the **RetentionCode** database table. In addition, you can check the **RecordStore** database table to see the recordStore IDs assigned to different tenants.
retentionCodeName	String	The name of the retention code.
beginSuperseded	Boolean	Set to True designate that each file the tag is applied to begin its disposition lifecycle only after the record and/or its category/ folder has been superseded (even when the trigger date has already passed). Records that are not superseded or are in an un-superseded category/folder will remain active records until the record and/or its entire category/folder is superseded, at which point the record begins its lifecycle in accordance with the defined disposition properties of the code.
beginDispositionAuthority	Boolean	Set to True to add Disposition Authority to the code. Disposition Authority tags define the legal authority that allows for disposition to be carried out upon files. They are used for tracking and reporting purposes.
retentionPeriod	Integer	Specify the retention period for the retention code. Use the **retentionUnit** field described below to specify the units for the retention period.

Schema Field	Type	Description
retentionUnit	String	Specify the retention unit for the retention code: **YEAR**, **WEEK**, **MONTH** or **DAY** For example, if **retentionPeriod** is set to 7 and this field is set to **MONTH**, the retention period would be 7 months.

Response Schema Fields

The fields included in the response schema are the same set of fields required in the request schema, as well as the ID that has been assigned to the retention code. These fields define the retention code's configuration.

FAM/Roles

In the FAM module, a role is an application-level set of permissions that determines what users can access and what actions users can perform in the application. The following sections describe the FAM/Roles endpoints available in the REST API. Use these endpoints to grant and revoke roles for use within the FAM module:

- *PUT: Grant User Roles (grantroles)*: Grant roles to a user.
- *PUT: Revoke User Roles (revokeroles):* Revoke roles from a user.

PUT: Grant User Roles (grantroles)

Grant roles to a user.

Path

http://localhost:8080/ps/api/v1/FAMRoles/grantroles

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to grant roles to. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
userRoles: The remaining input is an array of fields defining the roles to be granted. Specify the following for each **userRole** entry:		
roleId	Integer	The role ID. For information on retrieving role IDs, refer to the following sections: • *GET: Get All Custom Roles* on page 370 • *GET: Get All System Roles* on page 372
roleName	String	The display name of the role. For information on retrieving role names, refer to the following sections: • *GET: Get All Custom Roles* on page 370 • *GET: Get All System Roles* on page 372
scope	String	Specify whether the role should be granted globally so that it is applicable to all departments, or if it should be granted to specific departments only: • **Global**: Global • **InclRecur**: On selected departments recursively • **Incl**: On selected departments only

Schema Field	Type	Description
allScopeDomainIds	Integer	An array of domain IDs to specify the departments the role is applicable to for roles that are only granted on selected departments. These can be retrieved from the **ArchiveServer** database Table. Specify the domain IDs as a comma-separated list. For example: `"AllScopeDomainIds": [ 0,1,2,3 ]`

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each role specified in the request.		
success	Boolean	Indicates whether the role was granted successfully (True) or not (False).
result	String	The result of the request. The string will indicate how the role has been applied (on which departments, scope, role ID, etc).
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

PUT: Revoke User Roles (revokeroles)

Revoke roles that have been previously assigned to a user.

Path

http://localhost:8080/ps/api/v1/FAMRoles/revokeroles

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to modify. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
roles	Integer	An array of role IDs to specify the roles to be revoked. Specify the role IDs as a comma-separated list. For example: `"Roles": [ 0,1,2,3 ]` For information on retrieving role IDs, refer to the following sections: • *GET: Get All Custom Roles* on page 370 • *GET: Get All System Roles* on page 372

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each role specified in the request.		
success	Boolean	Indicates whether the role was revoked successfully (True) or not (False).
result	String	The result of the request. If successful, a message will display indicating that the role has been revoked.
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

UAA/Data Sources

To ingest any type of data into ZL UA, a reference pointing to its source server must be created in the ZL Unified Archival Admin (UAA) application. This reference - referred to as a ZL UA application server or a content server - integrates ZL UA with the physical server on the corporate network.

The following sections describe the UAA/DataSources endpoints available in the REST API. Use these endpoints to create and update servers in ZL UA:

- *POST: Create Box Server (createBoxServer)*
- *POST: Create An EWS Mail Server (createmailserver/ews)*
- *POST: Create File Server (createFileServer)*
- *POST: Create a Google Drive Server (creategoogledriveserver)*
- *POST: Create a Google Mail Server (creategooglemailserver)*
- *POST: Create a Lotus Domino Mail Server (createlotusdominomailserver)*
- *POST: Create an Exchange Mail Server (createmailserver/microsoftexchange)*
- *POST: Create a OneDrive Server (createonedriveserver)*
- *POST: Create a SharePoint Server (createsharepointserver)*
- *DELETE: Delete Server (deletemailserver)*
- *GET: Get All Servers (getallmailservers)*
- *POST: Get Cumulative Stats (getCumulativeStats)*
- *POST: Get Data License Stats (getDataLicenseStats)*
- *GET: Get Server Using ID (getmailserverusingid)*
- *GET: Get Server Using Name (getmailserverusingname)*
- *POST: Get Storage Stats (getStorageStats)*
- *POST: Get Tenant Data License Stats (getTenantDataLicenseStats)*
- *PUT: Update Box Server (updateBoxServer)*
- *PUT: Update an EWS Mail Server (updateewsmailserver)*
- *PUT: Update File Server (updatefileserver)*
- *PUT: Update a Google Drive Server (updategoogledriveserver)*
- *PUT: Update a Google Mail Server (updategooglemailserver)*
- *PUT: Update a Lotus Domino Mail Server (updatelotusdominoserver)*
- *PUT: Update an Exchange Mail Server (updatemailserver/exchange)*
- *PUT: Update a OneDrive Server's Configuration (updateonedriveserver)*
- *PUT: Update a SharePoint Server (updatesharepointserver)*

POST: Create Box Server (createBoxServer)

Create a Box server.

Path

http://localhost:8080/ps/api/v1/FileServer/createboxserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
serverName	String	Enter the name of the Box server.
credential	String	Enter the name of the Credentials object required to access the Box server. This must be created with the UAA module. The **CloudCredentials** database table includes the names of the Credentials objects that have been added to your installation. For instructions on creating Credentials objects for use with Box servers, refer to the *ZL UA Box Archiving Administrator's Guide.*

Response Schema Fields

The schema fields returned for the server are the same as those specified in the request schema, as well as the following additional fields. These fields define the server's configuration.

Schema Field	Type	Description
id	Integer	The ID assigned to the Box server.
name	String	The name assigned to the Box server.
serverType **serverSubType**	String	The server type and sub-type.
url	String	The server URL.

Schema Field	Type	Description
param	String	Additional information about the server, including the name of the Credentials object assigned to it.
discoveryName	String	The server discovery name.
group	String	The server group.
dateCreate	String	A timestamp indicating when the server was created or last updated.

POST: Create An EWS Mail Server (createmailserver/ews)

Create a Microsoft Exchange Web Services (EWS) mail server.

Path

http://localhost:8080/ps/api/v1/FileServer/createmailserver/ews

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
serverName	String	Specify the server name.
serverURL	String	Specify the URL of the Microsoft EWS server.
discoveryMethod	String	Enter the name of the Discovery object to use with the server. This must be created with the UAA module.
exchangeServiceUrl	String	Use the following URL: https://outlook.office365.com/EWS/Exchange.asmx This is the service URL for O365 presently being used by Microsoft. ZL UA will use this URL to look up mailbox information for users and teams.
userDefaultCredential	Boolean	Set to True to use the default credentials for the EWS server.
ewsServerVersion	String	Specify the EWS server version by entering any of the following values: 2010 2013 2016 2019 O365
ewsCredential	String	Enter the name of the Credentials object required to access the server. This must be created with the UAA module. The **CloudCredentials** database table includes the Credential object names within your installation. For detailed instructions on Credentials objects and other aspects of MS Teams server administration, refer to the *ZL UA MS Teams Administrator's Guide*.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server subtype
url	String	The server URL.
param	String	Additional information about the server, including the server version, Exchange server URL, impersonation mode setting and auto-discovery setting.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

POST: Create File Server (createFileServer)

Create a file server.

Path

http://localhost:8080/ps/api//v1/FileServer/createfileserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
serverName	String	Specify the server name.
ip	String	Specify the server IP address.
port	String	Type the port number used by the ZL File Connector. The default value is 9975. Leave the default value unless you are changing this port in the ZL File Connector as well.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server subtype.
url	String	The server URL.
param	String	The server port.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

POST: Create a Google Drive Server (creategoogledriveserver)

Create a Google Drive server.

Path

http://localhost:8080/ps/api/v1/FileServer/creategoogledriveserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	This field is not required when creating a Google Drive server.
serverName	String	Enter the display name for the Google Drive server.
serverURL	String	Enter the URL of the Google Drive server.
adminUserToImpersonate	String	Enter the email address used to create the Google Drive service account for this server. This can be left blank if the service account has been directly added to shared drives. In this case, the service account created must be present as a user in every shared drive to be crawled.

consumerCertFile	String	Specify the path and file name of the JSON key file downloaded when creating the Google Drive service account for this server. The file path entered must be relative to the server you are creating.
		
		For example, consider a ZL UA installation where there are 3 servers: Server1, Server2 and Server3. If the key file is on the D: drive of Server2 and you are creating the entry for Server2, you can enter the following file path:
		
		`D:\\file.txt`
		
		However, if you are creating the entry for one of the other servers, you need to enter the server name as part of the file path:
		
		`\\server2\D$\file.txt`

Schema Field	Type	Description
scopes	String	Specify the service account scopes. Use this to set the change level of Google Drive API permissions that can be assigned to ZL.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server subtype.
url	String	The server URL.
param	String	Information about the server connector.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

POST: Create a Google Mail Server (creategooglemailserver)

Create a Google Mail server.

Path

http://localhost:8080/ps/api//v1/FileServer/creategooglemailserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
serverName	String	Enter the display name for the Google Mail server.
serverURL	String	Enter the URL of the Google Mail server.
serverHost	String	Enter the hostname or IP address of the Google Mail server.
domainName	String	Enter the server domain name.
adminUserId	String	Specify the administrator user ID for the Google Mail server.
adminPassword	String	Specify the administrator password for the Google Mail server.
consumerKey	String	Specify the consumer key generated for the Google Mail server.
consumerSecret	String	Specify the secret key generated for the Google Mail server.
serviceAccountId	String	Enter the ID for the Google Mail service account created to administer this server.

certificateFile	String	Specify the path and file name of the certificate file for the Google Mail server. The file path entered must be relative to the server you are creating.
		For example, consider a ZL UA installation where there are 3 servers: Server1, Server2 and Server3. If the key file is on the D: drive of Server2 and you are creating the entry for Server2, you can enter the following file path:
		`D:\file.txt`
		However, if you are creating the entry for one of the other servers, you need to enter the server name as part of the file path:
		`\\server2\D$\file.txt`

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server subtype
url	String	The server URL.
param	String	Additional information about the server connector.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

POST: Create a Lotus Domino Mail Server (createlotusdominomailserver)

Create a Lotus Domino mail server.

Path

http://localhost:8080/ps/api/v1/FileServer/createlotusdominomailserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
serverName	String	Enter the display name for the Lotus Domino server.
serverURL	String	Type the loop back address (127.0.0.1) or IP address of the machine that is running the ZL Domino TCP Connector.
parentServer	String	Enter the parent server URL. If specified, the connector URL, Notes password, and remote temp directory will be taken from the parent server. Alternately, enter **None** for the Parent Server field and specify these parameters below.
notesAdminPassword	String	Type the password of the administrative service account you created in Notes.
remoteRouterHost	String	Type the name of the remote router if there is no local router.
discoveryMethod	String	Enter the name of the Discovery object to use with the server. This must be created with the UAA module.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server subtype
url	String	The server URL.

Schema Field	Type	Description
param	String	Additional information about the server, such as the parent server name, admin password, remote temporary directory and remote router host.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

POST: Create an Exchange Mail Server (createmailserver/microsoftexchange)

Create a Microsoft Exchange mail server.

Path

http://localhost:8080/ps/api/v1/FileServer/createmailserver/microsoftexchange

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
serverName	String	Specify the server name.
transportHost	String	Enter the hostname or URL of the machine running the ZL MAPI Connector for this server.
discoveryMethod	String	Enter the name of the Discovery object to use with the server. This must be created with the UAA module.
serverCn	String	Enter the server's common name.
casServer	String	Type the Client Access Server name. For O365, this should be set to match the SMTP address of the transport account for the server.
exchangeServerVersion	String	Specify the Exchange server version by entering any of the following values: 2010 2013 2016 2019 O365

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.

Schema Field	Type	Description
server Type	String	The server type.
serverSubType	String	The server subtype
url	String	The server URL.
param	String	Additional information about the server, including the server CN value, CAS server name and server version.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreated	String	The date the server was created.

POST: Create a OneDrive Server (createonedriveserver)

Create a OneDrive server.

Path

http://localhost:8080/ps/api//v1/FileServer/createonedriveserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
serverName	String	Specify the server name.
credential	String	Enter the name of the Credentials object created for the server in UAA. The **CloudCredentials** database table includes the Credential object names within your installation. For more information on creating Credentials objects for OneDrive servers, refer to the *ZL UA OneDrive Archiving Administrator's Guide*.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server subtype
url	String	The server URL.
param	String	Additional information about the server.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

POST: Create a SharePoint Server (createsharepointserver)

Create a SharePoint server.

Path

http://localhost:8080/ps/api/v1/FileServer/createsharepointserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	This field is not required..
serverName	String	Specify the server name.
serverUrl	String	Enter the IP address or host name of the host server on which you have installed the SharePoint connector.
sharePointFarmDbInstance	String	This information must be retrieved from the SharePoint server's SQL DB. Go to the SharePoint DB server, and select the database for SharePoint. Copy the name of this database and paste it into this text box.
credentials	String	Enter the name of the Credentials object created for the server in UAA. The **CloudCredentials** database table includes the Credential object names within your installation. For more information on creating Credentials objects for use with SharePoint servers, refer to the *ZL UA SharePoint Archiving Administrator's Guide*.
appOnly	Boolean	Set to True to use app-only authentication (a.k.a. application permissions) for this server. Set to True if you are using the SharePoint Online connector.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.

Schema Field	Type	Description
server Type	String	The server type.
serverSubType	String	The server subtype
url	String	The server URL.
param	String	Additional information about the server SharePoint farm database instance.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.
appOnly	Boolean	Set to True when app-only authentication (a.k.a. application permissions) is enabled for the server.

DELETE: Delete Server (deletemailserver)

Delete a server.

Path

http://localhost:8080/ps/api//v1/FileServer/deletemailserver/{serverId}

Request Parameters

Parameter	Type	Description
serverId	Integer	Specify the ID of the server to be deleted. You can retrieve server IDs with the GET: Get All Servers endpoint described on page 230.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the server was deleted successfully.

GET: Get All Servers (getallmailservers)

Retrieve a list of all file and mail servers that have been added to ZL UA.

Path

http://localhost:8080/ps/api//v1/FileServer/getallmailservers

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Server Entries: The response includes an entry for each server found. The schema fields included in each entry are described below.		
id	Integer	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server sub-type.
url	String	The server URL, if applicable.
param	String	Additional information about the server.
discoveryName	String	The name of the Discovery object assigned to the server.
group	String	The server group.
dateCreate	String	The date the server was created.

POST: Get Cumulative Stats (getCumulativeStats)

Retrieve cumulative statistics regarding the file and mail storage within your ZL UA installation.

Path

http://localhost:8080/ps/api/v1/FileServer/getcumulativestats

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
startDate **endDate**	String	Specify a date range for which the stats should be collected using the following format: YYYY/MM/DD

Response Schema Fields

Schema Field	Type	Description

resultMap	Array	The result map includes a series of entries indicating how much file and mail data was stored in your installation on the last day of each month within the specified date range. Storage values are provided in KB.
		For example, the following entry indicates that at the end of August 2025, the installation include1,000 KB of file data and 22,162,724,941 KB of mail data:
		`"2025-08": {`
		`"fileSize": 1000,`
		`"mailSize": 2162724941`
		`}`

POST: Get Data License Stats (getDataLicenseStats)

Retrieve data licensing statistics for your ZL UA installation.

Path

http://localhost:8080/ps/api//v1/FileServer/getdatalicensestats

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
startDate **endDate**	String	Specify a date range for which the stats should be collected using the following format: YYYY/MM/DD

Response Schema Fields

Schema Field	Type	Description

resultMap	Array	The result map includes a series of entries indicating how many files and mails were included in your ZL UA installation. The entries are divided into **monthly** and **cumulative** values:
		• **monthly:** Each entry indicates how many files or mails were ingested into your installation during the specified month. This data will be pro-rated based on the date range. For example, if you specify a date range of 10-15-2025 to 12-15-2025, the stats for October would reflect the time between 10-15-2025 and 10-31-2025, and the stats for December would reflect the time between 12-1-2025 and 12-15-2025.
		• **cumulative**: Each entry indicates how many total files or mails had been ingested into your installation as of the last day of the specified month.
		Separate counts are provided for archived files, In-Place files, archived or journaled mails and In-Place mails. For example:
		`"2025-09": {`
		`"inplaceFileCount": 0,`
		`"mailCount": 34,`
		`"inplaceMailCount": 107103,`
		`"fileCount": 51751`
		`}`

GET: Get Server Using ID (getmailserverusingid)

Retrieve a server configuration. Specify the server by its ID.

Path

http://localhost:8080/ps/api/v1/FileServer/getmailserverusingid/{serverId}

Request Parameters

Parameter	Type	Description
serverId	Integer	Specify the ID of the server whose configuration is to be retrieved. You can retrieve server IDs with the GET: Get Severs endpoint described on page 230.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
id	Integer	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server sub-type.
url	String	The server URL, if applicable.
param	String	Additional information about the server.
discoveryName	String	The name of the Discovery object assigned to the server.
group	String	The server group.
dateCreate	String	The date the server was created.

GET: Get Server Using Name (getmailserverusingname)

Retrieve a server configuration. Specify the server by its name.

Path

http://localhost:8080/ps/api/v1/FileServer/getmailserverusingname/{serverName}

Request Parameters

Parameter	Type	Description
serverName	String	Specify the name of the server to be retrieved. You can retrieve server names with the GET: Get All Severs endpoint described on page 230.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
id	Integer	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server sub-type.
url	String	The server URL, if applicable.
param	String	Additional information about the server.
discoveryName	String	The name of the Discovery object assigned to the server.
group	String	The server group.
dateCreate	String	The date the server was created.

POST: Get Storage Stats (getStorageStats)

Retrieve data storage stats for your installation.

Path

http://localhost:8080/ps/api/v1/FileServer/getstoragestats

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
startDate **endDate**	String	Specify a date range for which the stats should be collected using the following format: YYYY/MM/DD

Response Schema Fields

Schema Field	Type	Description

resultMap	Array	The result map includes a series of entries indicating how much file and mail data was added to your installation during each month within the specified date range. Storage values are provided in KB.
		For example, the following entry indicates that in August 2025, no file data was added and 2,162,724,941 KB of mail data was added:
		`"2025-08": {`
		`"fileSize": 0`
		`"mailSize": 2162724941`
		`}`

POST: Get Tenant Data License Stats (getTenantDataLicenseStats)

Retrieve data licensing statistics for a specific tenant in your ZL UA installation.

Path

http://localhost:8080/ps/api/v1/FileServer/getTenantDataLicenseStats

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
startDate **endDate**	String	Specify a date range for which the stats should be collected using the following format: YYYY/MM/DD
tenantId	Integer	The tenant ID of the ZL server.

Response Schema Fields

Schema Field	Type	Description

resultMap	Array	The result map includes a series of entries indicating how many files and mails were included on the specified tenant during the date range. The entries are divided into **monthly** and **cumulative** values:
		• **monthly:** Each entry indicates how many files or mails were ingested on the specified tenant during the specified month. This data will be pro-rated based on the date range. For example, if you specify a date range of 10-15-2025 to 12-15-2025, the stats for October would reflect the period between 10-15-2025 and 10-31-2025, and the stats for December would reflect the period between 12-1-2025 and 12-15-2025.
		• **cumulative**: Each entry indicates how many total files or mails had been ingested on the specified tenant as of the last day of the specified month.
		Separate counts are provided for archived files, In-Place files, archived or journaled mails and In-Place mails. For example:
		`"2025-09": {`
		`"inplaceFileCount": 0,`
		`"mailCount": 34,`
		`"inplaceMailCount": 107103,`
		`"fileCount": 51751`
		`}`

PUT: Update Box Server (updateBoxServer)

Update a Box server's configuration.

Path

http://localhost:8080/ps/api/v1/FileServer/updateboxserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
serverName	String	Enter the name of the Box server. You can retrieve server names with the GET: Get All Severs endpoint, as described on page 230.
credential	String	Enter the name of the Credentials object required to access the Box server. This must be created with the UAA module. The **CloudCredentials** database table includes the Credential object names within your installation. For instructions on creating Credentials objects for use with Box servers, refer to the *ZL UA Box Archiving Administrator's Guide.*

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
server Type	String	The server type.
serverSubType	String	The server subtype.
url	String	The server URL.
param	String	Additional information about the server.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

PUT: Update an EWS Mail Server (updateewsmailserver)

Update the configuration of a Microsoft EWS mail server.

Path

http://localhost:8080/ps/api/v1/FileServer/updateewsmailserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the Microsoft EWS server to be updated. You can retrieve server IDs with the GET: Get All Severs endpoint, as described on page 230.
serverUrl	String	Specify the URL of the Microsoft EWS server.
exchangeServiceUrl	String	Use the following URL: https://outlook.office365.com/EWS/Exchange.asmx This is the service URL for O365 presently being used by Microsoft. ZL UA will use this URL to look up mailbox information for users and teams.
enableAutoDiscovery	Boolean	The appropriate setting for auto-discovery is detected and configured while creating the server. Set to True to enable it or False to disable it.
discoveryMethod	String	Enter the name of the Discovery object to use with the server. This must be created with the UAA module.
ewsCredential	String	Enter the name of the Credentials object required to access the server. This must be created with the UAA module. The **CloudCredentials** database table includes the names of the Credentials objects that have been added to your installation. For more information on creating Credentials objects for use with OneDrive servers, refer to the *ZL UA Unified Archival Administrator's Guide*.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubtype	String	The server subtype
url	String	The server URL.
param	String	Additional information about the server.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

PUT: Update File Server (updatefileserver)

Update a file server's configuration.

Path

http://localhost:8080/ps/api/v1/FileServer/updatefileserver/{serverId}

Request Parameters

Parameter	Type	Description
serverId	Integer	Specify the ID of the file server to be updated. You can retrieve server IDs with the GET: Get All Severs endpoint described on page 230.

Request Body Schema Fields

Schema Field	Type	Description
serverName	String	Specify the file server name. You can retrieve server names with the GET: Get All Severs endpoint described on page 230.
ip	String	Specify the server IP address.
port	String	Specify the port number used by the ZL File Connector. The default value is 9975. Leave the default value unless you are changing this port in the ZL File Connector as well.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server subtype
url	String	The server URL.
param	String	The server port.
discoveryName	String	The name of the Discovery Settings object on the server.

Schema Field	Type	Description
group	String	The server group.
dateCreate	String	The date the server was created.

PUT: Update a Google Drive Server (updategoogledriveserver)

Update a Google Drive server's configuration.

Path

http://localhost:8080/ps/api/v1/FileServer/updategoogledriveserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the Google Drive server to be updated. You can retrieve server IDs with the GET: Get All Severs endpoint described on page 230.
serverName	String	Enter the display name for the Google Drive server.
serverURL	String	Enter the URL of the Google Drive server.
adminUserToImpersonate	String	Enter the email address used to create the Google Drive service account for this server. This can be left blank if the service account has been directly added to shared drives. In this case, the service account created has to be present as a user in every shared drive that is to be crawled.

consumerCertFile	String	Specify the path and file name of the JSON key file downloaded when creating the Google Drive service account for this server. The file path entered must be relative to the server you are creating.
		For example, consider a ZL UA installation where there are 3 servers: Server1, Server2 and Server3. If the key file is on the D: drive of Server2 and you are creating the entry for Server2, you can enter the following file path:
		`D:\file.txt`
		However, if you are creating the entry for one of the other servers, you need to enter the server name as part of the file path:
		`\\server2\D$\file.txt`

scopes	String	Specify the service account scopes. Use these to set the change level of Google Drive API permission that can be assigned to ZL.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server subtype.
url	String	The server URL.
param	String	Additional information about the server.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

PUT: Update a Google Mail Server (updategooglemailserver)

Update a Google Mail server's configuration.

Path

http://localhost:8080/ps/api/v1/FileServer/updategoogledriveserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the Google Mail server to be updated. You can retrieve server IDs with the GET: Get All Severs endpoint described on page 230.
serverhost	String	Enter the hostname or IP address of the Google Mail server.
domainName	String	Enter the server domain name.
adminUserId	String	Specify the administrator user ID for the Google Mail server.
adminPassword	String	Specify the administrator password for the Google Mail server.
consumerKey	String	Specify the consumer key generated for the Google Mail server.
consumerSecret	String	Specify the secret key generated for the Google Mail server.
serviceAccountId	String	Enter the ID for the Google Mail service account created to administer this server.

Schema Field	Type	Description

certificateFile	String	Specify the path and file name of the certificate file for the Google Mail server. The file path entered must be relative to the server you are creating.
		For example, consider a ZL UA installation where there are 3 servers: Server1, Server2 and Server3. If the key file is on the D: drive of Server2 and you are creating the entry for Server2, you can enter the following file path:
		`D:\file.txt`
		However, if you are creating the entry for one of the other servers, you need to enter the server name as part of the file path:
		`\\server2\D$\file.txt`

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server subtype
url	String	The server URL.
param	String	Additional information about the server.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

PUT: Update a Lotus Domino Mail Server (updatelotusdominoserver)

Update a Lotus Domino mail server's configuration.

Path

http://localhost:8080/ps/api/v1/FileServer/updatelotusdominomailserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the server to be updated. You can retrieve server IDs with the GET: Get All Severs endpoint, as described on page 230.
parentServer	String	Enter the parent server URL. If specified, the connector URL, Notes password, and remote temp directory will be taken from the parent server. Alternately, enter **None** for the Parent Server field and specify these parameters below.
serverURL	String	Type the loop back address (127.0.0.1) or IP address of the machine that is running the ZL Domino TCP Connector.
notesAdminPassword	String	Type the password of the administrative service account you created in Notes.
remoteRouterHost	String	Type the name of the remote router if there is no local router.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
serverType	String	The server type.
serverSubType	String	The server subtype
url	String	The server URL.

Schema Field	Type	Description
param	String	Additional information about the server.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

PUT: Update an Exchange Mail Server (updatemailserver/exchange)

Update a Microsoft Exchange mail server's configuration.

Path

http://localhost:8080/ps/api/v1/FileServer/updatemailserver/exchange/{serverId}

Request Parameters

Parameter	Type	Description
serverId	Integer	Specify the ID of the Microsoft Exchange mail server to be updated. You can retrieve server IDs with the GET: Get All Severs endpoint described on page 230.

Request Body Schema Fields

Schema Field	Type	Description
serverCn	String	Enter the server's common name.
casServerName	String	Type the Client Access Server name. For O365, this should be set to match the SMTP address of the transport account for the server.
mapiClientHost	String	Enter the hostname or URL of the machine running the ZL MAPI Connector for this server.
exchangeServerVersion	String	Specify the Exchange server version by entering any of the following values: 2010 2013 2016 2019 O365
discoveryMethod	String	Enter the name of the Discovery object to use with the server. This must be created with the UAA module.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
server Type	String	The server type.
serverSubType	String	The server subtype.
url	String	The server URL.
param	String	Additional information about the server.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

PUT: Update a OneDrive Server's Configuration (updateonedriveserver)

Update a OneDrive server's configuration.

Path

http://localhost:8080/ps/api/v1/FileServer/updateonedriveserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
serverName	String	Specify the server name. You can retrieve server names with the GET: Get All Severs endpoint, as described on page 230.
credential	String	Enter the name of the Credentials object created for the server in UAA. The **CloudCredentials** database table includes the Credential object names within your installation. For more information on creating Credentials objects for use with OneDrive servers, refer to the *ZL UA OneDrive Archiving Administrator's Guide*.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
server Type	String	The server type.
serverSubType	String	The server subtype.
url	String	The server URL.
param	String	Additional information about the server.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

PUT: Update a SharePoint Server (updatesharepointserver)

Update a SharePoint server's configuration.

Path

http://localhost:8080/ps/api/v1/FileServer/updatesharepointserver

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the SharePoint server to be updated. You can retrieve server IDs with the GET: Get All Severs endpoint described on page 230.
serverName	String	Specify the server name.
serverUrl	String	Enter the IP address or host name of the host server on which you have installed the SharePoint connector.
sharePointFarmDbInstance	String	This information must be retrieved from the SharePoint server's SQL DB. Go to the SharePoint DB server, and select the database for SharePoint. Copy the name of this database and paste it into this text box.
credentials	String	Enter the name of the Credentials object created for the server in UAA. The **CloudCredentials** database table includes the Credential object names within your installation. For more information on creating Credentials objects, refer to the *ZL UA SharePoint Archiving Administrator's Guide*.
appOnly	Boolean	Set to True when using app-only authentication (a.k.a. application permissions) for this server. This should be set to True when you are using the SharePoint Online connector.

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.
name	String	The server name.
server Type	String	The server type.
serverSubType	String	The server subtype.
url	String	The server URL.
param	String	Additional information about the server.
discoveryName	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.
appOnly	Boolean	If True, it indicates that the server is using app-only authentication (a.k.a. application permissions). This should be set to True when you are using the SharePoint Online connector.

FAM/Disposition

Disposition is the process by which files whose records management lifecycle has expired are deleted and removed from ZL UA. The Disposition Workflow includes the following steps:

- Disposition is enabled on a project. When enabling the Disposition Workflow, the Project Administrator also schedules Disposition Runs. During each Disposition Run, the system determines which files in the project are eligible for disposition and automatically creates Disposition Approval Requests for those files.
- Disposition Approval Requests are generated for files whose records management lifecycle has expired to be removed after each Disposition Run.
- When the Disposition Approval Requests are approved, the Disposition Taks can be executed to remove the files.

For more comprehensive details on the Disposition Workflow, refer to the *ZL File Analysis and Management Administrator's Guide.* The following sections describe the FAM/Disposition endpoints available in the REST API. Use these endpoints to enable and disable disposition, and view disposition status:

- *POST: Approve Disposition (approvedisposition):* Approve a disposition run that has been executed upon a project.
- *PUT: Disable Disposition (disabledisposition)*: Enable disposition on a project.
- *PUT: Enable Disposition (enabledisposition)*: Disable disposition on a project.
- *GET: Get Latest Disposition Run (getlatestrun)*: Retrieve information for the latest Disposition Run executed upon a project.
- *GET: Get Projects Pending Approval (getprojectspendingforapproval)*: Retrieve a list of projects with pending Disposition Approval Requests.
- *GET: Get Projects Ready For Disposition (getprojectsreadyfordisposition*): Retrieve a list of projects with approved Disposition Approval Requests.
- *POST: Start Disposition (startdisposition)*: Start a Disposition Run.

POST: Approve Disposition (appprovedisposition)

Approve a disposition run that has been executed upon a project.

Path

http://localhost:8080/ps/api/v1/FileTreeDisposition/approvedisposition

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be updated. You can retrieve FAM project IDs for projects with Disposition Runs that are pending approval with the GET: Get Projects Ready for Approval endpoint described on page 259.
runId	Integer	Specify the Disposition Run ID. You can retrieve Disposition Run IDs with the GET: Get Latest Disposition Run endpoint described on page 258.
approvalComment	String	Enter any comments or descriptive information needed for the Disposition Run approval.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the disposition run was approved successfully.

PUT: Disable Disposition (disabledisposition)

Disable disposition for a project.

Path

http://localhost:8080/ps/api/v1/FileTreeDisposition/disabledisposition/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be updated. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether disposition was disabled successfully for the project.

PUT: Enable Disposition (enabledisposition)

Enable disposition for a project.

Path

http://localhost:8080/ps/api/v1/FileTreeDisposition/enabledisposition

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
idProj	Integer	Specify the ID of the project to be updated. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
dateStart **dateEnd**	String	Specify a date range during which Disposition Runs should be executed. A Disposition Run will be executed for the first time on the start date, and subsequent runs will be executed once a month until the end date is reached. Leave the **dateEnd** field empty to execute the Disposition Runs indefinitely. During each Disposition Run, the system determines which files in the project are eligible for disposition and automatically creates a Disposition Approval Request for those files. Specify the dates using the following format: YYYY-MM-DD

Response Schema Fields

A string indicating whether disposition was enabled successfully for the project.

GET: Get Latest Disposition Run (getlatestrun)

Retrieve information for the latest Disposition Run for a project. The response data includes information such as the name of the user who approved the Disposition Run and any comments entered when approving the request.

Path

http://localhost:8080/ps/api/v1/FileTreeDisposition/getlatestrun/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be viewed. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
projectId	Integer	The project ID.
runId	Integer	The Disposition Run ID.
approvalComment	String	The comment that was entered when the Disposition Run was approved.
statusMsg	String	A message describing the current status of the Disposition Run.
approverFullName	String	The name of the user who approved the Disposition Run.
approverAddress	String	The email address of the user who approved the Disposition Run.
dateStart **dateEnd**	String	The start date and end date for the Disposition Run.
approved	Boolean	Indicates whether the Disposition Run has been approved.

GET: Get Projects Pending Approval (getprojectspendingforapproval)

Retrieve the project IDs of projects that have pending Disposition Approval Requests.

Path

http://localhost:8080/ps/api/v1/FileTreeDisposition/getprojectspendingforapproval

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Project IDs: Lists the project ID for each project with a pending Disposition Approval Request as a comma-separated list.		

GET: Get Projects Ready For Disposition (getprojectsreadyfordisposition)

Retrieve the project IDs of projects that are ready for disposition, meaning that a Disposition Run has been approved.

Path

http://localhost:8080/ps/api/v1/FileTreeDisposition/getprojectsreadyfordisposition

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Project IDs: Lists the project ID for each project that is ready for Disposition as a comma-separated list.		

POST: Start Disposition (startdisposition)

Execute a Disposition Run. This deletes the files included in the most recently approved Disposition Approval Request and removes them from ZL UA.

Path

http://localhost:8080/ps/api/v1/FileTreeDisposition/startdisposition/{projectId}/{runId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be updated. You can retrieve FAM project IDs that are ready for disposition with the GET: Get Projects Ready for Disposition endpoint described in the previous section.
runId	Integer	Specify the Disposition Run ID. You can retrieve Disposition Run IDs with the GET: Get Latest Disposition Run endpoint described on page 258.

Request Schema Fields

None.

Response Schema Fields

A string indicating whether the Disposition Run was executed successfully.

FAM/Tasks

The ZL File Analysis and Management module includes background tasks that must be executed after the completion of certain operations. For example, after creating a project, you must execute the **Run Crawl**, **Update Content Index**, **Update Index** and **Clear Cache** background tasks to scan the project directories for files and update the project's content and tag indices.

The following sections describe the FAM/Tasks endpoints available in the REST API. Use these endpoints to initiate FAM background tasks:

- *POST: Close File Tree (closeFileTree):* Close the file tree for a project. You should consider closing the file tree for projects when you are running REST API calls on many projects at once, as this will remove the project's load from the server.
- *POST: Update Content Index (updateContentIndex)*: Update the project's content index, so that you can search the contents of any new files that have been added to the project. You can specify whether you want to update the content index for an entire project, or for a specific set of files returned by a search.
- *POST: Run Content Tagger (runcontenttagger)*: Run this task after uploading a tag specification to apply the tags defined in the specification to the files included in the project.
- *POST: Run Crawl (runcrawl*): Scan the project directories specified for a project and adds the contents to the project.
- *POST: Run Metadata Tagger (runmetadatatagger)*: Applies any metadata tag specifications that have been added to the project to the project files. A metadata tag specification defines a set of rules that will be checked against each file's metadata properties (e.g., date created, date last modified, date last accessed, ACL owner), and applies tags to the files matching those rules.
- *POST: Run PII Tagger (runpiitagger)*: Applies the tags defined in any PII tag specifications the project includes to the files included in the project.
- *POST: Run Remediation Task (runremediation)*: Execute the Run Remediation task on a project. All remediation actions configured for the project will be performed when you run this task.
- *POST: Update Index (updateindex):* Updates the tagging information for the files included in the project.

POST: Close File Tree (closeFileTree)

Close the file tree for a project. You should consider closing the file tree for projects when you are running REST API calls on many projects at once, as this will remove the project's load from the server.

Path

http://localhost:8080/ps/api/v1/FileTreeTask/closefiletree/{projectId}

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Response Schema Fields

A string indicating whether the file tree was closed successfully.

POST: Update Content Index (updateContentIndex)

Start the Update Content Index task for a project. This task updates the project's content index, so that you can search the contents of any new files that have been added to the project. You can specify whether you want to update the content index for an entire project, or for a specific set of files returned by a search or a file sampling operation.

Path

http://localhost:8080/ps/api/v1/FileTreeTask/updatecontentindex/{projectId}/FilesToIndex/{filesToIndex}

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
idProj	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
filesToIndex	String	Specify which files you want to update: • **indexAll:** The entire project. • **indexSavedSearches:** A saved search. • **indexSampledOnly:** Sampled files only.
searchNames	Array	If you specified **indexSavedSearches** for the **filesToIndex** field, specify the name(s) of the saved search(es) to update. The files returned by that search will be updated. If you want to update the files returned by multiple saved searches, specify the search names as a comma-separated list. For example: `"SearchNames": [ "sa1","sa2" ]`

Response Schema Fields

A string indicating whether the task was executed successfully.

POST: Run Content Tagger (runcontenttagger)

Run the Content Tagger task on a project. You can run this task after uploading a tag specification to apply the tags defined in the specification to the files included in the project.

Path

http://localhost:8080/ps/api/v1/FileTreeTask/runcontenttagger/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project you want to run the task on. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the task was executed successfully.

POST: Run Crawl (runcrawl)

Execute the Run Crawl task on a project. This task scans the project directories specified for the project and add the contents to the project.

Path

http://localhost:8080/ps/api/v1/FileTreeTask/runcrawl/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project you want to crawl. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the task was executed successfully.

POST: Run Metadata Tagger (runmetadatatagger)

Execute the Run Metadata Tagger task on a project. This applies any metadata tag specifications that have been added to the project to the project's files. A metadata tag specification defines a set of rules that will be checked against each file's metadata properties (e.g., date created, date last modified, date last accessed, ACL owner) and applies tags to the files matching those rules.

Path

http://localhost:8080/ps/api/v1/FileTreeTask/runmetadatatagger/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project you want to run the task on. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the task was executed successfully.

POST: Run PII Tagger (runpiitagger)

Execute the Run PII Tagger task on a project. This applies the tags defined in any PII tag specifications the project includes to the files included in the project. PII tag specifications are compared to each file's content to search specifically for files that include personal information such as credit card numbers, social security numbers, and addresses, and applies tags to the files matching the specification parameters.

Path

http://localhost:8080/ps/api/v1/FileTreeTask/runpiitagger/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project you want to run the task on. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the task was executed successfully.

POST: Run Remediation Task (runremediation)

In ZL UA, you can assign an action (e.g., copy to a new folder, delete, archive, etc) to a tag, so that the action can be carried out upon all the files that the tag has been applied to at once. This is referred to as a remediation. Use this endpoint to execute the Run Remediation task on a project. All remediation actions configured for the project will be performed when you run this task.

Path

http://localhost:8080/ps/api/v1/FileTreeTask/runremediation/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project you want to run the task on. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the task was executed successfully.

POST: Update Index (updateindex)

Execute the Update Index task on a project. This updates the tagging information for the files included in the project.

Path

http://localhost:8080/ps/api/v1/FileTreeTask/updateindex/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project you want to run the task on. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the task was executed successfully.

UAA/License

This section describes the POST: Get User License Report endpoint.

POST: Get User License Report (getUserLicense)

Use this endpoint to retrieve licensing information regarding the mailing lists and users within your ZL UA installation.

Path

http://localhost:8080/ps/api/v1/license/getUserLicense

Request Parameters

Parameter	Type	Description
page **pageSIze**	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.

Request Body Schema Fields

Schema Field	Type	Description
idTenant	Integer	The tenant ID of the ZL server.
dateStart **dateEnd**	String	Specify a start date and an end date (inclusive). Licensing data for this date range will be included in the response .

Response Schema Fields

Schema Field	Type	Description
dateStart **dateEnd**	String	The date range specified in the request schema.
activeUserCount	Integer	The number of active users within the specific tenant within the specified tenant.
terminatedMailingListCount	Integer	The number of terminated mailing lists within the specified tenant.
mailingListWithoutDataCount	Integer	The number of mailing lists that do not have data within the specified tenant. This means that no mails sent to or from the members of those mailing lists have been indexed in ZL UA.
activeMailingListCount	Integer	The number of active users within the specified tenant.
terminatedUserCount	Integer	The number of terminated users within the specified tenant.
userWithoutDataCount	Integer	The number of users that do not include data within the specified tenant. This means that no mails sent to or from those users have been indexed in ZL UA.

Workspace/Preservation

When running a search to build a workspace, you have the option to save the messages or files returned by the search as a *preservation* and place them on *legal hold*. Once messages or files have been placed on legal hold, they cannot be removed from ZL UA until the legal hold is removed. This section describes the endpoints you can use to create and manage preservations within your workspaces.

Before proceeding, please review the following summary of the preservation workflow. The names applied to the various tasks that are involved in this workflow are used in the names of the endpoints and request and response fields described later in this section:

- When you run a search to create a workspace or add data to an existing workspace, that search is referred to as a *workspace data source*. You can preserve the contents of each individual data source.
- After you create a workspace, you can run *filter searches* or create *Dataset Manipulations* to search the contents of the workspace. You can preserve the results of these searches as well.
- When you preserve the contents of a data source, filter search or Dataset Manipulation that includes In-Place data, ZL initiates a *selective archive search* to archive the In-Place items into the ZL Vault.
- After you create a preservation, you can search its contents. This is referred to as a *preservation search*. Preservation searches can be saved for later use.

Use the following endpoints to create and manage preservations within your workspaces:

- *DELETE: Delete Preservation Saved Search (deletePreservationSavedSearch)*
- *GET: Get All Selective Archive Searches Using Workspace ID (getAllSelectiveArchiveSearchesUsingId)*
- *GET: Get Latest Preservation Task Status (getLatestPreservationTaskStatus)*
- *POST: Get Workspace Preserved Data Source View (getWorkspacePreservedDataSourceView)*
- *POST: Get Selective Archive Search Items (getSelectiveArchivingSearchItems)*
- *GET: Get Selective Archive Search Statistics (getSelectiveArchivingSearchStatsUsingId)*
- *POST: Run Preservation Search (doPreservationSearch)*
- *POST: Preserve Data Source (preserveDataSource)*
- *POST: Preserve Filter Search or Data Set Manipulation (preserveSearchReslts)*

DELETE: Delete Preservation Saved Search (deletePreservationSavedSearch)

Delete a saved preservation search.

Path

http://localhost:8080/ps/api/v1/preservation/deletepreservationsavedsearch/{workspaceId}/{searchName}

Request Parameters

Parameter	Type	Description
workspaceId	Integer	The ID of the workspace containing the preservation search to be deleted. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.
searchName	String	The name of the preservation search to be deleted. You specify the preservation search name when you create it with the POST: Run Preservation Search (doPreservationSearch) endpoint described on page 284. You can retrieve preservation search names with the GET: Get Search Tree endpoint described on page 55. When doing so, please note that the GET: Get Search Tree requires a case ID as input. Do not enter the workspace ID for this input. You must enter the ID of the case attached to the workspace (a case is created internally when data is preserved within the workspace). You can retrieve the ID of the case attached to a workspace from the **CaseWorkspacePreservation** table. To do so, find the entry for the workspace by checking for the workspace ID in the **cwpWorkspaceId** column of the **CaseWorkspacePreservation** table. The **cwpCaseId** column for this entry will lists the ID of the case attached to the workspace.

Request Body Schema Fields

None.

Response Schema Fields

The endpoint returns a string indicating whether the preservation search was deleted successfully.

GET: Get All Selective Archive Searches Using Workspace ID (getAllSelectiveArchiveSearchesUsingId)

Retrieve the names and ID values assigned to the selective archive searches within the workspace. When you preserve the contents of a data source, filter search or Dataset Manipulation that includes In-Place data, ZL initiates a selective archive search to add the In-Place items into the ZL Vault.

Path

http://localhost:8080/ps/api/v1/preservation/getallselectivearchivesearchesusingid/{workspaceId}

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	
workspaceId	Integer	The ID of the workspace containing the selective archive searches to be viewed. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.

Request Schema Fields

None.

Response Schema Fields

The response data includes the following details for each selective archive search found within the workspace.

Schema Field	Type	Description
searchId **dataSourceId**	Integer	The **searchId** represents the ID value assigned to the selective archive search. The **dataSourceId** is the ID assigned to governance result for the selective archive search. The governance result is created as part of the process of running the selective archive search and adding the In-Place data to the ZL Vault. This value is stored internally in the S**electiveArchiveSearch** database table in the **sasUcontextDataSourceId** column.
searchName	String	The name assigned to the selective archive search.

GET: Get Latest Preservation Task Status (getLatestPreservationTaskStatus)

Retrieve the status of the most recent preservation-related task that has been executed within a workspace.

Path

http://localhost:8080/ps/api/v1/preservation/getLatestPreservationTaskStatus/{workspaceId}

Request Parameters

Parameter	Type	Description
workspaceId	Integer	The ID of the workspace to be viewed. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.

Request Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
status	String	The task status.
statusMsg	String	Additional information about the task status.
startTime	Integer	The time that the task started. These times are delivered as epoch timestamps. For more information on epoch timestamps, refer to: https://www.epochconverter.com/
taskName	String	The task name.
pid	String	The process ID assigned to the task.
currentPhase **totalPhases**	Integer	The total number of phases included in the task, and the phase that the task is currently performing.
doneDate	String	The date and time the task completed.

Schema Field	Type	Description
statusMap	Array	Additional status information about the task. The information displayed here varies depending on the task type, but each **additionalProp** element generally includes descriptive information about the task, such as the task name, its processing status, and other details such as the number of items affected by the task.
counters	Array	The **counters** array includes the following information regarding the items affected by the task. It may have one or more entries, depending on which task is performed:
key	String	A description of the counter. For example, the Workspace Preservation task includes an **Items Found** key which indicates how many items were found by the task.
value	Integer	The value associated with the task key.
phases	Array	The **phases** array includes an entry for each phase included within the task. Each entry includes the following information:
phase	String	The phase ID represents the index assigned to the phase, corresponding to the start and end timestamps of the phase.
status	String	The phase status includes information such as a timestamp indicating when the task started, and the overall status of the phase.
props	Array	Each **prop** entry is map of key value pairs corresponding to the name and value of the corresponding properties of a background task.

POST: Get Workspace Preserved Data Source View (getWorkspacePreservedDataSourceView)

Retrieve a list of the items included within a preserved data source.

Path

http://localhost:8080/ps/api/v1/preservation/getWorkspacepreserveddatasourceview

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	The ID of the workspace containing the preserved data source to be viewed. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.
stDisplayName	String	The display name to assign to the items included the preserved data source. This is used internally to process the request data.
stViewId	String	The view ID assigned to the data set you are viewing. Leave this value null the first time you run the endpoint, and use the view ID that is returned in the response the next time you run it. This way if the endpoint returns multiple pages of data, you only need to update the **start** page the next time you run the endpoint to view the next page of search results.
start	Integer	The starting index for the search results.
itemsPerPage	Integer	The number of search results to be returned per page.
searchName	String	Specify the ID of the preserved data source that you want to view. You can retrieve the workspace's data source IDs from the **CaseWorkpacePreservation** database table. To do so, find the entry (or entries) for the workspace by checking for the workspace ID in the **cwpWorkspaceId** column. The **cwpCaseDataSourceId** column for each of the workspace's entries will list the ID of a data source that has been added to the workspace.

Response Schema Fields

Schema Field	Type	Description
items	Array	The **items** array includes an entry for each file or message included in the preserved data source. Each entry includes the following fields:
rowID	Integer	Internal use only.
viewItemId	Integer	The view ID assigned to the item.
refItemId	Integer	The reference ID assigned to the item.
size	Integer	The size of the item, in KB or MB.
relevancy	Integer	The relevancy rating assigned to the item. The relevancy rating is an indication of how well the item matched the search criteria used to build the data source. A low relevancy rating indicates the message matched a greater part of the search criteria (e.g., items that are assigned a relevancy rating of 100.0 matched the most search criteria). **Please note that relevancy scores are a Beta Level feature as of the release of ZL UA 11.1.2.**
type	Integer	The type of the file or message.
deletedSearchItem	Boolean	If True, it indicates that the item was retrieved from a search store for deleted items.
viewId	Integer	The view ID assigned to the search result set.
totalRecords	Integer	The number of items included in the search results.
itemsPerPage	Integer	The number of search results to be returned per page.
isfHasAll	Boolean	For internal use only.
isfCachedView	Boolean	For internal use only.
totalDisplayRecords	Integer	The number of items displayed in the search results.
totalRows	Integer	The number of rows included in the search results.

POST: Get Selective Archive Search Items (getSelectiveArchivingSearchItems)

When a data source or search result set is preserved, ZL initiates a selective archiving search to add the In-Place items included in the data source or search result set into the ZL Vault. Use this endpoint to view the items that were included in a selective archiving search.

Path

http://localhost:8080/ps/api/v1/preservation/getselectivearchivingsearchitems

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
Start	Integer	The starting index for the search results.
itemsPerPage	Integer	The number of search results to be returned per page.
saSearchId	String	The search ID assigned to the selective archive search you want to view. You can use the GET: Get All Selective Archive Searches Using Workspace ID (getAllSelectiveArchiveSearchesUsingId) endpoint described on page 275 to retrieve the ID values of the selective archive searches that have been initiated within a workspace.
status	Integer	Specify a status value to return items that have been assigned that status: • 100: Not yet processed • 199: Error downloading • 300: Staged • 400: Archived • 500: Deleted in source • 600: Preserved

Response Schema Fields

Schema Field	Type	Description
resultMap	Array	A list of the reference IDs assigned to the items included in the selective archive search.
totalRecords	Integer	The number of items included in the selective archive search.
totalDisplayRecords	Integer	The number of displayed items included in the selective archive search. This typically matches the value of the **totalRecords** field.

GET: Get Selective Archive Search Statistics (getSelectiveArchivingSearchStatsUsingId)

When a data source or search result set is preserved, a selective archiving search is initiated to archive the In-Place items included in the data source or search result set into the ZL Vault. Use this endpoint to retrieve status information for the items included in a selective archive search.

Path

http://localhost:8080/ps/api/v1/preservation/getselectivearchivingsearchstatsusingid/{workspaceId}/{searchId}

Request Parameters

None.

Request Schema Fields

Schema Field	Type	Description
workspaceId	Integer	The ID of the workspace containing the selective archive search to be viewed. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.
searchId	Integer	The ID of the selective archive search to be viewed. You can use the GET: Get All Selective Archive Searches Using Workspace ID endpoint described on page 275 to retrieve the IDs of the selective archive searches that have been initiated within a workspace.

Response Schema Fields

The endpoint response indicates how many files or messages within the selective archive search are assigned each status value.

Schema Field	Type	Description
status	Integer	The ID assigned to the status: • 100: Not yet processed • 199: Error downloading • 300: Staged • 400: Archived • 500: Deleted in source • 600: Preserved
count	Integer	The number of files or messages in the selective archive search that have been assigned the status value.
name	String	The name of the selective archive search.

POST: Run Preservation Search (doPreservationSearch)

Search the contents of a preservation that has been added to a workspace.

Path

http://localhost:8080/ps/api/v1/preservation/dopreservationsearch

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	The ID of the workspace containing the preservation to be searched. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.
searchName	String	Enter the name of the preservation search. This is only required when **quickSearch** is set to False, meaning that you want to save the search for future reference.
quickSearch	Boolean	Set to True to run the search and view its results without saving the search results. Set to False to save the input search criteria as a saved preservation search, to be viewed both now and at a later point in time in the Enterprise Analytics module's Data Explorer. When you choose this option, you must also set the **autorun** field to True, and configure the **searchName** field.
autorun	Boolean	Set to True to create a custom search that will be saved and accessible indefinitely in the Data Explorer's **Custom** node. Otherwise, Set to False to create a temporary search for viewing purposes.
viewId	String	The view ID assigned to the data set you are viewing. Leave this value null the first time you run the endpoint, and use the view ID that is returned in the response the next time you run it. This way if the endpoint returns multiple pages of data, you only need to update the **start** page the next time you run the endpoint to view the next page of search results.
start	Integer	The starting index for the search results.
itemsPerPage	Integer	The number of search results to be returned per page.

Parameter	Type	Description
searchParamsDTO	Use the remaining fields to filter the preservation.	
sourceId	String	To restrict the search to documents that were saved to the workspace via a specific data source (i.e., a search that was used to create or add data to the workspace, specify the data source ID assigned to the search here. You can specify multiple data source IDs as a comma-separated list, if desired. You can use the POST: Get Workspace Data Sources Using Workspace ID endpoint described on page 457 to retrieve the ID values for all data sources that have been added to a particular workspace in a single operation.
rawFrom **searchFromRawRcptOperator** **rawRcpt**	String	**Not applicable to files.** For the **rawFrom** field, enter a comma-separated list of email addresses to limit the search results to messages sent by the specified email addresses. For the **rawRcpt** field, enter a comma-separated list of email addresses to limit the search results to messages sent to the specified email addresses. For the **searchFromRawRcptOperator** field, enter either **or**, **and** or **between** to determine how the email addresses specified for the **rawFrom** and **rawRcpt** fields will be used to filter the search results. For example, you could specify **between** to filter the search to return messages sent between these addresses. Or you could specify **or** to return messages that were either sent to the addresses specified by the **rawRcpt** field, or were sent from the addresses specified by the **rawFrom** field.
startDate **endDate**	String	To restrict the search to a specific date range, enter the start date and end date for the date range. The search will return messages that were sent or received (or files that were last modified) during this date range, These dates are inclusive, meaning that messages sent or received (or files that were last modified) on the start and end dates will be included in the search results.
srchDesc **searchName**	String	Enter a name and description of the search. Required for saved searches (i.e., searches where the **isQuick** field is set to False).

Parameter	Type	Description
isfQuick	Boolean	Set to True if you are running a quick search, or set to False if you are running a search that will be saved for future reference. If you run a saved search, use the **srchName** and **srchDesc** fields desceibed previously to specify the name and description of the search.
subject	String	Enter the words or phrases that must appear in the message subject or file name for a case document to be returned

Response Schema Fields

Schema Field	Type	Description
items	Array	The **items** array includes an entry for each file or message that was included in the preservation. Each entry includes the following fields:
rowID	Integer	Internal use only.
viewItemId	Integer	The view ID assigned to the item.
refItemId	Integer	The reference ID assigned to the item.
size	Integer	The size of the item, in KB or MB.
relevancy	Integer	The relevancy rating assigned to the item. The relevancy rating is an indication of how well the item matched the search criteria used to build the data source. A low relevancy rating indicates the message matched a greater part of the search criteria (e.g., items that are assigned a relevancy rating of 100.0 matched the most search criteria). **Please note that relevancy scores are a Beta Level feature as of the release of ZL UA 11.1.2.**
type	Integer	The type of the file or message.
deletedSearchItem	Boolean	If True, it indicates that the item was retrieved from a search store for deleted items.
viewId	Integer	The view ID assigned to the search result set.
totalRecords	Integer	The number of items returned by the search.
totalDisplayRecords	Integer	The number of items displayed in the search results.
totalRows	Integer	The number of rows included in the search results.

Schema Field	Type	Description
itemsPerPage	Integer	The number of search results to be returned per page.
isfHasAll	Boolean	For internal use only.
isfQuickSearch	Boolean	Indicates whether the search is a quick search (True) or a saved search (False).
metaDataQuery	String	A string representing the search query.
isfCachedView	Boolean	For internal use only.

POST: Preserve Data Source (preserveDataSource)

Preserve a data source. A data source is a search that was used to create or add data to a workspace.

Please note that you must execute the **Selective Archive** and **Case Importer** global tasks in the SysAdmin module after executing the endpoint to complete the process of creating the preservation and placing the files on legal hold. These global tasks cannot be executed from the REST API. For more information, contact your System Administrator or refer to the *ZL UA Workspace Administrator's Guide*.

Path

http://localhost:8080/ps/api/v1/ws/preservation/preserve/{workspaceId}/{dataSourceId}

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
workspaceId	Integer	The ID of the workspace containing the data source to be preserved. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.
datasourceId	String	Specify the ID of the data source you want to preserve. You can use the POST: Get Workspace Data Sources Using Workspace ID endpoint described on page 457 to retrieve the ID values for all data sources that have been added to a particular workspace in a single operation.

Response Schema Fields

Schema Field	Type	Description
error	If errors occurred, the **message** and **exception** strings provide information describing them.	
success	Boolean	Indicates whether the task was submitted successfully (True) or not (False).
result	String	The result of the request. If **Success** is set to True, a message will display indicating that the data source has been submitted for preservation.

POST: Preserve Filter Search or Data Set Manipulation (preserveSearchResults)

Preserve a filter search or Data Set Manipulation that was performed within a workspace.

You must execute the **Selective Archive** and **Case Importer** global tasks in the SysAdmin module after executing the endpoint to complete the process of creating the preservation and placing the files on legal hold. These global tasks cannot be executed from the REST API. For more information, contact your System Administrator or refer to the *ZL UA Workspace Administrator's Guide*.

Path

http://localhost:8080/ps/api/v1/ws/preservation/preserve/{workspaceId}

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
workspaceId	Integer	The ID of the workspace containing the search or data source to be preserved. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.
viewId	String	The view ID assigned to the filter search or Dataset Manipulation. View IDs are returned after you perform a search on a workspace with endpoints such as POST: Workspace Dataset Manipulation Search and POST: Workspace Filter Search. For more information, refer to *Workspace/Search* on page 490

Response Schema Fields

Schema Field	Type	Description
error	If errors occurred, the **message** and **exception** strings provide information describing them.	
success	Boolean	Indicates whether the task was submitted successfully (True) or not (False).
result	String	The result of the request. If **Success** is set to True, a message will display indicating that the data source has been submitted for preservation.

.

Global Search

This section describes endpoints you can use to perform Global Searches in ZL UA:

- *POST: Global Search on Archive Files Store (searchFiles) and File In Place Store (searchFilesInplace)*
- *POST: Global Search on Archived/Journaled Mails Store (searchMails) and Mail In Place Store (searchMailsInplace)*

POST: Global Search on Archive Files Store (searchFiles) and File In Place Store (searchFilesInplace)

Use the **searchFiles** endpoint to perform a Global Search on the **Archive File Store**, or the **searchFilesInplace** endpoint to perform a Global Search on the **In-Place File Store**. The input required for these endpoints is the same for archived files as it is for In-Place files. However, there are some variances in the information that will be included in the response data for archived and In-Place files. Further details are provided in the *Response Schema Fields* section.

Path

Archived Files: http://localhost:8080/ps/api/v1/search/files
In-Place Files: http://localhost:8080/ps/api/v1/search/files/inplace

Request Parameters

None.

Request Schema Fields

Schema Field	Type	Description
viewId	String	Leave empty.
viewType	String	Leave empty.
sort	String	Specify the attribute by which the list of files search results should be sorted: • **name** • **createDate** • **size** • **relevancy** • **sort_dummy** • **lastModified** • **lastAccessed** Use the **sortDir** field to specify whether the search results should be sorted in ascending or descending order.
sortDir	String	Specify **ASC** or **DESC** to indicate the order (ascending or descending) that should be used to sort the list of files in the search results.

Schema Field	Type	Description
start **length**	Integer	Some operations return too many values for the UI to display. You can use the **start** parameter to indicate the page number on which these values should be displayed, and the **length** parameter to indicate the maximum number of items that can be displayed on the page.
fileSearch: Use the following fields to filter the search results.		
recursive	Boolean	Reserved. Leave set to True.
Use the following fileSearch fields to filter the search by file name.		
fileNameSearchConditionType		To filter the search results by file name, specify a value to determine how the file names will be searched: • **2:** Return files where **all the words** specified within the **fileName** field are included in the file name. • **3**: Return files where **any of the words** specified within the **fileName** field are included in the file name. • **7:** Return files where the file name **exactly matches** the string specified as the **fileName** field. Leave this field empty if you do not want to filter the search results by file name.
filename	String	To filter the search results by file name, enter a word (or group of words). The search will return files that include some (or all) of the words in their file name. Use the **fileNameSearchConditionType** field to determine how the file names will be searched. Leave this field empty if you do not want to filter the search results by file name.

Schema Field	Type	Description
Use the following fields to filter the search by included file content. Files that include the phrases or keywords specified by these fields within their contents will be <u>included in</u> the search results.		
searchContentConditionType	Integer	To filter a search by included file content, specify a value to determine how the file contents will be searched: • **1:** Return files where the **exact phrase** specified within the **searchContent** field is included in the file content. • **2:** Return files where **all the words** specified within the **searchContent** field are included in the file content. • **3**: Return files where **any of the words** specified within the **searchContent** field are included in the file content. • **4**: Run a proximity search. The search will return files where the words specified within the **searchContent** field occur within a certain proximity of each other (e.g., within 10 words of each other). In this case, use the **searchContentTypeProximityWindow** field to determine the proximity requirement. Leave this field empty if you do not want to filter the search results by file content.
searchContent	String	To filter the search results by file content, enter a word (or group of words). The search results will include files that **include** some (or all) of the specified word(s) in their contents. Use the **searchContentConditionType** field to determine how the file content will be searched for the words. Leave this field empty if you do not want to filter the search results by file content.
searchContentTypeProximityWindow	Integer	You only need to configure this setting when you set the **searchContentConditionType** field to 4 to run a proximity search on the file contents. The search will return files where the words specified within the **searchContent** field appear within this number of words of each other. For example, if you use the default value of 10 and the **searchContent** field is set to guarantee rebate, the search would return files where *guarantee* and *rebate* occur within 10 words of each other.

Schema Field	Type	Description
Use the following fields to filter the search by excluded file content. Files that include the phrases or keywords specified by these fields within their contents will be excluded from the search results.		
searchContentExcludeOn	Boolean	Set to True to filter the search so that files that include specified phrases or keywords in their contents will be **excluded from the search results**. In this case, use the **searchContentExclude** field to specify the words to search for, and the **searchContentExcludeConditionType** field to determine how the file contents will be searched for those words. Set to False if you do not want to filter the search in this manner.
searchContentExcludeConditionType	Integer	If **searchContentExcludeOn** is set to True, use this field to determine how the file contents will be searched to determine which files should be excluded from the search results: • **1:** Exclude files where the **exact phrase** specified within the **searchContentExclude** field is included in the file contents. • **2:** Exclude files where **all the words** specified within the **searchContentExclude** field are included in the file contents. • **3**: Exclude files where **any of the words** specified within the **searchContentExclude** field are included in the file contents. • **4**: Run a proximity search. The search results will exclude files where the words specified within the **searchContentExclude** field **do not** occur within a certain proximity of each other (e.g., within 10 words of each other). In this case, use the **searchContentTypeProximityWindow** field to determine the proximity requirement
searchContentExclude	String	If **searchContentExclueOn** is set to True, enter a word (or group of words) to filter the search by excluded file content. Files that include some (or all) of the specified word(s) will be excluded from the search results. Use the **searchContentExcludeConditionType** field to determine how the file contents will be searched for the words.

Schema Field	Type	Description
searchContentTypeExcludeProximityWindow	Integer	Configure this setting when you set the **searchContentExcludeConditionType** field to 4 to run a proximity exclusion search on the file contents. The search will exclude files where the words specified within the **searchContentExclude** field appear within this number of words of each other. For example, if you use the default value of 10 and the **searchContentExclude** field is set to guarantee rebate, the search results would exclude files where *guarantee* and *rebate* occur within 10 words of each other.
Use the following fields to search for files that were created during a specific time frame.		
searchCreateDateStart **searchCreateDateEnd**	String	Specify the start and end date and times for the file creation date range.
searchUseGMT	Boolean	Set to True to use GMT time zone for the file creation date time frame. Set to False to use the local server time.
The following fields are not supported. Leave these fields empty, or use the default values.		
searchFileExtension	Boolean	Leave empty.
searchAttachmentNameType	Integer	Use default value (3).
searchAttachmentName	Integer	Leave empty.
searchAttachmentContentType	Integer	Use default value (3).
searchAttachmentContent	Boolean	Leave empty.
searchAttachmentContentTypeProximityWindow	Integer	Use default value (10).
searchAttachmentContentExcludeOn	Integer	Use default value of False.
searchAttachmentContentTypeExclude	Integer	Use default value (3).
searchAttachmentContentExclude	Boolean	Leave empty.

Schema Field	Type	Description
searchAttachmentContentTypeExcludeProximityWindow	Integer	Use default value (10).
searchAttachmentFileExtension	Boolean	Leave empty.
Use the following fields to filter the search based on various file attributes such as the file size or flag status.		
searchFileSizeLow **searchFileSizeHigh**	Integer	To filter the search by file size, specify a size range to search for (in KB). Use **searchFileSizeLow** to specify the lower limit and **searchFileSizeHigh** to specify the upper limit. You can leave either field blank to search for files that are either larger than the size specified for **searchFileSizeLow,** or lower than the size specified for **searchFileSizeHigh**.
searchFileFlags	String	Enter any of the following values to filter the search by the status of file flags: • **SAF_NOTSCANNED**: Not scanned • **SAF_NOTSCANNEDPOLICY**: Not scanned by policy • **SAF_CONTAINER**: Container policy • **SAF_TRUNCATEDTEXTCONTENT**: Text not completely indexed due to a large number of tokens
The following fields are not supported. Leave these fields empty, or leave the default values.		
searchOwner	N/A	Leave empty.
searchFileAclName	N/A	Leave empty.
searchExplicitAclOnly	N/A	Leave empty.
searchAclOwnerName	Boolean	Use default value (False).

Schema Field	Type	Description
Use the remaining fields to filter the search based on various file and project attributes.		
searchProjectId	Integer	Specify the IDs of the projects you want to search as a comma-separated list. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338. Make sure to specify projects that are of the project type specified by the **projectType** field described below.
searchMessageCategoryId	Integer	Reserved. Leave empty.
includeDeletedProjects	Boolean	Set to True if any of the projects specified in the **searchProjectId** field have been deleted. Otherwise, set to False.
projectType	Integer	Specify the type of project to search: • **1**: File Share • **3**: SharePoint • **10**: OneDrive
searchDomain	Integer	Reserved. Set to 1.
searchRawQuery	Integer String	If the **searchMode** field is set to 2 to run a **Raw Query Search,** specify the raw query for the search you want to run. In this case, leave the rest of the **FileSearch** inputs empty, and use the **searchRawQuery** field to enter the raw query you want to run. For details on raw query search syntaxes, refer to the *Raw Query File Searches* section of the *ZL UA Raw Query Search User's Guide*.

Response Schema Fields

Schema Field	Type	Description
domainId	Integer	Not applicable to Global Searches for files.
items	Array	The **items** array includes an entry for each file returned by the search. Each entry includes the following fields:
deletedSearchItem	Boolean	If True, it indicates that the item was retrieved from a search store for deleted files.
refItemId	String	The reference ID assigned to the file.
relevancy	String	The relevancy rating assigned to the file. The relevancy rating is an indication of how well the file matched the search criteria used to conduct the search. A low relevancy rating indicates the file matched a greater part of the search criteria (e.g., files that are assigned a relevancy rating of 100.0 matched the most search criteria). **Please note that relevancy scores are a Beta Level feature as of the release of ZL UA 11.1.2.**
size	Integer	The file size.
type	Integer	The file type.
viewId	String	The view ID assigned to the search.
viewItemId	String	The view ID assigned to the file.
category	String	The lexical category assigned to the archived file. **This field is not applicable to In-Place files.**
createDate **dateLastAccessed**	String	The dates the file was created and last accessed.
fileName	String	The file name.
fileOwner	String	The file owner. **This field is not applicable to archived files.**
folder	String	The folder location.
itemId	String	The ID assigned to the file.
itemType	String	A string describing the file type.

Schema Field	Type	Description
lastModifiedDate	String	The date the file was last modified.
mapFlags	Boolean	For archived files, the **mapFlags** array is a series of Boolean values indicating the status of various file attributes. For example, **saf_record** will be set to True if the file has been declared a record and **saf_legalhold** will be set to True if the file has been placed on legal hold. For In-Place files, the **mapFlags** array is a series of Boolean values indicating the status of In-Place file attributes. For example, **ipfitemflags_reccatmanualapplied** will be set to True if the file's Record Category was applied manually.
projectId	Integer	The ID the project the file belongs to.
recordCategory	Array	**The recordCategory array is only applicable to In-Place files.** The **recordCategory** array includes a series of fields identifying the Record Category that has been applied to the file (e.g., the name, ID, restricted filing status, retention code ID). These fields were defined when the Record Category was created with the POST: Create Record Category endpoint. For further details on each field, refer to *POST: Create Record Category (createRecordCategory)* on page 193.
staticTags	Array	**The staticTags array is only applicable to In-Place files.** The **staticTags** array includes an entry for each Static Tag that has been applied to the file. Each entry includes details such as the tag's name, display name**,** and ID. If the **deleted** Boolean is set to True, it indicates that the tag has been deleted. This means that it cannot be applied to additional files, but can still be used to search for files it was applied to before its deletion.
versionId	Integer	The version ID assigned to the archived file. **This field is not apppliable to In-Place files.**
itemsPerPage	Integer	The number of files per page.
rawQuery	String	A representation of the search parameters specified.
searchStoreId	Integer	The ID assigned to the search store used to run the search.

Schema Field	Type	Description
totalRecords	Integer	The number of files returned by the search.
viewId	Integer	The view ID assigned to the search.

POST: Global Search on Archived/Journaled Mails Store (searchMails) and Mail In Place Store (searchMailsInplace)

Use the **searchMails** endpoint to perform a Global Search on the **Archived or Journaled Mail Stores**, or the **searchMailsInplace** endpoint to perform a Global Search on the **In-Place Mail Store**. Most of the input required for these endpoints is the same. However, there are some variances in the information that will be included in the response data for archived and In-Place messages. Further details are provided in the *Response Schema Fields* section.

Path

Archived and Journaled Mails Store: http://localhost:8080/ps/api/v1/search/mails
Mail In Place Store: http://localhost:8080/ps/api/v1/search/mails/inplace

Request Parameters

None.

Request Schema Fields

Schema Field	Type	Description
viewId	String	Leave empty.
viewType	String	Leave empty.
sort	String	Specify the attribute by which the messages included in the search results should be sorted: • **date** • **size** • **relevancy** • **sort_dummy** Use the **sortDir** field to specify whether the search results should be sorted in ascending or descending order.
sortDir	String	Specify **ASC** or **DESC** to indicate the order (ascending or descending) that should be used to sort the list of messages in the search results.

Schema Field	Type	Description
start **length**	Integer	Some operations return too many values for the UI to display. You can use the **start** parameter to indicate the page number on which these values should be displayed, and the **length** parameter to indicate the maximum number of items that can be displayed on the page.
mailSearch: Use the following fields to filter the search results.		
searchDataType	Integer	Set to 1 to search for mails.
trac	Boolean	Reserved. Set to False.
searchMode	Integer	Use the **searchMode** field to specify the search type: • **1:** Run a standard search using the search filter options that are available as **mailSearch** inputs. In this case, use the other search filter inputs described in this table to define the search filter criteria. Make sure to set the **searchRawQuery** field to: ***:*** • **2: Raw Query Search**. Create a raw query for an advanced search. In this case, leave the rest of the **mailSearch** inputs described in this table empty, and use the **searchRawQuery** field to enter the raw query you want to run. For details on raw query search syntaxes, refer to the *Raw Query Email Searches* section of the *ZL UA Raw Query Search User's Guide*.

Schema Field	Type	Description
searchDomain	Boolean	To filter the search to include messages that were sent between users of a specific department, specify the department's domain ID here: • **1**: Root • **3**: Unknown Users • **5**: Review Escalation • **11**: External Users • **13**: Federated Search • **15**: Secure Mail Users By default, the **searchDomain** field is set to 1 to search the root department. Enter additional departments to search by providing the IDs as a comma-separated list. For example, if you want to search Root and External Users, you would enter: 1,11 You could also exclusively search the External Users department by specifying 11 here. Use the **recursive** field described later to determine whether the specified department(s) will be searched recursively (meaning that child departments would be included in the search results).
searchRawQuery	String	If the **searchMode** field is set to 2 to run a **Raw Query Search,** specify the raw query for the search you want to run. In this case, leave the rest of the **mailSearch** inputs empty, and use the **searchRawQuery** field to enter the raw query you want to run. For details on raw query search syntaxes, refer to the *Raw Query Mail Searches* section of the *ZL UA Raw Query Search User's Guide*.

Schema Field	Type	Description
searchType		Specify the type of mails to search for: • **1**: Journaled Mails. Use this when searching the search store for journaled mails with the **searchMails** endpoint. • **2**: Archived Mails. Use this when searching the search store for archived mails with the **searchMails** endpoint. • **9**: In-Place Mails. Use this when searching the search store for In-Place mails with the **searchMailsInplace** endpoint.
recursive	Boolean	Set the **recursive** field to True to conduct the search recursively, meaning that the search results will include messages sent between users that belong to the child departments of the departments specified by the **searchDomain** field. Set this to False to exclude messages from the child departments.
projectType	Integer	Reserved. Leave empty.

Schema Field	Type	Description
Use the following mailSearch fields to filter the search by sender and recipients, so that only messages that have been sent to or from certain users will be included in the search results.		
searchFrom	String	Use this field to specify the email address of the sender.
searchFromRecipientOperator	String	Use this field to indicate how the sender and recipient email addresses specified for the **searchFrom** and **searchRecipients** fields) should be used to filter the search results: • **OR:** Return messages that were either sent by the email sender address or received by the recipient email address. • **AND:** Return messages that contain both the sender and recipient email addresses. • **BETWEEN:** Return messages that were sent between the sender and recipient email addresses. • **PRIVATE:** Return private messages that were sent between the sender and recipient email addresses.
searchRecipients	String	Specify the email address of the recipient(s). You can enter multiple email addresses as a comma-separated list.

Schema Field	Type	Description
Use the following mailSearch fields to filter the search by message content inclusion. Messages that include the keywords or phrases defined by these fields in their contents will be included in the search results.		
searchContentType	Integer	When filtering a search by message content inclusion, specify a value to determine how the message contents will be searched: • **1:** Return messages where the **exact phrase** specified within the **searchContent** field is included in the message contents. • **2:** Return messages where **all the words** specified within the **searchContent** field are included in the message contents. • **3**: Return messages where **any of the words** specified within the **searchContent** field are included in the message contents. • **4**: Run a proximity search. The search will return messages where the words specified within the **searchContent** field occur within a certain proximity of each other (e.g., within 10 words of each other) in the message contents. In this case, use the **searchContentTypProximityWindow** field to determine the proximity requirement. Leave this field empty if you do not want to filter the search results by message content.
searchContentTypeProximityWindow	Integer	You only need to configure this setting when you set the **searchContentType** field to 4 to run a proximity search on the message contents. The search will return files where the words specified within the **searchContent** field appear within this number of words of each other in the message contents. For example, if you use the default value of 10 and the **searchContent** field is set to guarantee rebate, the search would return messages where *guarantee* and *rebate* occur within 10 words of each other.

Schema Field	Type	Description
searchContent	String	Enter a word (or group of words) to search for in the message contents. Messages that include some (or all) of the specified word(s) in their contents will be returned by the search. Use the **searchContentType** field to determine exactly how the message contents will be searched. Leave this field empty if you do not want to filter the search results by message contents.
The following fields are reserved.		
searchNameType	String	Leave empty.
searchNameProximityWindow	Integer	Leave set to the default value of 10.
Use the following fields to specify how message flags will be used to filter the search result.		
searchFlagsMode	Integer	Specify a value to indicate how the search should use message flag attribute values to **include** messages in the search results: • -1: Do not filter messages by flag status. • 2: Return messages where all flags specified by the **searchMessageFlags** field are present. • 3: Return messages where any of the flags specified by the **searchMessageFlags** field are present.

Schema Field	Type	Description
searchExcludeFlagsMode	Integer	Specify a value to indicate how the search should use message flag attribute values to **exclude** messages from the search results: • **-1**: Do not exclude messages based on message flag status. • **2:** Exclude messages where all flags specified by the **searchExcludeMessageFlags** field are present. • **3**: Exclude messages where any of the flags specified by the **searchExcludeMessageFlags** field are present.
Use the following mailSearch fields to filter the search by message subject inclusion. Messages that include the content specified by these fields in their subject will be included in the search results.		
searchSubjectType	String	When filtering the search results by message subject, specify a value to determine how the message subjects will be searched: • **1:** Return messages where the **exact phrase** specified within the **searchSubject** field is included in the message subject. • **2:** Return messages where **all the words** specified within the **searchSubject** field are included in the subject. • **3**: Return messages where **any of the words** specified within the **searchSubject** field are included in the subject. • **4**: Run a proximity search. The search will return messages where the words specified within the **searchSubject** field occur within a certain proximity of each other (e.g., within 10 words of each other). In this case, use the **searchSubjectTypeProximityWindow** field to determine the proximity requirement. Leave this field empty if you do not want to filter the search results by message subject.

Schema Field	Type	Description
searchSubject	Integer	Enter a word (or group of words) to search for in the message subjects. Messages that include some (or all) of the specified words in their subject will be included in the search results. Use the **searchSubectType** field to determine how the message subjects will be searched. Leave this field empty if you do not want to filter the search results by message subject.
searchSubjectTypeProximityWindow	Integer	You only need to configure this setting when you set the **searchSubjectType** field to 4 to run a proximity search on the message subjects. The search will return messages where the words specified within the **searchSubject** field appear within this number of words of each other. For example, if you use the default value of 10 and the **searchSubject** field is set to guarantee rebate, the search would return messages where *guarantee* and *rebate* occur within 10 words of each other in the message subject.

Schema Field	Type	Description
Use the following mailSearch fields to filter the search by message subject exclusion. Messages that include the content specified by these fields in their subject will be excluded from the search results.		
searchSubjectTypeExclude	Boolean	If **searchSubjectExcludeOn** is set to True, use this field to determine how the message subjects will be searched to determine which messages should be excluded from the search results: • **1:** Exclude messages where the **exact phrase** specified within the **searchSubjectExclude** field is included in the message subject. • **2:** Exclude messages where **all the words** specified within the **searchSubjectExclude** field are included in the message subject. • **3**: Exclude messages where **any of the words** specified within the **searchSubjectExclude** field are included in the message subject. • **4**: Run a proximity search. The search results will exclude messages where the words specified within the **searchSubjectExclude** field occur within a certain proximity of each other (e.g., within 10 words of each other). In this case, use the **searchSubjectTypeExcludeProximityWindow** field to determine the proximity requirement.
searchSubjectExclude	String	If **searchSubjectExcludeOn** is set to True, enter a word (or group of words) to filter the search by message subject, so that messages that include some (or all) of the specified word(s) will be **excluded from the search results**. Use the **searchSubjectTypeExclude** field to determine how the message subjects will be searched.

Schema Field	Type	Description
searchSubjectExcludeOn	Integer	Set to True to filter the search so that messages that include a specified word(s) in their subject will be **excluded from the search results**. In this case, use the **searchSubjectExclude** field to specify the words to search for, and the **searchSubjectTypeExclude** field to determine how the message subjects will be searched. Set to False if you do not want to filter the search in this manner.
searchSubjectTypeExcludeProximity Window	Integer	You only need to configure this setting when you set the **searchSubjectTypeExclude** field to 4 to run a proximity exclusion search on the message subjects. The search will exclude messages where the words specified within the **searchSubjectExclude** field appear within this number of words of each other in the message subject. For example, if you use the default value of 10 and the **searchSubjectExclude** field is set to guarantee rebate, the search would exclude messages where *guarantee* and *rebate* occur within 10 words of each other.

Schema Field	Type	Description
Use the following mailSearch fields to filter the search by message body exclusion. Messages that include the keywords or phrases defined by these fields in their body will be excluded from the search results.		
searchBodyTypeExclude	String	When filtering the search results by message body exclusion, specify a value to determine how the message bodies will be searched: • **1:** Messages where the **exact phrase** specified within the **searchBodyExclude** field is included in the message body will be excluded from the search results. • **2:** Messages where **all the words** specified within the **searchBodyExclude** field are included in the body will be excluded from the search results. • **3**: Messages where **any of the words** specified within the **searchBodyExclude** field are included in the body will be excluded from the search results. • **4**: Run a proximity search. Messages where the words specified within the **searchBodyExclude** field occur within a certain proximity of each other (e.g., within 10 words of each other) will be excluded from the search results. In this case, use the **searchBodyTypeExcludeProximityWindow** field to determine the proximity requirement. Leave this field empty if you do not want to filter the search results by message body exclusion.

Schema Field	Type	Description
searchBodyExclude	String	Use this field to filter the search by message body exclusion. Enter a word (or group of words) to search for in the message bodies. Messages that include some (or all) of the specified words in their bodies will be excluded from the search results. Use the **searchBodyTypeExclude** field to determine how the message bodies will be searched. Leave this field empty if you do not want to filter the search results by message body exclusion.
searchBodyExcludeOn	Boolean	Set to True to filter the search by message body exclusion. Set to False if you do not want to filter the search results by message body exclusion.
searchBodyTypeExcludeProximityWindow	String	You only need to configure this setting when you set the **searchBodyTypeExclude** field to 4 to run a proximity search on the message bodies. The search will return messages where the words specified within the **searchBodyExclude** field appear within this number of words of each other within the message body. For example, if you use the default value of 10 and the **searchBodyExclude** field is set to guarantee rebate, the search results would exclude messages where *guarantee* and *rebate* occur within 10 words of each other.

Schema Field	Type	Description
Use the following mailSearch fields to filter the search by message body inclusion. Messages that include the keyworkds or phrases defined by these fields in their body will be <u>included in</u> the search results.		
searchBodyType	String	When filtering the search results by message body, specify a value to determine how the message bodies will be searched: • **1:** Return messages where the **exact phrase** specified within the **searchBody** field is included in the message body. • **2:** Return messages where **all the words** specified within the **searchBody** field are included in the message body. • **3**: Return messages where **any of the words** specified within the **searchBody** field are included in the message body. • **4**: Run a proximity search. The search will return messages where the words specified within the **searchBody** field occur within a certain proximity of each other (e.g., within 10 words of each other) within the message body. In this case, use the **searchBodyTypProximityWindow** field to determine the proximity requirement. Leave this field empty if you do not want to filter the search results by message body.
searchBody	String	Use this field to filter the search by message body inclusion. Enter a word (or group of words) to search for in the message bodies. Messages that include some (or all) of the specified words in their bodies will be included in the search results. Use the **searchBodyType** field to determine how the message bodies will be searched. Leave this field empty if you do not want to filter the search results by message body.

Schema Field	Type	Description
searchBodyTypeProximityWindow	String	You only need to configure this setting when you set the **searchBodyType** field to 4 to run a proximity search on the message bodies. The search will return messages where the words specified within the **searchBody** field appear within this number of words of each other within the message body. For example, if you use the default value of 10 and the **searchBody** field is set to guarantee rebate, the search would return messages where *guarantee* and *rebate* occur within 10 words of each other.

Schema Field	Type	Description
Use the following mailSearch fields to filter the search by message attachment file name. Messages that include the keywords specified by these fields in their attachment name will be included in the search results.		
searchAttachmentNamesType	String	When filtering the search results by message attachment file name, specify a value to determine how the attachment file names will be searched: • **1:** Return messages where the **exact phrase** specified within the **searchAttachmentNames** field is included in the attachment file name. • **2:** Return messages where **all the words** specified within the **searchAttachmentNames** field are included in the attachment file name. • **3**: Return messages where **any of the words** specified within the **searchAttachmentNames** field are included in the attachment file name. • **4**: Run a proximity search. The search will return messages where the words specified within the **searchAttachmentNames** field occur within a certain proximity of each other (e.g., within 10 words of each other). In this case, use the **searchAttachmentNameProximityWindow** field to determine the proximity requirement. Leave this field empty if you do not want to filter the search results by file attachment name.
searchAttachmentNames	String	Enter a word (or group of words) to search for in the message attachment file names. Messages that include some (or all) of the specified words in the file names of their attachments will be returned by the search. Use the **searchAttachmentNamesType** field to determine how the attachment file names will be searched. Leave this field empty if you do not want to filter the search results by attachment file name.

Schema Field	Type	Description
searchAttachmentNamesTypeProximityWindow	String	You only need to configure this setting when you set the **searchAttachmentNamesType** field to 4 to run a proximity search on the message attachment file name. The search will return messages where the words specified within the **searchAttachmentNames** field appear within this number of words of each other in the attachment file name. For example, if you use the default value of 10 and the **searchAttachmentNames** field is set to guarantee rebate, the search would return messages where *guarantee* and *rebate* occur within 10 words of each other.
Use the following mailSearch fields to filter the search by message attachment body. Messages that include the keywords defined by these fields in the bodies of their attachments will be included in the search results.		

Schema Field	Type	Description
searchAttachmentBodyType	String	When filtering the search results by message attachment body content, specify a value to determine how the bodies of the message attachments will be searched: • **1:** Return messages where the **exact phrase** specified within the **searchAttachmentBody** field is included in the attachment body. • **2:** Return messages where **all the words** specified within the **searchAttachmenBody** field are included in the attachment body. • **3**: Return messages where **any of the words** specified within the **searchAttachmentBody** field are included in the attachment body. • **4**: Run a proximity search. The search will return messages where the words specified within the **searchAttachmentBody** field occur within a certain proximity of each other (e.g., within 10 words of each other). In this case, use the **searchAttachmentBodyProximityWindow** field to determine the proximity requirement. Leave this field empty if you do not want to filter the search results by message file attachment body.
searchAttachmentBody	String	Use this field to filter the search by message attachment body. Enter a word (or group of words) to search for in the bodies of messages attachments. Messages that include some (or all) of the specified words in the bodies of their attachments will be returned by the search. Use the **searchAttachmentBodyType** field to determine how the bodies of the attachments will be searched. Leave this field empty if you do not want to filter the search results by message attachment body.

Schema Field	Type	Description
searchAttachmentBodyTypeProximityWindow	String	You only need to configure this setting when you set the **searchAttachmentBodyType** field to 4 to run a proximity search on the message attachment bodies. The search will return messages where the words specified within the **searchAttachmentBody** field appear within this number of words of each other in the message attachment body. For example, if you use the default value of 10 and the **searchAttachmentBody** field is set to guarantee rebate, the search would return messages where *guarantee* and *rebate* occur within 10 words of each other.
Use the following mailSearch fields to filter the search based on various message attributes such as the message ID, the message direction or the date the message was sent.		
searchFileExtension	Boolean	Reserved. Leave empty.
searchOtherHeader	String	Message headers include some metadata that is indexed and searchable. Use this field to specify a search term. Messages that include this term in the header metadata fields will be returned by the search.
searchMessageId	String	Every message in ZL UA is assigned a unique ZLP message ID. Use this field to search for a message by this ID value. Specify a single ID value.
searchDirectionMode	String	If the **searchType** field is set to 1 for journaled mails and you want to search for messages sent in a specific direction (i.e., inbound or outbound), set this field to **specific** and use the **searchDirection** field to specify the message direction to search for. You can leave this empty if you want to search for messages sent in all directions. If the **searchType** field is set to 2 or 9 for archived or In-Place mails, this setting is not applicable. It will be sent as an empty value.

Schema Field	Type	Description
searchDirection	String	If the **searchType** field is set to 1 for journaled mails and the **searchDirectionMode** field is set to **specific**, specify the message direction to search for. You can specify multiple directions if desired. In this case, enter them as a comma-separated list: • **I**: Return inbound messages. • **O:** Return outbound messages. • **N:** Return Internal messages. If the **searchType** field is set to 2 or 9 for archived or In-Place mails, this field is not applicable. It will be sent as an empty value.
searchTypeMode	String	If the **searchType** field is set to 1 for journaled mails and you want to search for specific message types, set this field to **specific** and use the **searchMessageType** field to specify the message types to search for. You can leave this empty if you do not want to filter the search by message direction. If the **searchType** field is set to 2 or 9 for archived or In-Place mails, this field is not applicable. It will be sent as an empty value.

Schema Field	Type	Description
searchMessageType	String	If the **searchType** field is set to 1 for journaled mails and the **searchTypeMode** field is set to **specific**, specify the message types to search for: • **Mail** • **Im** • **Fax** • **Bloomberg** • **Logs** • **Socialmedia** • **Reuters** • **Voice** You can specify multiple values as a comma-separated list. If the **searchType** field is set to 2 or 9 for archived or In-Place mails, this field is not applicable. It will be sent as an empty value.
searchExcludeMessageFlags	String	If **searchExcludeFlagsMode** is set to 2 or 3, enter the flags to search for as a comma-separated list: • **MF_ATTACH**: Has attachments • **MF_ATTACH_NOT_SCANNED**: Attachments not scanned • **MF_ATTACHMENT_NOT_SCANNED_POLICY**: Attachments not scanned by policy • **MF_CALENDAR**: Calendar item • **MF_PRIVATE_COMM**: Private Communication • **MF_TRUNCATED_TEXT_CONTENT**: Text Not Fully Indexed

Schema Field	Type	Description
searchMessageFlags	String	If **searchFlagsMode** is set to 2 or 3, enter the flags to search for as a comma-separated list: • **MF_ATTACH**: Has attachments • **MF_ATTACH_NOT_SCANNED**: Attachments not scanned • **MF_ATTACHMENT_NOT_SCANNED_POLICY**: Attachments not scanned by policy • **MF_CALENDAR**: Calendar item • **MF_PRIVATE_COMM**: Private Communication • **MF_TRUNCATED_TEXT_CONTENT**: Text Not Fully Indexed
searchMessageSubtype	Boolean	Reserved. Leave empty.
searchDateMode **searchDateStart** **searchDateEnd** **searchUseGMT**	String String String Boolean	To filter the search to return messages that were sent during a specific date range, follow these steps: 1. Set **searchDateMode** to **between** to return messages that were sent during a specific date range. 2. Use the **searchDateStart** and **searchDateEnd** fields to specify the date range. The date range is inclusive. Use the following format to specify the dates: YYYY/MM/DD 3. Set the **searchUseGMT** field to True to use GMT time for the date range. Set to False to use the local server time. Leave these fields empty if you don't want to filter by the dates that messages were sent.
SearchMessaeSizeUnit **SearchMessageSizeUnit2** **SearchMessageSize** **SearchMessageSize2**	String String String String	Reserved. Leave these values empty.

Schema Field	Type	Description
recipientsExclude **fromExclude**	String Boolean	To exclude messages sent by a particular email address (or group of addresses) from the search results, set the f**romExclude** field to True and use the **recipientsExclude** field to specify the email addresses you want to exclude here. You can enter multiple email addresses as a comma-separated list. Otherwise, set **fromExclude** to False.
Use the following mailSearch fields to filter the search based on message size:		
searchMessageSizeMode	Integer	To filter the search by message size, follow these steps: 1. Specify the mode to use to filter messages based on size: • -1 for **any size** • 3 for **less than** • 4 for **greater than** • 5 for **between** 2. Use the **searchMessageSizeLow** and **searchMessageSizeLowUnit** fields described below to specify the lower limit of the size range. 3. Use the **searchMessageSizeHigh** and **searchMessageSizeHighUnit** fields described below to specify the upper limit of the size range. Steps 2 and 3 may not be necessary depending on the **searchMessageSizeMode** selected. For example, if you choose **4** for **greater than**, you would only need to specify the lower end of the size range.
searchMessageSizeLow	String	Lower limit of message size for filtering.
searchMessageSizeLowUnit	Integer	Unit for the lower size limit: • **2**: Kilobytes • **3**: Megabytes

Schema Field	Type	Description
searchMessageSizeHigh	String	Upper limit of message size for filtering.
searchMessageSizeHighUnit	Integer	Unit for the upper size limit (e.g., KB, MB).
Use the following mailSearch fields to filter the search based on other message attributes:		
addressResolution	Boolean	Some ZL UA users have multiple email addresses and aliases that they use for different types of communications. Set this to True to include all aliases in the searches for known ZL UA users.
searchFromExclude **searchRecipientsExclude**	Boolean String	To exclude messages sent from a particular email address (or group of addresses) from the search results, set the **searchFromExclude** field to True and use the **searchReceipientsExclude** field to specify the email addresses you want to exclude. You can enter multiple email addresses as a comma-separated list. Otherwise, set the **searchFromExclude** field to False.
The following fields are reserved.		
includeIdFrom	String	Leave empty.
includeIdRecipients	String	Leave empty.
callerAppId	Integer	Set to 202.
contextId	Integer	Set to -1.

Response Schema Fields

Schema Field	Type	Description
domainId	Integer	The domain ID associated with the search.
items	Array	The **items** array includes an entry for each message returned by the search. Each entry includes the following fields:
deletedSearchItem	Boolean	This field is not applicable to messages.

Schema Field	Type	Description
refItemId	String	The reference ID assigned to the message.
relevancy	String	The relevancy rating assigned to the message. The relevancy rating is an indication of how well the message matched the search criteria used to conduct the search. A low relevancy rating indicates the message matched a greater part of the search criteria (e.g., messages that are assigned a relevancy rating of 100.0 matched the most search criteria). **Please note that relevancy scores are a Beta Level feature as of the release of ZL UA 11.1.2.**
rowId	String	Internal use only.
size	Integer	The size of the message, in KB or MB.
type	Integer	The message type.
viewItemId	String	The view ID assigned to the message.
createDate	String	The date the message was sent.
itemId	String	The item ID assigned to the message.
mapFlags	Array	Includes a series of flags showing the status of various message attributes. For example, the **MF_ATTACH** flag indicates whether the message includes attachments (True) or not (False), and the **MF_CALENDAR f**lag indicates whether the message is a calendar item (True) or not (False).
recipient	Array	The **recipient** array includes an entry for each recipient of the message. Each entry includes strings showing the name (**name)** and e-mail address (**addr**) of the message recipients. The **fHasMore** Boolean will be set to False for the last recipient in the list to indicate that it is the last entry in the array.
retention	Array	Includes retention-related information. The **dateEnd** parameter is a string showing the end of the message's retention period.
sender	Array	The **sender** array includes an entry for the user who sent the message. The entry includes strings showing the name (**name)** and e-mail address (**addr**) of the message recipients. The **fHasMore** Boolean will be set to False to indicate the last entry in the array.
seqId	Integer	The sequence ID assigned to the message.

Schema Field	Type	Description
subject	String	The message subject.
itemsPerPage	Integer	The number of messages per page.
rawQuery	String	A representation of the search parameters specified.
searchStoreId	Integer	The ID assigned to the search store used to run the search.
totalRecords	Integer	The number of messages returned by the search.
viewId	Integer	The view ID assigned to the search.

FAM/Project Privileges

In the FAM module, privileges are granted to users and security groups to allow them to perform operations within a project. Privileges are granted on a project-by-project basis, so a user or security group can have different privilege levels within different projects. An example of a privilege that can be granted to users or security groups is the “Search” privilege.

Use these endpoints to manage project privileges within the EFM module:

- *GET: Get Project Privileges Using Project ID (getprojectprivileges):* Get the project privileges that have been assigned within a project.
- *PUT: Grant Group Project Privileges (grantgroupprojectprivileges)*: Assign project privileges to a security group.
- *PUT: Grant User Project Privileges (grantuserprojectprvileges):* Assign project privileges to a user.
- *PUT: Revoke All Project Privileges (revokeallprojectprivileges):* Revoke all privileges that have been assigned within a project.
- *PUT: Revoke Group Project Privileges (revokegroupprojectprivileges):* Revoke project privileges from a security group.
- *PUT: Revoke User Project Privileges (revokeuserprojectprivileges):* Revoke project privileges from a user.

GET: Get Project Privileges Using Project ID (getprojectprivileges)

Get the project privileges that have been assigned within a project.

Path

http://localhost:8080/ps/api/v1/privileges/getprojectprivileges/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project from which you want to retrieve the information. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
User/Security Group Project Privileges: Includes an entry that includes the following fields for each user or security group that has been assigned privileges within the project.		
username	String	The name of the user or security group.
privileges	String	The privileges assigned to the user or security group.

PUT: Grant Group Project Privileges (grantgroupprojectprivileges)

Assign project privileges to a security group(s).

Path

http://localhost:8080/ps/api/v1/privileges/grantgroupprojectprivileges

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
zlpUserIds	Integer	Specify a list of ZLP user IDs to specify the security groups to grant the privileges to. Specify the group IDs as a comma-separated list. For example: `"zlpUserIds": [ 0,1,2,3 ]` The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
The remaining fields are Boolean values. Set to True to grant the privilege to the security groups specified in the request.		
fProjectAdminPrivilege	Boolean	Has access to all FAM functionality for the project: project configurations, administrative tasks, tag management, searches, reports, audit trails and granting project roles to other users.
fReadPrivilege	Boolean	Can view the contents of the project and run reports.
fAnalyticsReview	Boolean	Can view the contents of the project, tag items, and run searches and reports

Schema Field	Type	Description
fAnalyticsSearch	Boolean	Can view the contents of the project and run searches and reports.
fAnalyticsAudits	Boolean	Can view audit trails, view the contents of the project, and run reports.
fRetentionApproval	Boolean	Can approve the destruction of files that are eligible for disposition because their records management lifecycle has expired.

Response Schema Fields

Schema Field	Type	Description
entityId	Integer	The ID assigned to the project entity created by the request. Each project entity represents a user or security group that has been assigned privileges within the project, and you can use the ID to retrieve information regarding this with other endpoints.
entityType	Integer	The entity type.
projectPrivileges	String	An array of strings identifying the privileges assigned to the security group.

PUT: Grant User Project Privileges (grantuserprojectprvileges)

Assign project privileges to a user (or group of users).

Path

http://localhost:8080/ps/api/v1/privileges/grantuserprojectprivileges

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
zlpUserIds	Integer	An array of user IDs to specify the users to whom the project privileges should be granted. Specify the user IDs as a comma-separated list. For example: `"zlpUserIds": [ 0,1,2,3 ]` The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
The remaining fields are Boolean values. Set to True to grant the privilege to the users specified in the request.		
fProjectAdminPrivilege	Boolean	Has access to all FAM functionality for the project: project configurations, administrative tasks, tag management, searches, reports, audit trails and granting project roles to other users.
fReadPrivilege	Boolean	Can view the contents of the project and run reports.
fAnalyticsReview	Boolean	Can view the contents of the project, tag items, and run searches and reports
fAnalyticsSearch	Boolean	Can view the contents of the project and run searches and reports.

Schema Field	Type	Description
fAnalyticsAudits	Boolean	Can view audit trails, view the contents of the project, and run reports.
fRetentionApproval	Boolean	Can approve the destruction of files that are eligible for disposition because their records management lifecycle has expired.

Response Schema Fields

Schema Field	Type	Description
entityId	Integer	The ID assigned to the privileges entity created by the request.
entityType	Integer	The entity type.
projectPrivileges	String	An array of strings identifying the privileges assigned to the user.

PUT: Revoke All Project Privileges (revokeallprojectprivileges)

Revoke all privileges that have been assigned within a project.

Path

http://localhost:8080/ps/api/v1/privileges/revokeallprojectprivileges/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project from which you want to revoke privileges. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

A string message indicating whether the privileges were successfully revoked or not.

PUT: Revoke Group Project Privileges (revokegroupprojectprivileges)

Revoke privileges that have been assigned to a specific list of security groups within a project.

Path

http://localhost:8080//v1/privileges/revokegroupprojectprivileges

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
zlpUserIds	Integer	An array of group IDs to specify the security groups from which the project privileges should be revoked. Specify the group IDs as a comma-separated list. For example: `"zlpUserIds": [ 0,1,2,3 ]` The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.

Response Schema Fields

Schema Field	Type	Description
Additional Prop: Includes the following fields for each security group specified in the request.		
success	Boolean	Indicates whether privileges were revoked from the security group successfully (True) or not (False).
result	String	The result of the request. If **Success** is set to True, a message will display indicating that privileges have been revoked.
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

PUT: Revoke User Project Privileges (revokeuserprojectprivileges)

Revoke privileges that have been assigned to a specific list of users within a project.

Path

http://localhost:8080/ps/api/v1/privileges/revokeuserprojectprivileges

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
zlpUserIds	Integer	An array of user IDs to specify the users from whom the project privileges should be revoked. Specify the user IDs as a comma-separated list. For example: `"zlpUserIds": [ 0,1,2,3 ]` The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.

Response Schema Fields

Schema Field	Type	Description
Additional Prop: Includes the following fields for each user specified in the request.		
success	Boolean	Indicates whether privileges were revoked from the group successfully (True) or not (False).
result	String	The result of the request. If successful, a message will be displayed that privileges have been revoked.
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

UAA/Projects and FAM/Projects

A project is a list of folders or sites that is grouped together to be scanned whenever a server is crawled. Projects are created to determine which system directories or sites (and, subsequently, which items) in the selected server are to be ingested into ZL UA and managed together.

The following sections describe the UAA/Projects and FAM/Projects endpoints available in the REST API. Use these endpoints to create and manage Google Drive, OneDrive, SharePoint, Box and File Share projects for use in the UAA and FAM modules:

- *POST: Crawl Project (runcrawl)*
- *POST: Create File Share Project (createfileshareproject)*
- *POST: Create SharePoint Project (createsharepointproject)*
- *DELETE: Remove Project (deleteusingid)*
- *GET: Get Sub-Folders (subfolders)*
- *GET: Get Project Info by ID (getprojectusingid)*
- *GET: Get Project Info by Name (getprojectusingname)*
- *GET: Get Projects Using Search (getprojectusingpatternsearch)*
- *PUT: Update Box Project (updateBoxProject)*
- *PUT: Update File Share Project (updatefileshareproject)*
- *PUT: Update OneDrive Project (updateonedriveproject)*
- *PUT: Update SharePoint Project (updatesharepointproject)*

POST: Crawl Project (runcrawl)

Run a full scan upon a project. This scans the server for any changes that have been made to previously ingested files and makes the necessary changes in ZL UA. It also identifies, captures, and acts on any new files on the server or any old files that are ready to be archived from the server.

Path

UAA/Projects: http://localhost:8080/ps/api/v1/uaaprojects/runcrawl/{projectId}

FAM/Projects: This endpoint is not available for use with FAM projects. You can use the *POST: Run Crawl (runcrawl)* endpoint described on page 266 (available under **FAM/Tasks**) to run a full scan upon a FAM project.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be crawled. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the crawl has been started successfully. The string also includes the task ID assigned to the project crawl.

POST: Create File Share Project (createfileshareproject)

Create a project on a File Share server.

Path

http://localhost:8080/ps/api/v1/storage/project/createfileshareproject

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the project ID. This is only required when updating a File Share project with the PUT endpoint. It is not required when creating a File Share project. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
ownerId	Integer	Specify the ZLP user ID assigned to the user who should be the project owner. For policy resolution and ease of management purposes, the project will be associated with the department of the ZL UA user specified as the project owner.
type	Integer	Specify the project type. Enter 1 for file share projects.
name	String	Specify the project's name.
displayName	String	Specify the project's display name.
mailServerId	Integer	Type the server ID assigned to the project's file server. You can retrieve server IDs with the GET: Get All Severs endpoint described on page 230.
domainId	Integer	Specify the domain ID.

Schema Field	Type	Description
crawlType	Integer	Specify the crawl type: • **400**: Metadata Analysis (No Index) • **200**: Metadata Index + Content Analysis • **100**: Metadata Index + Content Analysis + Content Index
scheduleStartDate **scheduleEndDate**	String	Specify the start and end dates during which disposition should be carried out upon the project. Use the following formats for dates: YYYY-MM-DDTMM:HH:SS.SSZ For example: **2024-11-01T11:05:44.575Z** This setting is applicable to projects where the Disposition Workflow is enabled, meaning that the **disposition_enabled** setting is set to True. In addition, scheduling for the Disposition Workflow must be enabled, meaning that the **fScheduleEnabled** setting is set to True.
fScheduleEnabled	Boolean	Specify whether disposition scheduling should be enabled for the project. This setting is applicable to projects where the disposition setting (**disposition_enabled**) is enabled.
allowVersioning	Boolean	Specify whether file versioning should be used in this project. Input to this field is not required for FAM projects.

Schema Field	Type	Description
flags	Boolean	A set of Boolean values used to specify whether each project attribute should be enabled or not. Set to True to enable, or False to disable. • **allowsAddition**: Allow new files to be added to the project during file crawls. • **disable_acl_capture**: Disable the capture of ACL permissions for all folders and files in the project. In this case, the ZL File Connector will not check the ACL permissions while crawling. • **add_users_based_folder_acl**: Any ZL UA user with at least 'Read' ACL privileges for a folder within the project will automatically have access to the folder in the project in the ZL File Archiving application. • **fetch_file_acl**: Retrieve the ACL list for each file included in the project. • **crawlfetchlatestver**: Retrieve the latest versions of previously added files during project crawls. • **lock_crawl**: Disable future crawls of the project. • **disable_full_crawl**: Indicate whether the entire project should be scanned (and subject to archiving) when it is crawled. • **disposition_enabled**: Enable disposition on the project. Disposition is the process by which files whose records management lifecycle has expired are deleted and removed from the ZL UA system. • **disposition_approval_required**: Require approval for Disposition Runs within the project.

Schema Field	Type	Description
folderRoot	These fields are output fields used to define the project's root folder. Input is not required here when using the endpoint.	
dirId	Integer	Input to these fields is not required.
parentId	Integer	
name	String	
displayName	String	
description	String	
relativePath	String	
folderType	Integer	
storageSize	Integer	
itemCount	Integer	
deleted	Boolean	This field is used for output purposes. Not required.
dirSpec	Specify which folders and sub-folders will be included in the project. Each entry in the `allEntry` array identifies a project folder and includes the following fields:	
symbolicLink	String	The path display name.

rootPath	String	Specify the root path of the folder. For example:
		`C:\\Users\\bross\\Downloads\\fileset`
		You could also specify the path as:
		`C:/Users/bross/Downloads/fileset`

exclusionPathSet	String	Specify the relative paths of any sub-folders that should not be included when crawling the project. For example:
		`C:\\Users\\adunna\\Downloads\\fam`
		You could also specify the path as:
		`C:/Users` This field is used for output purposes. No input is required.`/adunna/Downloads/fam`

Schema Field	Type	Description
treeCrawl	Boolean	These fields are used for output purposes. No input is required.
propertyMap	String	
privilegesSettings	Boolean	A set of Boolean values used to specify the default permissions users should be granted for the project: • **fprojectAdminPrivilege:** Perform administrative actions on the project. • **fReadPrivilege**: Read the contents of the project. • **fSearchPrivilege**: Search the contents of the project. • **fSharePrivilege**: Not supported. • **fWebDavPrivilege**: Not supported. • **fAuditPrivilege**: View audit trial data for the project.

Response Schema Fields

The fields included in the response schema are the same set of fields required in the request schema. These fields define the project configuration.

POST: Create SharePoint Project (createsharepointproject)

Create a project on a SharePoint site.

Path

http://localhost:8080/ps/api/v1/storage/project/createsharepointproject

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the project ID. This is only required when updating a SharePoint project with the PUT endpoint. It is not required when creating a SharePoint project. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
ownerId	Integer	Specify the ZLP user ID assigned to the user who should be the project owner. For policy resolution and ease of management purposes, the project will be associated with the department of the ZL UA user specified as the project owner.
type	Integer	Specify the project type. Enter 3 for SharePoint.
name	String	Specify the project's name.
displayName	String	Specify the project's display name.
mailServerId	Integer	Type the server ID assigned to the project's server. You can retrieve server IDs with the GET: Get All Severs endpoint described on page 230.
domainId	Integer	Specify the domain ID.

Schema Field	Type	Description
crawlType	Integer	Specify the crawl type: • **400**: Metadata Analysis (No Index), • **200**: Metadata Index + Content Analysis • **100**: Metadata Index + Content Analysis + Content Index
scheduleStartDate **scheduleEndDate**	String	Specify the start and end dates during which disposition should be carried out upon the project. Use the following formats for dates: YYYY-MM-DDTMM:HH:SS.SSZ For example: **2024-11-01T11:05:44.575Z** This setting is applicable to projects where the Disposition Workflow is enabled, meaning that the **disposition_enabled** setting is set to True. In addition, scheduling for the Disposition Workflow must be enabled, meaning that the **fScheduleEnabled** setting is set to True.
fScheduleEnabled	Boolean	Specify whether disposition scheduling should be enabled for the project. This setting is applicable to projects where the disposition setting (**disposition_enabled**) is enabled.
allowVersioning	Boolean	Specify whether file versioning should be used in this project. Input to this field is not required for FAM projects.

Schema Field	Type	Description
flags	Boolean	A set of Boolean values used to specify whether each project attribute should be enabled or not. Set to True to enable, or False to disable. • **allowsAddition**: Allow new files to be added to the project during file crawls. • **disable_acl_capture**: Disable the capture of ACL permissions for all folders and files in the project. In this case, the ZL File Connector will not check the ACL permissions while crawling. • **add_users_based_folder_acl**: Any ZL UA user with at least 'Read' ACL privileges for a folder within the project will automatically have access to the folder in the project in the ZL File Archiving application. • **fetch_file_acl**: Retrieve the ACL list for each file included in the project. • **crawlfetchlatestver**: Retrieve the latest versions of previously added files during project crawls. • **lock_crawl**: Disable future crawls of the project. • **disable_full_crawl**: Indicate whether the entire project should be scanned (and subject to archiving) when it is crawled. • **disposition_enabled**: Enable disposition on the project. Disposition is the process by which files whose records management lifecycle has expired are deleted and removed from the ZL UA system. • **disposition_approval_required**: Require approval for Disposition Runs within the project.

Schema Field	Type	Description
folderRoot	These fields are output fields used to define the project's root folder. Input is not required here when using the endpoint.	
dirId	Integer	Input to these fields is not required.
parentId	Integer	
name	String	
displayName	String	
description	String	
relativePath	String	
type	Integer	
storageSize	Integer	
itemCount	Integer	
deleted	Boolean	This field is used for output purposes. Input is not required.
dirSpec	Specify which folders and sub-folders will be included in the project. Each entry in the `allEntry` array identifies a project folder and includes the following fields:	
symbolicLink	String	The path display name.

rootPath	String	Specify the root path of the folder. For example:
		`C:\\Users\\bross\\Downloads\\fileset`
		You could also specify the path as:
		`C:/Users/bross/Downloads/fileset`

exclusionPathSet	String	Specify the relative paths of any sub-folders that should not be included when crawling the project. For example:
		`C:\\Users\\adunna\\Downloads\\fam`
		You could also specify the path as:
		`C:/Users/adunna/Downloads/fam`

Schema Field	Type	Description
treeCrawl	Boolean	These fields are for output purposes. No input is required.
propertyMap	String	
privilegesSettings	Boolean	A set of Boolean values used to specify the default permissions users should be granted for the project: • **fprojectAdminPrivilege:** Perform administrative actions on the project. • **fReadPrivilege**: Read the contents of the project. • **fSearchPrivilege**: Search the contents of the project. • **fSharePrivilege, fWebDavPrivilege**: Not supported. Leave at default values. • **fAuditPrivilege**: View audit trial data for the project.
siteTitle	String	Enter the site title.
siteURL	String	Enter the SharePoint site URL.
guid	String	Enter the SharePoint site GUID. Refer to the *SharePoint Archiving Administrator's Guide* for instructions to follow when retrieving SharePoint site GUID values.
username	String	When using the SharePoint Connector for on-premise SharePoint installations, type the user ID of a site collection administrator who has access to the site URL. When using the SharePoint Online CSOM Connector, ensure that you append "o365:" to the user ID/name. For example, if the username is **john@demo.onmicrosoft.com**, then you would specify the value in this field as **o365:john@demo.onmicrosoft.com.**
password	String	When using either SharePoint Online Connector, type the password of the user whose ID was specified for the **username** field.
spServerAPI	Boolean	Set to True to use the server API rather than the web services API. When this option is True, the **username** and **password** fields are not applicable. When using the Cloud SharePoint Connector to archive the project files, set this to False.

Schema Field	Type	Description
skipHidden	Boolean	Set to True to skip archiving lists that are hidden in SharePoint.
crawlAllVersions	Boolean	Set to True to archive all the versions of a file. If the check box is not selected, only the latest version of the file is crawled.
inclusive **exclusive**	String	Use these fields to specify any SharePoint sites/directories that should (inclusive) or should not (exclusive) be crawled when the project is scanned. Specify them as a comma-separated list. Within the inclusion and exclusion filters there are two different settings: • **T**: Indicates the List Type. For example, Document Library, Announcements, Events, Calendar etc. Syntax: **T:DocumentLibrary** • **L**: Indicates the various names of the type of the Lists. For example, Shared Documents and Site Assets are lists under Document Library. Syntax: **L:Shared Documents** **L:Lists/WORMProject** For some lists, the name is prefixed with 'List/'. You must check the URL before including it in the necessary fields. **Exclusion/Inclusion Parameter Syntax** • All document libraries: **T:DocumentLibrary** • Specific Document Libraries: **Home/regulatory/L:Registration Reports** **Home/L:WORMProject** **Home/regulatory/L:DomFilings; where regulatory is a sub-site of Home.** • Separate libraries with a semi-colon and no spaces: **Home/L:WORMProject;Home/regulatory/L:DomFilings** Please note that Exclusion takes a precedence over inclusions. For example, if you have specified inclusion parameters as *Home/regulatory/L:Test* and exclusion parameter as: *Home/regulatory*, "regulatory" will not be crawled.

Response Schema Fields

The fields included in the response schema are the same set of fields required in the request schema. These fields define the project configuration.

DELETE: Remove Project (deleteusingid)

Delete a project.

Path

http://localhost:8080/ps/api/v1/storage/project/deleteusingid/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be deleted. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the project was deleted successfully.

GET: Get Sub-Folders (subfolders)

Retrieve a list of sub-folders for a parent folder.

Path

http://localhost:8080/ps/api/v1/storage/project/{projectId}/subFolders

Request Parameters

Parameter	Type	Description
projId	Integer	Specify the ID of the project to be viewed. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
parent	Integer	Specify the parent folder ID. Specify -1 for the root folder. The response schema returned by this call includes information for each sub-folder beneath the parent folder.
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None

Response Schema Fields

Schema Field	Type	Description
Folders: Includes an entry for each sub-folder that is found. The following fields are included for each entry:		
dirId	Integer	The folder ID.
parentID	Integer	The ID of the parent folder.
name	String	The folder name.
displayName	String	The folder display name.

Schema Field	Type	Description
description	String	Description.
relativePath	String	The relative path of the folder.
type	Integer	The project type: • 1: File Share • 3: SharePoint • 7: Google Drive • 9: Box • 10: OneDrive
storageSize	Integer	The folder storage size, in bytes.
itemCount	Integer	The number of items contained within the folder.

GET: Get Project Info by ID (getprojectusingid)

Retrieve a project's configuration. Specify the project by its ID.

Path

http://localhost:8080/ps/api/v1/storage/project/getprojectusingid/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	The ID of the project to be viewed. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

The schema fields returned for the project are the same as those required in the request schema when creating the project. They represent the configuration of the project. For descriptions of these fields, refer to the following sections:

- **OneDrive Projects**: *PUT: Update OneDrive Project (updateonedriveproject)* on page 363
- **File Share Projects**: *POST: Create File Share Project (create)* on page 340
- **SharePoint Projects:** *POST: Create SharePoint Project* on page 345
- **Box Projects:** *PUT: Update Box Project (updateBoxProject)* on page 358

GET: Get Project Info by Name (getprojectusingname)

Retrieve project's configuration. Specify the project by its name.

Path

http://localhost:8080/ps/api/v1/storage/project/getprojectusingname/{projectName}

Request Parameters

Parameter	Type	Description
projectName	String	Specify the name of the project to be viewed. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.

Request Body Schema Fields

None.

Response Schema Fields

The schema fields returned for the project are the same as those required in the request schema when creating the project. They represent the configuration of the project. For descriptions of these fields, refer to the following sections:

- **OneDrive Projects**: *PUT: Update OneDrive Project (updateonedriveproject)* on page 363
- **File Share Projects**: *POST: Create File Share Project (create)* on page 340
- **SharePoint Projects:** *POST: Create SharePoint Project* on page 345
- **Box Projects:** *PUT: Update Box Project (updateBoxProject)* on page 358

GET: Get Projects Using Search (getprojectusingpatternsearch)

Retrieve a project by searching for a specific project name.

Path

http://localhost:8080/ps/api/v1/storage/project/getprojectusingpatternsearch/{pattern}

Request Parameters

Parameter	Type	Description
pattern	Integer	Enter the search pattern. The search will return projects whose name includes (or is similar to) the search pattern.
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

The schema fields returned for the project are the same as those required in the request schema when creating the project. They represent the configuration of the project. For descriptions of these fields, refer to the following sections:

- **OneDrive Projects**: *PUT: Update OneDrive Project (updateonedriveproject)* on page 363
- **File Share Projects**: *POST: Create File Share Project (create)* on page 340
- **SharePoint Projects:** *POST: Create SharePoint Project* on page 345
- **Box Projects:** *PUT: Update Box Project (updateBoxProject)* on page 358

PUT: Update Box Project (updateBoxProject)

Update a Box project's configuration.

Path

http://localhost:8080/ps/api/v1/storage/project/updateboxproject

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the project to be updated. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
ownerId	Integer	Specify the ZLP user ID assigned to the project owner. For policy resolution and ease of management purposes, the project will be associated with the department of the ZL UA user specified as the project owner.
type	Integer	Specify the project type. Enter 9 for Box projects.
name	String	Specify the project's name.
displayName	String	Specify the project's display name.
mailServerId	Integer	Specify the server ID assigned to the project's server. You can acquire server IDs with the You can retrieve server IDs with the GET: Get All Mail Severs endpoint. For more information, refer to the *ZL UA Rest API Kit Reference Guide.*
domainId	Integer	Not required.

Schema Field	Type	Description
crawlType	Integer	Specify the crawl type: • **400**: Metadata Analysis (No Index), • **200**: Metadata Index + Content Analysis, • **100**: Metadata Index + Content Analysis + Content Index
scheduleStartDate **scheduleEndDate**	String	Specify the start and end dates during which disposition should be carried out upon the project. Use the following formats for dates: YYYY-MM-DDTMM:HH:SS.SSZ For example: **2024-11-01T11:05:44.575Z** This setting is applicable to projects where the Disposition Workflow is enabled, meaning that the **disposition_enabled** setting is set to True. In addition, scheduling for the Disposition Workflow must be enabled, meaning that the **fScheduleEnabled** setting is set to True.
fScheduleEnabled	Boolean	Specify whether disposition scheduling should be enabled for the project. This setting is applicable to projects where the disposition setting (**disposition_enabled**) is enabled.
allowVersioning	Boolean	Specify whether file versioning should be used in this project. Input to this field is not required for FAM projects.

Schema Field	Type	Description
flags	Boolean	A set of Boolean values used to specify whether each project attribute should be enabled or not. Set to True to enable, or False to disable. • **allowsAddition**: Allow new files to be added to the project during file crawls. • **disable_acl_capture**: Disable the capture of ACL permissions for all folders and files in the project. In this case, the ZL File Connector will not check the ACL permissions while crawling. • **add_users_based_folder_acl**: Any ZL UA user with at least 'Read' ACL privileges for a folder within the project will automatically have access to the folder in the project in the ZL File Archiving application. • **fetch_file_acl**: Retrieve the ACL list for each file included in the project. • **crawlfetchlatestver**: Retrieve the latest versions of previously added files during project crawls. • **lock_crawl**: Disable future crawls of the project. • **disable_full_crawl**: Indicate whether the entire project should be scanned (and subject to archiving) when it is crawled. • **disposition_enabled**: Enable disposition on the project. Disposition is the process by which files whose records management lifecycle has expired are deleted and removed from the ZL UA system. • **disposition_approval_required**: Require approval for Disposition Runs within the project.
folderRoot	Array	These fields are output fields used to define the project's root folder. Input is not required here when using the endpoint.
deleted	Boolean	This field is used for output purposes. Not required.
dirSpec	Specify which folders and sub-folders will be included in the project. Each entry in the `allEntry` array identifies a project folder and includes the following fields:	
symbolicLink	String	The path display name.

Schema Field	Type	Description

rootPath	String	Specify the root path of the folder. For example:
		`C:\\Users\\bross\\Downloads\\fileset`
		You could also specify the path as:
		`C:/Users/bross/Downloads/fileset`

exclusionPathSet	String	Specify the relative paths of any sub-folders that should not be included when crawling the project. For example:
		`C:\\Users\\adunna\\Downloads\\fam`
		You could also specify the path as:
		`C:/Users/adunna/Downloads/fam`

treeCrawl	Boolean	These fields are for output purposes. No input is required.
propertyMap	String	
privilegesSettings	Boolean	A set of Boolean values used to specify the default permissions users should be granted for the project: • **fprojectAdminPrivilege:** Perform administrative actions on the project. • **fReadPrivilege**: Read the contents of the project. • **fSearchPrivilege**: Search the contents of the project. • **fSharePrivilege**: Not supported. • **fWebDavPrivilege**: Not supported. • **fAuditPrivilege**: View audit trial data for the project.
userId	Integer	Specify the user ID assigned to the Box project owner.
username	String	Specify the user name of the Box project owner.
emailId	String	Specify the email address of the Box project owner.

Response Schema Fields

The fields included in the response schema are the same set of fields required in the request schema, as well as the ID that has been assigned to the Box project. These fields define the project configuration.

PUT: Update File Share Project (updatefileshareproject)

Update a file share project's configuration.

Path

http://localhost:8080/ps/api/v1/storage/project/updatefileshareproject

Request Parameters

None.

Request Body Schema Fields

The schema fields required to update the file share project's configuration are the same as those that must be specified when creating the file project, as described in the *POST: Create File Share Project (createfileshareproject)* section on page 340. You must also specify the ID of the project being updated.

Response Schema Fields

The schema fields returned for the project are the same as those required in the request schema when creating it (or when updating it using this endpoint). They represent the configuration of the project. For descriptions of these fields, refer to the *POST: Create File Share Project (createfileshareproject)* section on page 340.

PUT: Update OneDrive Project (updateonedriveproject)

Update a OneDrive project's configuration.

Path

http://localhost:8080/ps/api/v1/storage/project/updateonedriveproject

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the project to be updated. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
ownerId	Integer	Specify the ZLP user ID assigned to the user who should be the project owner. For policy resolution and ease of management purposes, the project will be associated with the department of the ZL UA user specified as the project owner.
type	Integer	Specify the project type. Enter 10 for OneDrive.
name	String	Specify the project's name.
displayName	String	Specify the project's display name.
mailServerId	Integer	Type the server ID assigned to the project's OneDrive server. You can retrieve server IDs with the GET: Get All Severs endpoint, as described on page 230.
domainId	Integer	Specify the domain ID.
crawlType	Integer	Specify the crawl type: • **400**: Metadata Analysis (No Index), • **200**: Metadata Index + Content Analysis, • **100**: Metadata Index + Content Analysis + Content Index

Schema Field	Type	Description
scheduleStartDate **scheduleEndDate**	String	Specify the start and end dates during which disposition should be carried out upon the project. Use the following formats for dates: YYYY-MM-DDTMM:HH:SS.SSZ For example: **2024-11-01T11:05:44.575Z** This setting is applicable to projects where the Disposition Workflow is enabled, meaning that the **disposition_enabled** setting is set to True. In addition, scheduling for the Disposition Workflow must be enabled, meaning that the **fScheduleEnabled** setting is set to True.
fScheduleEnabled	Boolean	Specify whether disposition scheduling should be enabled for the project. This setting is applicable to projects where the disposition setting (**disposition_enabled**) is enabled.
allowVersioning	Boolean	Specify whether file versioning should be used in this project. This is not required for FAM projects.

Schema Field	Type	Description
flags	Boolean	A set of Boolean values used to specify whether each project attribute should be enabled or not. Set to True to enable, or False to disable. • **allowsAddition**: Allow new files to be added to the project during file crawls. • **disable_acl_capture**: Disable the capture of ACL permissions for all folders and files in the project. In this case, the ZL File Connector will not check the ACL permissions while crawling. • **add_users_based_folder_acl**: Any ZL UA user with at least 'Read' ACL privileges for a folder within the project will automatically have access to the folder in the project in the ZL File Archiving application. • **fetch_file_acl**: Retrieve the ACL list for each file included in the project. • **crawlfetchlatestver**: Retrieve the latest versions of previously added files during project crawls. • **lock_crawl**: Disable future crawls of the project. • **disable_full_crawl**: Indicate whether the entire project should be scanned (and subject to archiving) when it is crawled. • **disposition_enabled**: Enable disposition on the project. Disposition is the process by which files whose records management lifecycle has expired are deleted and removed from the ZL UA system. • **disposition_approval_required**: Require approval for Disposition Runs within the project.

Schema Field	Type	Description
folderRoot	These are output fields that are used to define the project's root folder. Input is not required here when using the endpoint.	
dirId	String	Input to these fields is not required.
parentId	Integer	
name	String	
displayName	String	
description	String	
relativePath	String	
type	Integer	
storageSize	Integer	
itemCount	Integer	
deleted	Boolean	This field is used for output purposes. Input is not required.
dirSpec	Specify which folders and sub-folders will be included in the project. Each entry in the `allEntry` array identifies a project folder and includes the following fields:	
symbolicLink	String	The path display name.

rootPath	String	Specify the root path of the folder. For example:
		`C:\\Users\\bross\\Downloads\\fileset`
		You could also specify the path as:
		`C:/Users/bross/Downloads/fileset`

exclusionPathSet	String	Specify the relative paths of any sub-folders that should not be included when crawling the project. For example:
		`C:\\Users\\adunna\\Downloads\\fam`
		You could also specify the path as:
		`C:/Users/adunna/Downloads/fam`

Schema Field	Type	Description
treeCrawl	Boolean	These fields are for output purposes. No input is required.
propertyMap	String	
privilegesSettings	Boolean	A set of Boolean values used to specify the default permissions users should be granted for the project: • **fprojectAdminPrivilege:** Perform administrative actions on the project. • **fReadPrivilege**: Read the contents of the project. • **fSearchPrivilege**: Search the contents of the project. • **fSharePrivilege:** Not supported. • **fWebDavPrivilege**: Not supported. • **fAuditPrivilege**: View audit trial data for the project.
crawlAllVersions	Boolean	Set to True to archive all the versions of a file. If the check box is not selected, only the latest version of the file is crawled.

Response Schema Fields

The fields included in the response schema are the same set of fields required in the request schema. These fields define the project configuration.

PUT: Update SharePoint Project (updatesharepointproject)

Update a SharePoint project's configuration.

Path

UAA: http://localhost:8080/ps/api/v1/storage/project/updatesharepointproject

Request Parameters

None.

Request Body Schema Fields

The schema fields required to update the SharePoint project's configuration are the same as those that must be specified when creating the project with the POST: Create SharePoint Project endpoint, as described in the *POST: Create SharePoint Project (createsharepointproject)* section on page 345. You must also specify the ID of the project to be updated.

Response Schema Fields

The schema fields returned for the project are the same as those required in the request schema when creating it with the POST: Create SharePoint Project endpoint (or when updating it using this endpoint). They represent the configuration of the project. For descriptions of these fields, refer to the *POST: Create SharePoint Project (createsharepointproject)* section on page 345.

UAA/Roles

A role is an application or department-level set of permissions that determines what operations users can perform within the system. Roles can be assigned globally, or for a specific department(s). For example, a Global Discovery Manager role would enable the user's assigned role for all cases. A Discover Manager role for a specific department would restrict the user's role to the cases defined within that department.

The following sections describe the UAA/Roles endpoints available in the REST API. You can use these endpoints to grant and revoke roles within ZL UA:

- *GET: Get All Custom Roles (getallcustomroles):* Retrieve a list of custom roles that have been added to the system.
- *GET: Get All System Roles (getallsystemroles):* Retrieve a list of system roles that are included in the system.
- *GET: Get User Roles (getroleofuser):* Retrieve a list of the roles that have been assigned to a user.
- *PUT: Grant User Roles (grantroles):* Grant roles to a user.
- *PUT: Revoke User Roles (revokeroles):* Revoke roles from a user.

GET: Get All Custom Roles (getallcustomroles)

Retrieve a list of all custom roles in the system. A custom role is a combination of different ZL system roles.

Path

http://localhost:8080/ps/api/v1/uaaroles/getallcustomroles

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Roles: Includes an entry for each custom role in the system. The following fields are included in each entry:		
roleId	Integer	The role ID.
displayName	String	The role's display name.
description	String	A description of the role.
fSystemRole	Boolean	Indicates whether the role is a system role (True) or not.
appId	Integer	The application ID assigned to the custom role. For custom roles, the application is primarily an organizational feature. The application you choose determines the application role group in which the custom role will be included.
auditRecordLevel	Integer	The audit trail record value that will be recorded when a user with this role performs an action.
auditClearanceLevel	Integer	The minimum value that the role should have to view actions recorded by other roles.
allGrantableRolesIds	Integer	The IDs of the roles that users assigned this custom role can grant to other users.
allOperationNames	String	A list of strings identifying the operations allowed by the custom role.
systemRoleIds	Integer	A list of integer ID values identifying the system roles included in the custom role.

GET: Get All System Roles (getallsystemroles)

Retrieve a list of all system roles. The schema fields include in the response includes the ID and display names of all system roles, as well as information identifying the operations allowed by each role.

Path

http://localhost:8080/ps/api/v1/uaaroles/getallsystemroles

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Roles: Includes an entry for each system role. The following fields are included in each entry:		
roleId	Integer	The role ID.
displayName	String	The role's display name.
description	String	A description of the role.
fSystemRole	Boolean	Indicates whether the role is a system role (True) or not.
appId	Integer	The application ID assigned to the system role.
auditRecordLevel	Integer	The audit trail record value that will be recorded when a user with this role performs an action.
auditClearanceLevel	Integer	The minimum value that the role should have to view actions recorded by other roles.
allGrantableRolesIds	Integer	The IDs of the roles that users assigned this custom role can grant to other users.
allOperationNames	String	A list of strings identifying the operations allowed by the custom role.
systemRoleIds	Integer	A list of integer ID values identifying the other system roles included in this role.

GET: Get User Roles (getroleofuser)

Retrieve a list of the roles assigned to a user.

Path

http://localhost:8080/ps/api/v1/uaaroles/getroleofuser/{zlpuserId}

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	The ZLP user ID assigned to the user whose roles you want to view. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
User Roles: Includes an entry for each role that has been assigned to the user. The following fields are included in each entry:		
roleId	Integer	The role ID.
scope	String	The scope of the role. A role can be granted globally so that it is applicable to all departments, or it can be granted to specific departments only: • **Global**: Global • **InclRecur**: On selected departments recursively • **Incl**: On selected departments only
allScopeDomainIDs	Integer	Specifies the domain IDs of the departments the role is applicable to for roles that are only granted on selected departments. These are retrieved from the **ArchiveServer** database table.

PUT: Grant User Roles (grantroles)

Grant roles to a user.

Path

http://localhost:8080/ps/api/v1/v1/uaaroles/grantroles

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to grant roles to. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
User Roles: The remaining input is an array of fields defining the roles to be granted. Specify the following for each **userRole** entry:		
roleId	Integer	The role ID. For information on retrieving role IDs, refer to the following sections: • *GET: Get All Custom Roles* on page 370 • *GET: Get All System Roles* on page 372
roleName	String	The role name. For information on retrieving role names, refer to the following sections: • *GET: Get All Custom Roles* on page 370 • *GET: Get All System Roles* on page 372
scope	String	Specify whether the role should be granted globally so that it is applicable to all departments, or if it should be granted to specific departments only: • **Global**: Global • **InclRecur**: On selected departments recursively • **Incl**: On selected departments only

Schema Field	Type	Description
allScopeDomainIds	Integer	An array of domain IDs to specify the departments the role is applicable to for roles that are only granted on selected departments. These can be retrieved from the **ArchiveServer** database table. Specify the domain IDs as a comma-separated list. For example: `"AllScopeDomainIds": [ 0,1,2,3 ]`

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each role specified in the request.		
success	Boolean	Indicates whether the role was granted successfully (True) or not (False).
result	String	The result of the request. The string will indicate how the role has been applied (on which departments, scope, role ID, etc).
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

PUT: Revoke User Roles (revokeroles)

Revoke roles that have been previously assigned to a user.

Path

http://localhost:8080/ps/api/v1/uaaroles/revokeroles

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to modify. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
roles	Integer	An array of role IDs to specify the roles to be revoked. Specify the role IDs as a comma-separated list. For example: `"Roles": [ 0,1,2,3 ]` For information on retrieving role IDs, refer to the following sections: • *GET: Get All Custom Roles* on page 370 • *GET: Get All System Roles* on page 372

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each role specified in the request.		
success	Boolean	Indicates whether the role was revoked successfully (True) or not (False).
result	String	The result of the request. If successful, a message will display indicating that the role has been revoked.
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

FAM/Security Groups

A security group represents a specific group of users in ZL UA. When you assign project privileges to a security group within the FAM module, those privileges are granted to all users in the security group. The following sections describe the FAM/Security Groups endpoints available in the REST API. You can use these endpoints to create and manage security groups:

- *PUT: Ensure Security Group (ensuresecuritygroup):* Create a security group.
- *GET: Get All Security Groups (getallsecuritygroups):* Retrieve a list of the security groups that have been added to the system.
- *PUT: Remove Security Group (removesecuritygroup):* Remove a security group.

PUT: Ensure Security Group (ensuresecuritygroup)

Create a security group.

Path

http://localhost:8080/ps/api/v1/SecurityGroup/ensuresecuritygroup

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
aIdZlpUser	Integer	An array of ZLP user IDs to specify the users who should be added to the security group. Specify the user IDs as a comma-separated list. For example: `"aIdzlpUser": [ 0,1,2,3 ]`

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each user specified in the request.		
success	Boolean	Indicates whether the users were added successfully (True) or not (False).
result	String	The result of the request. If **Success** is True, then this contains information regarding the security group.
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

GET: Get All Security Groups (getallsecuritygroups)

Retrieve a list of all security groups.

Path

http://localhost:8080/ps/api/v1/SecurityGroup/getallsecuritygroups

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Security Groups: Includes an entry for each user added to the security group. The response schema fields included in each entry are the same as those included in the response returned after creating the security group. For descriptions of these fields, refer to *POST: Create User* on page 409.		

PUT: Remove Security Group (removesecuritygroup)

Remove a security group.

Path

http://localhost:8080/ps/api/v1/SecurityGroup/removesecuritygroup

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
aIdZlpUser	Integer	An array of ZLP user IDs to specify the security groups to be removed. Specify the security group IDs as a comma-separated list. For example: `"aIdzlpUser": [ 0,1,2,3 ]`

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each security group specified in the request.		
success	Boolean	Indicates whether the security group was removed successfully (True) or not (False).
result	String	The result of the request. If **Success** is set to True, this will indicate that the security group was removed successfully.
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

UAA/Agents

Server agents are components that perform various tasks on the servers, such as crawling a mail server for mailbox data, archiving a file server or performing user synchronization. Generally, each server added to ZL UA must have at least one server agent. The following sections describe the UAA/Agents endpoints available in the REST API:

- *POST: Create a Server Agent (createserveragent):* Create a File Archive server or a Mailbox Crawl agent. These agents can be used with multiple server types, as specified later.
- *GET: Get All Agents of Mail Server (getagentsusingidmailserver)*: Retrieve the configurations of the server agents that have been added to a specific server.

POST: Create a Server Agent (createserveragent)

Create a server agent for the following purposes:

- **File Archive:** Any agent to archive files from any of the following server types: File Share, SharePoint, Google Drive and OneDrive.
- **Mailbox Crawl:** Any agent to crawl mailboxes for any of the following server types: Google Mail, Microsoft Exchange and Microsoft EWS.

Path

http://localhost:8080/ps/api/v1/Agents/createserveragent

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
agentName	String	Enter the server agent's name.
agentType	String	Specify the server agent type: • For a file share archiving agent, enter **file**. • For a SharePoint archiving agent, enter **sharepointFile**. • For a GoogleDrive archiving agent, enter **gdDriveFile**. • For a OneDrive archiving agent, enter **onedriveFile**. • For an agent to crawl mailboxes for Google Mail, Microsoft Exchange and Microsoft EWS servers, enter **mbCrawl**.
mask	Integer	Specify the desired mask number. Each server agent can be associated with multiple server masks. The users on the mail server get distributed almost equally among Mail Server Agent Global Tasks corresponding to associated masks when the agent runs. This is useful in properly scheduling the times at which mailbox crawling is performed and scaling up the crawling performance.

Schema Field	Type	Description
useSystemDefault	Boolean	Set to True to designate that the agent should inherit the default run interval time. Set to False to specify a custom run interval time for the server agent with the **runInterval** field.
runInterval	Integer	If **useSystemDefault** is set to False, specify the custom run interval time (in seconds) for the server agent.
mailServerId	Integer	Specify the ID of the server the agent will run on. You can retrieve server IDs with the GET: Get All Severs endpoint described on page 230.

Response Schema Fields

Schema Field	Type	Description
agentName	String	The server agent's name.
agentType	String	The server agent's type.
mask	Integer	The server agent's mask value. Each server agent can be associated with multiple server masks. The users on the mail server get distributed almost equally among Mail Server Agent Global Tasks corresponding to associated masks when the agent runs. This is useful in properly scheduling the times at which mailbox crawling is performed and scaling up the crawling performance.
useSystemDefault	Boolean	If True, the agent will inherit the default run interval time. If False, the agent will use a custom run interval time.
runInterval	Integer	The run interval time (in seconds) for the server agent.
mailServerId	Integer	The ID of the server the agent will run on.
mapParams	String	Additional information about the agent. This varies depending on the agent type. For a mailbox crawling agent, it will indicate how many threads the agent uses. For a user synchronization agent, it will indicate the agent's options for automatic user creation and termination.

Schema Field	Type	Description
iterationId	Integer	These fields provide information about the last iteration of the server agent, i.e., the last time the server agent ran. This includes the date and time the last iteration started and ended, and the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

GET: Get All Agents of Mail Server (getagentsusingidmailserver)

Retrieve all server agents that have been added to a particular server.

Path

http://localhost:8080/ps/api/v1/Agents/getagentsusingidmailserver/{idMailServer}

Request Parameters

Parameter	Type	Description
idMailServer	Integer	Specify the ID of the server that the server agent belongs to. You can retrieve server IDs with the GET: Get All Severs endpoint described on page 230.
page **pageSize**	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Server Agents: The endpoint returns an entry for each server agent that has been added to the specified server. Each entry includes the following fields:		
agentName	String	The server agent's name.
agentType	String	The server agent's type.
mask	Integer	The server agent's mask value. Each server agent can be associated with multiple server masks. The users on the mail server get distributed almost equally among Mail Server Agent Global Tasks corresponding to associated masks when the agent runs. This is useful in properly scheduling the times at which mailbox crawling is performed and scaling up the crawling performance.

Schema Field	Type	Description
useSystemDefault	Boolean	If True, the agent will inherit the default run interval time. If False, the agent will use a custom run interval time.
runInterval	Integer	The run interval time (in seconds) for the server agent.
mailServerId	Integer	The ID of the server the agent will run on.
mapParams	String	Additional information about the agent. This varies depending on the agent type. For a mailbox crawling agent, it will indicate how many threads the agent uses. For a user synchronization agent, it will indicate the agent's options for automatic user creation and termination.
iterationId	Integer	These fields provide information about the last iteration of the server agent, i.e., the last time the server agent ran. This includes the date and time the last iteration started and ended, and the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

FAM/Tags

Tags are customizable labels that can be applied to documents for various purposes. You could apply tags to the results of a search or file sampling to mark those files for retrieval later or apply tags to mark files that are subject to review, and so on.

You can also use tags for remediation. When you configure remediation, you assign an action to a tag (e.g., to copy, delete, or move the file). When you execute remediation, that action will be applied to all the files that the tag has been applied to. For example, you could use remediation to move all files that a tag has been applied to from one folder to another.

You can upload tag definition files and tag specifications into ZL UA to create tags for use in your system:

- A tag definition file defines and creates tags. These tags can be applied to files manually, or via a tag specification.
- A tag specification defines a set of rules and conditions, each of which specifies a tag that will be applied to files that meet the terms of the rules and conditions. For example, you could create a tag specification to tag all files that contain the phrase confidential agreement" in the body of an email with the "Privileged" tag. You can upload tag specifications that will tag files based on content, metadata and PII data.

The following sections describe how to manage tags in REST API, and how to upload tag definition files and tag specification files:

- *DELETE: Delete Tag (deleteTag)*
- *GET: Get All Tags of a Project (getAllTags)*
- *GET: Get Tag Using ID (getTagUsingId)*
- *GET: Get Tag Using Name (getTagUsingName)*
- *POST: Upload Content Spec (uploadContentTagSpec)*
- *POST: Upload MetaData Spec (uploadMetadataSpec)*
- *POST: Upload PII Tags (uploadPIIspec)*
- *POST: Upload Tag Definition File (uploadTags)*

DELETE: Delete Tag (deleteTag)

Delete a tag.

Path

http://localhost:8080/ps/api/v1/tags/tags/deletetag/{projectId}/{tagId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project that the tag to be deleted belongs to. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
tagId	Integer	Specify the ID of the tag to be deleted. You can retrieve tag IDs with the GET: Get All Tags of a Project (getalltags) endpoint described on page 391.

Request Schema Fields

None.

Response Schema Fields

None.

GET: Get All Tags of a Project (getAllTags)

Retrieve the tags that have been created within a specific project.

Path

http://localhost:8080/ps/api/v1/tags/tags/getalltags/{projectId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
page **pageSize**	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.

Request Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
Tag Entries: The endpoint returns an entry for each tag that has been added to the specified project. The schema fields included in each entry are described below.		
Parent	String	The name of the tag's parent tag.
Id	Integer	The tag ID.
projectId	Integer	The ID of the project the tag belongs to.
parentId	String	The ID of the tag's parent tag.
Name	String	The internal name of the tag.
displayName	String	The display name of the tag.

Schema Field	Type	Description
tagFlags	Array	An array of Boolean values indicating the status of various tag attributes.
root_node	Boolean	If True, it indicates that the tag is a root-level tag.
read_only		If True, it indicates that the tag is read-only.
enduser_tag		If True, it indicates that the tag can be applied manually.
auto_tag		If True, it indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag		If True, it indicates that the tag is a mutually exclusive tag.
tag_32		If True, it indicates that the tag is a PII tag.
tag_64		If True, it indicates that the tag is a content tag.
description		A description of the tag.
createDate		The date and time that the tag was created.

GET: Get Tag Using ID (getTagUsingId)

Retrieve the configuration of a specific tag. Specify the tag by its ID.

Path

http://localhost:8080/ps/api/v1/tags/tags/gettagusingid/{projectId}/{tagId}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project the tag belongs to. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
tagId	Integer	Specify the ID of the tag you want to view. You can retrieve tag IDs with the GET: Get All Tags of a Project (getalltags) endpoint described on page 391.

Request Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
parent	String	The name of the tag's parent tag.
id	Integer	The tag ID.
projectId	Integer	The ID of the project the tag belongs to.
parentId	String	The ID of the tag's parent tag.
name	String	The internal name of the tag.
displayName	String	The display name of the tag.

Schema Field	Type	Description
tagFlags	Array	An array of Boolean values indicating the status of various tag attributes.
root_node	Boolean	If True, it indicates that the tag is a root-level tag.
read_only		If True, it indicates that the tag is read-only.
enduser_tag		If True, it indicates that the tag can be applied manually.
auto_tag		If True, it indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag		If True, it indicates that the tag is a mutually exclusive tag.
tag_32		If True, it indicates that the tag is a PII tag.
tag_64		If True, it indicates that the tag is a content tag.
description		A description of the tag.
createDate		The date and time that the tag was created.

GET: Get Tag Using Name (getTagUsingName)

Retrieve the configuration of a specific tag. Specify the tag by its name.

Path

http://localhost:8080/ps/api/v1/tags/tags/gettagusingname/{projectId}/{tagName}

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project the tag belongs to. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338.
tagName	String	Specify the name of the tag you want to view. You can retrieve tag names with the GET: Get All Tags of a Project (getalltags) endpoint described on page 391.

Request Schema Fields

None.

Response Schema Fields

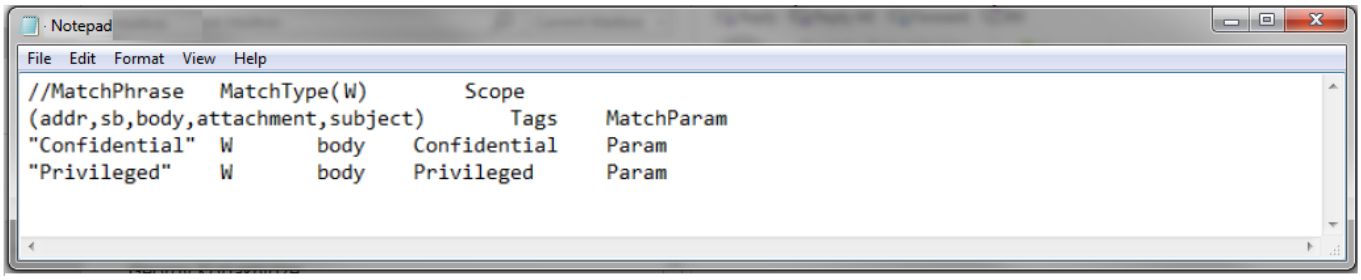
Schema Field	Type	Description
parent	String	The name of the tag's parent tag.
id	Integer	The tag ID.
projectId	Integer	The ID of the project the tag belongs to.
parentId	String	The ID of the tag's parent tag.
name	String	The internal name of the tag.
displayName	String	The display name of the tag.

Schema Field	Type	Description
tagFlags	Array	An array of Boolean values indicating the status of various tag attributes.
root_node	Boolean	If True, it indicates that the tag is a root-level tag.
read_only		If True, it indicates that the tag is read-only.
enduser_tag		If True, it indicates that the tag can be applied manually.
auto_tag		If True, it indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag		If True, it indicates that the tag is a mutually exclusive tag.
tag_32		If True, it indicates that the tag is a PII tag.
tag_64		If True, it indicates that the tag is a content tag.
description		A description of the tag.
createDate		The date and time that the tag was created.

POST: Upload Content Spec (uploadContentTagSpec)

Upload a Content Tag Specification. A Content Tag Specification is a formatted, tab-delimited TXT file that defines a set of rules, each of which includes a user-specified phrase and a tag. The rules are checked against the contents of each file within a project, and if a file includes the specified phrase, then the tag for the rule is applied to the file.

For example, you could create a rule to apply the “Privileged” tag to all files that contain the phrase “confidential agreement.” This rule is shown in the example below:



Each rule includes several parameters, and uses the following format:

```
//MatchPhrase MatchType Scope Tags Match Param
```

The parameter values must be separated by tabs. They are described below:

- **MatchPhrase**: The word or phrase to identify for tagging. This string must be surrounded by double quotes.
- **MatchType**: Must be set to “W,” indicating that a file’s content must match the word/phrase specified as the MatchPhrase for the tag to be applied.
- **Scope**: The part of the message to check for auto-tagging purposes. This must be set to “body.”
- **Tags**: The existing case tag to be automatically applied to case items that contain the specified MatchPhrase within the specified content Scope. This tag must already exist for the tag spec rule to be successfully added.
- **MatchParam**: Must be “Param.”

Please note that after uploading a Content Tag Specification into a project, you must execute the **Run Content Tagger** background task to apply the tag specification to the project, and then execute the **Update Index** and **Clear Cache** background tasks to update the project tag index and clear the cache. For more information, refer to *FAM/Tasks* on page 262.

Path

http://localhost:8080/ps/api/v1/tags/tags/uploadcontenttagspec

Request Parameters

None.

Request Schema Fields

Schema Field	Type	Description
projectId	Integer	The ID of the project that the Content Tag Specification will be uploaded to. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338. The tags defined within the specification will be applied to all files included in the project the next time the Run Content Tagger and Update Index background tasks are run. For more information, refer to *POST: Run Content Tagger* on page 265 and *POST: Update Index* on page 270.
tagContent	String	Copy the contents of the tab-delimited text file.

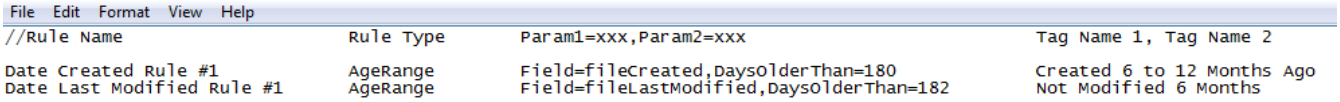
Response Schema Fields

A string indicating whether the upload was successful.

POST: Upload MetaData Spec (uploadMetaDataSpec)

Upload a Metadata Tag Specification. A Metadata Tag Specification is a formatted, tab-delimited TXT file that defines a set of rules to be compared to each file's metadata properties (e.g., date created, date last modified, date last accessed, ACL owner). Each rule specifies 1 or 2 tags that will be applied to files with metadata that satisfy the conditions of the rule. For example, a rule can be created to apply a tag to all files that were created within a specified date range.

The following is an example metadata tag specification that defines two rules:



Each rule consists of several tab-separated fields, and uses the following format:

```
//RuleName  RuleType     Param1=xxx,Param2=xxx   Tag Name 1, Tag Name 2
```

The fields are described below:

Rule Name

The name of the rule.

Rule Type

The type of rule. This includes the following options:

- **AgeRange:** Check the age of a file.
- **ACLDepartment**: Check the ACL department associated with a file.
- **PathRegex**: Check if the file name or folder name includes the term defined by a RegEx (regular expression) pattern. The FAM module supports the standard RegEx syntax.

Param1=xxx, Param2=xxx

These determine how the rule will be compared to each file's metadata. A rule may include one or more parameters, which must be comma-separated. The available parameters vary depending on the **RuleType** selected:

- **AgeRange:** This includes the following options:

  fileLastModified, fileLastAccessed, fileCreated, DaysOlderThan, DaysYoungerThan

  Param1 should specify which file attribute is being examined: the date the file was last modified, the date it was last accessed, or the date it was created (**fileLastModified**, **fileLastAccessed**, or **fileCreated**). Each rule can check one of these attributes.

  Param2 should specify how the attribute should be checked (**DaysOlderThan** or **DaysYoungerThan**).

For example, the following parameters would return True and tag files that were created more 180 days ago:

```
Field=fileCreated, DaysOlderThan=180
```

The following examples would return True and tag files that were modified in the last 30 days:

```
Field=fileLastModified, DaysYoungerThan=30
```

The following examples would return True and tag files that were last accessed between 30 and 60 days ago:

```
Field = fileLastAccessed, DaysOlderThan=30, DaysYoungerThan=60
```

- **ACLDepartment:** This includes the following options:

  OwnerOnly (true/false), Departmentname = [name of department in ZL]

  To check if file's ACL owner belongs to a certain department, use **OwnerOnly** as **Param1** and the **DepartmentName** as **Param2**. For example, the following parameters would return True and tag any file whose "ACL owner" belongs to the "zlps" department:

  ```
  OwnerOnly=true, DepartmentName=zlps
  ```

  To check if a file's ACL user list includes members of a certain department, use **DepartmentName** as **Param1** and omit the second parameter. The following would return True and tag any file that has users from the "zlps" departments on its ACL:

  ```
  DepartmentName=zlps
  ```

- **PathRegex:** ParseFolderNames (true/false), ParseFileName (true/false), Regex

  Param1 should specify whether the file's name or the file's folder name should be checked (**ParseFolderNames** or **ParseFileName**).

  Param2 should define a regular expression (**Regex**). The condition will check the file name or the file's folder name for that expression. The FAM module supports the standard RegEx syntax.

  For example, the following rule would return True and tags files where the file name includes "HR or "human":

  ```
  ParseFileName=true, Regex=(?i)(hr|human)
  ```

Tag Name 1, Tag Name 2

Specify one or two tags that will be applied to the files that meet the specified conditions. These must match the names of existing tags (i.e., the metadata tag specification cannot create new tags).

Please note that after uploading a Metadata Tag Specification into a project, you must execute the **Run Metadata Tagger** background task to apply the tag specification to the project, and then execute the **Update Index** and **Clear Cache** background tasks to update the project tag index and clear the cache. For more information, refer to *FAM/Tasks* on page 262.

Path

http://localhost:8080/ps/api/v1/tags/tags/uploadmetadataspec

Request Parameters

None.

Request Schema Fields

Schema Field	Type	Description
projectId	Integer	The ID of the project the Metadata Tag Specification will be uploaded to. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338. The tags defined within the specification will be applied to all files included in the project the next time the Run Metadata Tagger and Update Index background tasks are run. For more information, refer to *POST: Run Metadata Tagger* on page 267 and *POST: Update Index* on page 270.
tagContent	String	Copy the contents of the tab-delimited text file.

Response Schema Fields

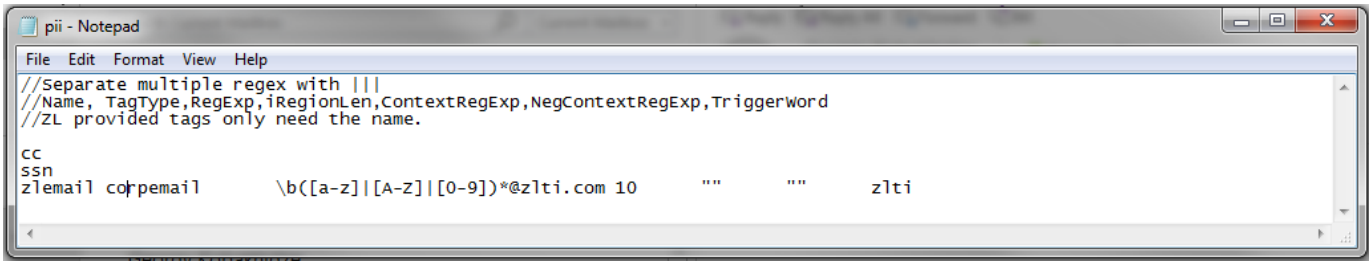
A string indicating whether the upload was successful.

POST: Upload PII Tags (uploadPIIspec)

Upload a PII Tag Specification. A PII Tag Specification is a formatted, tab-delimited TXT file defining a set of rules that will be compared to each file's content. Each rule specifies a tag that will be applied to files with content that satisfy the conditions of the rule.

This is similar to Content Tag Specifications. However, PII tag specifications differ from content tag specifications in several ways:

- PII patterns are intended to be used to search specifically for files that include information like credit card numbers, social security numbers, and addresses. For example, a rule can be created to tag all items that contain an email address or a social security number.
- PII tag specifications not only define the rules used to determine which files PII tags should be applied to, but also define the PII tags themselves. PII tag specifications cannot use existing tags.



Each rule includes several parameters, and uses the following format:

```
//Name, TagType, RegExp, iRegionLen, ContextRegExp, NegContextRegExp, TriggerWord
```

The parameter values must be separated by tabs. They are described below.

- **Name**: The name of the PII tag that will be created and applied to files that meet the conditions of this rule. Each PII tag must be assigned a unique name.
- **TagType:** A description of the PII tag. This can match the name, if desired.
- **RegExp**: A regular expression (RegEx) to search for. The FAM module supports the standard RegEx syntax. A file must include this expression in order for the PII tag to be applied.
- **iRegionLen**: Optional. Specify the length of the region (in characters) to look for the terms specified for the ContextRegExp and/or NegContextRegExp fields in relation to the term specified for the **RegEx** field.
- **ContextRegExp** [Optional]: An additional RegEx term. If specified, this term must occur within X characters of the term defined by the RegEx field for the PII tag to be applied, where X is the value of the **iRegionLen** field.
- **NegContextRegExp** [Optional]: An additional RegEx term. If specified, this term may **not** occur within X characters of the term defined by the **RegEx** field for the PII tag to be applied, where X is the value of the **iRegionLen** field.
- **TriggerWord** == [Optional]: An additional keyword(s) that the file must include for the PII tag to be applied. These keyword(s) can be anywhere in the file, meaning that they are not affected by the **iRegionLen** field.

- You can apply multiple trigger keywords. They must be separated with the following characters:

  |||

  For example, if you wanted to specify “litigation”, “copyrite” and “trial” as additional keywords, you would enter:

  ```
  litigation ||| copyrite ||| trial
  ```

Please note that after uploading a PII Tag Specification into a project, you must execute the **Run PII Tagger** background task to apply the tag specification to the project, and then execute the **Update Index** and **Clear Cache** background tasks to update the project tag index and clear the cache. For more information, refer to *FAM/Tasks* on page 262.

Path

http://localhost:8080/ps/api/v1/tags/tags/uploadPIISpec

Request Parameters

None.

Request Schema Fields

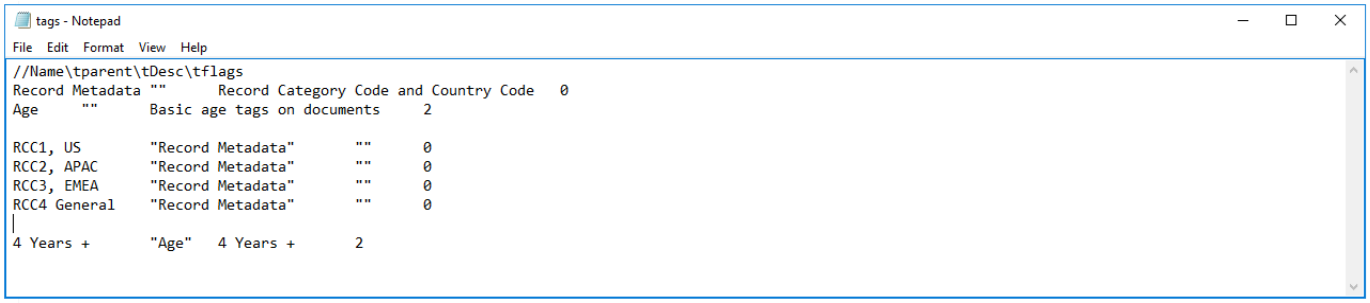
Schema Field	Type	Description
projectId	Integer	The ID of the project the PII Tag Specification will be uploaded to. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project’s configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338. The tags defined within the specification will be applied to all files included in the project the next time the Run Content Tagger and Update Index background tasks are run. For more information, refer to *POST: Run PII Tagger* on page 268 and *POST: Update Index* on page 270.
tagContent	String	Copy the contents of the tab-delimited text file.

Response Schema Fields

A string indicating whether the upload was successful.

POST: Upload Tag Definition File (uploadTags)

Upload a Tag Definition File. A Tag Definition File is a formatted, tab-delimited TXT file that defines new tags to be applied within a project. The following image depicts a sample tag definition file:



Each tag definition consists of four tab-separated fields, and uses the following format:

```
//Name	tParent	tDesc tFlags
```

The fields are described below:

- **Name:** The name of the tag to be created.
- **tParent:** The name of the existing parent tag under which the new tag will be created. The parent tag must already exist for the new tag to be added successfully.

  If the new tag doesn't have a parent, leave this field empty as " ".
- **tDesc**: The description of the tag.
- **tFlags:** Specifies how the tag can be applied to files. If the tag is to be applied manually or via a tag specification, enter 0. If the tag is only to be applied via a tag specification only, enter 2.

Please note that after uploading a Tag Definition File into a project, you must execute the **Update Index** and **Clear Cache** background tasks to update the project tag index and clear the cache. For more information, refer to *FAM/Tasks* on page 262.

Path

http://localhost:8080/ps/api/v1/tags/tags/uploadtags

Request Parameters

None.

Request Schema Fields

Schema Field	Type	Description
projectId	Integer	The ID of the project the Tag Definition File will be uploaded to. The REST API includes several endpoints you can use to retrieve project ID values and other configuration data. For example, you could use the GET: Get Projects Using Search endpoint to retrieve a project's configuration by searching for its name. For more information, refer to *UAA/Projects and FAM/Projects* on page 338. The tags defined within the specification will be available for application within that project the next time the Update Index background task is run. For more information, refer to *POST: Update Index* on page 270.
tagContent	String	Copy the contents of the tab-delimited text file.

Response Schema Fields

A string indicating whether the upload was successful.

UAA/Users

A user is a person whose email address(es) and alias(es) are registered in the ZL UA system. All users registered in ZL UA are associated with a department, from which those users inherit policy settings. However, privileged users can configure custom settings to override the inherited department settings for a particular user. A ZL UA user typically has a primary email address, as well as one or more *alias* email addresses which can be used to locate that user.

The following sections describe the UAA/Users endpoints available in the REST API. You can use these endpoints to create and manage users within ZL UA:

- *PUT: Add User Alias (addUserAlias)*
- *POST: Create User (createUser)*
- *DELETE: Delete User (deleteUser)*
- *GET: Get All Department Users (getAllDepartmentUsers)*
- *GET: Get All User Aliases (getAllUserAliases)*
- *GET: Get User Using Address (getUserUsingAddress)*
- *GET: Get User Using Alias Address (getUserUsingAliasAdress)*
- *GET: Get User Using External Reference (getUserUsingExternalReference)*
- *GET: Get User Using Owner (getUserUsingOwner)*
- *GET: Get User Using ZLP ID (getUserUsingId)*
- *PUT: Move User to a New Department (moveUserToNewDepartment)*
- *PUT: Remove User Alias (removeAlias)*
- *PUT: Restore Terminated User (restoreTerminatedUser)*
- *PUT: Terminate User (terminateUser)*
- *PUT: Update User Account Information (updateUserAccountInfo)*
- *PUT: Update User Email Address (updateUserEmailAddress)*
- *PUT: Update User Mail Server Information (updateUserMailServerInfo)*
- *PUT: Update User Owner (updateUserOwnerField)*
- *PUT: Update User Sync Status (updateUserSyncExclude)*

PUT: Add User Alias (addUserAlias)

Add an alias to a user.

Path

http://localhost:8080/ps/api/v1/users/adduseralias

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	Integer	Specify the ZLP user ID of the user you want to add the alias to. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.
alias	String	Enter the alias name.
aliasType	Integer	Specify the alias type: • 0: Default (Email) • 1: X500 DN • 2: Address • 3: Manual • 4: Exchange Legacy DN • 5: Transformed Lotus DN • 6: NetBios User Name • 100: IM • 200: Bloomberg • 300: Parlano

Response Schema Fields

Schema Field	Type	Description
alias	String	The alias name.
zlpUserId	Integer	The ZL user ID of the user the alias has been added to.

Schema Field	Type	Description
archiveServerDept	Integer	The department the user is assigned to.
iType	Integer	The alias type: • 0: Default (Email) • 1: X500 DN • 2: Address • 3: Manual • 4: Exchange Legacy DN • 5: Transformed Lotus DN • 6: NetBios User Name • 100: IM • 200: Bloomberg • 300: Parlano
dateCreated	String	The date and time the alias was created.

POST: Create User (createUser)

Create a new user. After creating the user, you should configure the user's mail server information with the PUT: Update Mail Server Information endpoint. For more information, refer to *PUT: Update User Mail Server Information* on page 427.

Path

http://localhost:8080/ps/api//v1/users/createuser

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
departmentName	String	Specify the name of the department to add the user to.
userAddress	String	Specify the primary email address associated with the user.
owner	String	Specify the user's owner. This is meant to identify the user's manager or the user's creator.
externalReference	String	Enter any unique IDs or information used outside of ZL UA that is relevant to the user here.
userTags	String	Specify any user tags to apply to the user as a comma-separated list.
retentionTags	String	Specify any retention tags to apply to the user as a comma-separated list.
altReviewDepartment	String	Specify the Alternative Review Department to add the user to.
fullName	String	The first and last name of the user.
userType	Integer	Specify 0 to create a new user, or 100 to create a user group.
dateHired	String	The date the user was hired. This should match the format shown in Swagger.
fTerminated	Boolean	Specify if the user has been terminated (True) or not (False).
dateTerminated	String	Specify the date the user was terminated, if applicable. This should match the format shown in Swagger.

Schema Field	Type	Description
miscField1 **miscField2**	String	Use these fields to enter any additional information required for the user.
fSyncIncluded	Boolean	Specify whether the user should be included (True) in LDAP User Synchronizations or not (False).

Response Schema Fields

Schema Field	Type	Description
idZlpUser	Integer	The ZL user ID assigned to the user.
type	Integer	The user type: 0 for a user, or 100 for a user group.
address	String	The user's primary email address.
owner	String	The user's owner. This is meant to identify the user's manager or the user's creator.
extReference	String	Unique ID or information used outside of ZL UA that is relevant to the user.
userTags	String	User tags applied to the user.
retTags	String	Retention tags applied to the user.
altReviewDepts	String	The Alternate Review Department(s) the user is assigned to.
deptName	String	The department the user is assigned to.
reviewDeptName	String	The Review Department the user is assigned to.
mailServerName	String	The mail server for this user.
mailStoreInfo	String	Mail store information for this user.
syncExclude	Boolean	Indicates whether the user should be excluded (True) from User Synchronizations or not (False).
archive	Boolean	Set to True if the user is available for archiving and journaling, respectively.
journal	Boolean	

Schema Field	Type	Description
fullName	String	The first and last name of the user.
dateCreate	String	The date and time the user was created.
dateLastUpdate	String	The date and time the user last updated.
connectUserId	String	The user ID used to connect to the user's mail server.
dateHire	String	The date and time the user was hired.
dateTerminated	String	The date and time the user was terminated, if applicable.
terminated	Boolean	Indicates whether the user has been Terminated (True) or not.
dateIterStart	String	These fields indicate that date and time that the last user synchronization process started and ended, and the date and time that the user's information was updated during synchronization.
dateIterEnd	String	
dateIterUpdate	String	
dateFullScanStart	String	The date and time that the last full scan of the user's mailbox began.
dateFullScanEnd	String	The date that the last full scan of the user's mailbox ended.
dateArchiveBegin	String	The date and time that archiving of the user's date began.
miscField1 **miscField2**	String	Additional information entered for the user.

DELETE: Delete User (deleteUser)

Delete a user.

Path

http://localhost:8080/ps/api/v1/users/deleteuser/{zlpUserId}

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to delete. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.

Request Body Schema Fields

None.

Response Schema Fields

A string message indicating whether the user was successfully deleted or not.

GET: Get All Department Users (getAllDepartmentUsers)

Retrieve a list of users that are assigned to a specific department.

Path

http://localhost:8080/ps/api/v1/users/getalldepartmentusers/{departmentName}

Request Parameters

Parameter	Type	Description
departmentName	String	Specify the name of the department whose users you want to view. The REST API includes several endpoints you can use to retrieve department configuration names. For further details on these endpoints, refer to *UAA/Departments* on page 90
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
User Entries: Includes an entry for user that is included in the specified department. The fields included in each entry are the same as those included in the response after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.		

GET: Get All User Aliases (getAllUserAliases)

Retrieve a list of aliases that are assigned to a specific user.

Path

http://localhost:8080/ps/api/v1/users/getalluseraliases/{zlpUserId}

Request Parameters

Parameter	Type	Description
zlpUserId	String	Specify the ID of the user whose aliases you want to view. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
User Alias Entries: Includes an entry for each alias that has been assigned to the user. The following fields are included in each entry:		
alias	String	The alias name.
zlpUserId	Integer	The ZL user ID of the user the alias has been added to.
archiveServerDept	Integer	The department the user is assigned to.

Schema Field	Type	Description
iType	Integer	The alias type: • 0: Default (Email) • 1: X500 DN • 2: Address • 3: Manual • 4: Exchange Legacy DN • 5: Transformed Lotus DN • 6: NetBios User Name • 100: IM • 200: Bloomberg • 300: Parlano
dateCreated	String	The date and time the alias was created.

GET: Get User Using Address (getUserUsingAddress)

Retrieve information for a user by specifying the user's primary email address.

Path

http://localhost:8080/ps/api/v1/users/getuserusingaddress/{userAddress}

Request Parameters

Parameter	Type	Description
userAddress	String	Specify the primary email address associated with the user in ZL UA.

Request Body Schema Fields

None.

Response Schema Fields

The response schema fields returned for the specified user are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

GET: Get User Using Alias Address (getUserUsingAliasAddress)

Retrieve information for a user by specifying an alias assigned to the user.

Path

http://localhost:8080/ps/api/v1/users/getuserusingaliasaddress/{aliasAddress}

Request Parameters

Parameter	Type	Description
aliasAddress	String	Specify an alias assigned to the user in ZL UA.

Request Body Schema Fields

None.

Response Schema Fields

The response schema fields returned for the specified user are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

GET: Get User Using External Reference (getUserUsingExternalReference)

Retrieve information for a user by specifying the external reference data assigned to the user.

Path

http://localhost:8080/ps/api/v1/users/getuserusingexternalreference/{extReference}

Request Parameters

Parameter	Type	Description
extReference	String	Specify the external reference data assigned to the user in ZL UA.

Request Body Schema Fields

None.

Response Schema Fields

The response schema fields returned for the specified user are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

GET: Get User Using Owner (getUserUsingOwner)

Retrieve information for a user by specifying the owner assigned to the user.

Path

http://localhost:8080/ps/api/v1/users/getuserusingowner/{owner}

Request Parameters

Parameter	Type	Description
owner	String	Specify the name of the owner assigned to the user in ZL UA.

Request Body Schema Fields

None.

Response Schema Fields

The response schema fields returned for the specified user are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

GET: Get User Using ZLP ID (getUserUsingId)

Retrieve information for a user by specifying the ZLP ID assigned to the user.

Path

http://localhost:8080/ps/api/v1/users/getuserusingid/{zlpUserId}

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ZLP user ID assigned to the user in ZL UA. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.

Request Body Schema Fields

None.

Response Schema Fields

The response schema fields returned for the specified user are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

PUT: Move User to a New Department (moveUserToNewDepartment)

Move a user to a new department or review department.

Path

http://localhost:8080/ps/api/v1/users/moveusertonewdepartment

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to move. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.
newDeptInfoId	Integer	Specify the ID of the department you want to move the user to. The REST API includes several endpoints you can use to retrieve department ID values. For further details on these endpoints, refer to *UAA/Departments* on page 90.
newReviewDeptInfoId	Integer	Specify the ID of the review department you want to assign the user to. The REST API includes several endpoints you can use to retrieve department ID values. For further details on these endpoints, refer to *UAA/Departments* on page 90.

Response Schema Fields

The response schema fields returned for the user who is being moved are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

PUT: Remove User Alias (removeAlias)

Remove a user alias.

Path

http://localhost:8080/ps/api/v1/users/removealias/{zlpUserId}/{alias}

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to modify. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.
Alias	String	Specify the alias you want to remove.

Request Body Schema Fields

None.

Response Schema Fields

A string indicating whether the alias was removed successfully.

PUT: Restore Terminated User (restoreTerminatedUser)

Restore a terminated user to active status.

Path

http://localhost:8080/ps/api/v1/users/restoreterminateduser

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to restore. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.
restoreAddress	Boolean	Set to True to restore the primary email address previously assigned to the user. If desired, you can update the user's primary email address with the PUT: Update User Email Address endpoint described on page 426.
restoreAvailableAliases	Boolean	Set to True to restore the aliases previously assigned to the user.

Response Schema Fields

The response schema fields returned for the user being restored are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

PUT: Terminate User (terminateUser)

Terminate an active user.

Path

http://localhost:8080/ps/api/v1/users/terminateuser/{zlpUserId}

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to terminate. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.

Request Body Schema Fields

None.

Response Schema Fields

The response schema fields returned for the user being terminated are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

PUT: Update User Account Information (updateUserAccountInfo)

Update a user's account information.

Path

http://localhost:8080/ps/api//v1/users/updateuseraccountinfo

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to modify. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.
externalReference	String	Enter any unique IDs or information used outside of ZL UA that is relevant to the user here.
userTags	String	Specify any user tags to apply to the user.
retentionTags	String	Specify any retention tags to apply to the user.
altReviewDepartment	String	Specify the Alternative Review Department to add the user to.
fullName	String	The first and last name of the user.
dateHired	String	The date the user was hired.
miscField1 **miscField2**	String	Use these fields to enter any additional information required for the user.

Response Schema Fields

The response schema fields returned for the user being updated are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

PUT: Update User Email Address (updateUserEmailAddress)

Update a user's primary email address.

Path

http://localhost:8080/ps/api/v1/users/updateuseremailaddress/{zlpUserId}/{address}

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to modify. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.
address	String	Specify the new primary email address to assign to the user.

Request Body Schema Fields

None.

Response Schema Fields

The response schema fields returned for the user being updated are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

PUT: Update User Mail Server Information (updateUserMailServerInfo)

Update the mail server information associated with a user.

Path

http://localhost:8080/ps/api/v1/users/updateusermailserverinfo

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	String	Specify the ZL user ID of the user you want to modify. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.
mailServerName	String	Specify the mail server name for the user.
mailStoreInfo	String	Specify the mail store name for the user.
connectUserId	String	Specify the user ID required for the user to access the server.
connectPassword	String	Specify the password required for the user to access the server.

Response Schema Fields

The response schema fields returned for the user being updated are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

PUT: Update User Owner (updateUserOwnerField)

Update a user's owner. The user owner is meant to identify the user's manager or the user's creator.

Path

http://localhost:8080/ps/api/v1/users/updateuserownerfield/{zlpUserId}/{owner}

Request Parameters

Parameter	Type	Description
zlpUserId	String	Specify the ZL user ID of the user you want to update. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.
owner	String	Specify the name of the new owner.

Request Body Schema Fields

None.

Response Schema Fields

The response schema fields returned for the user being updated are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

PUT: Update User Sync Status (updateUserSyncExclude)

Update a user's user synchronization status. This determines whether the user will be modified during user synchronizations.

Path

http://localhost:8080/ps/api/v1/users/updateusersyncexclude/{zlpUserId}/{fSyncExclude}

Request Parameters

Parameter	Type	Description
zlpUserId	String	Specify the ZL user ID of the user you want to update. UAA/Users includes several endpoints you can use to retrieve user configuration data, including the ZLP user ID. For example, you could use the *GET: Get All Department Users (getAllDepartmentUsers)* endpoint to retrieve user configuration data for all the users included in a specific department.
fSyncExclude	Boolean	Set to True to exclude the user from being updated during user LDAP user synchronizations.

Request Body Schema Fields

None.

Response Schema Fields

The response schema fields returned for the user being updated are the same as those included in the response returned after creating a user. This includes the user's basic account information, as well as additional information such as the ZLP user ID assigned to the user. For descriptions of these fields, refer to *POST: Create User* on page 409.

Workspace/Recategorization

This section describes endpoints you can use to apply Record Categories and Static Tags to files. Endpoints are available to apply Record Categories and Static tags to either a data source or the files returned by a filter search:

- *POST: Recategorize Data Source (globalTags)*
- *POST: Recategorize Search View (globalTags)*

Recategorization is available for In-Place files in both lite and full workspaces. It is not available for archived files or messages.

Please note that to apply Static Tags or Record Categories to files, the **Enable Recategorization in the EA Module for InPlace Files** system registry setting in the ZL SysAdmin module must be enabled. You can check this setting's status with the GET: Get Workspace System Registry Configuration (getWorkspaceAppConfigurations) endpoint described on page 479. Contact your System Administrator or refer to the *ZL System Administrator's Guide* for instructions to follow when updating the system registry.

POST: Recategorize Data Source (globalTags)

Apply Static Tags or Record Categories to the files included in a data source. A data source is a search used to create or add data to a workspace.

Path

http://localhost:8080/ps/api/v1/ws/recategorization/workspace/{workspaceId}/datasource/{dataSourceId}/globalTags

Request Parameters

Parameter	Type	Description
workspaceId	Integer	Enter the ID of the workspace containing the data source to be updated. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation. You can set the **searchStoreType** to 53 (In-Place files) when executing the endpoint to ensure that the workspaces returned by the search include In-Place files.
dataSourceId	Integer	Specify the ID of the data source to be updated. All files included within the data source will be re-categorized. You can use the POST: Get Workspace Data Sources Using Workspace ID endpoint described on page 457 to retrieve the ID values for all data sources that have been added to a particular workspace in a single operation. Check the **datatype** field in the response scheme to ensure that the data source selected includes In-Place files (meaning that **datatype** should be set to **InplaceFiles**). Archived files and messages cannot be re-categorized.

Request Schema Fields

Schema Field	Type	Description
tagType	String	Specify **static** to apply or remove Static Tags, or **record** to apply or remove Record Categories.

Schema Field	Type	Description
tagIds	Integer	Specify the ID of the Static Tags or Record Category to be applied or removed. You can use the POST: Get Record Categories (getRecordCategories) endpoint described on page 199 to retrieve Record Category ID values and other Record Category configuration data. You can also retrieve Record Category IDs from the **FilePlan** database table. You can retrieve Static Tag IDs from the **InplaceTag** database table. Please take note of the following: • You can apply multiple Static Tags to files in a single operation. When doing so, enter the Static Tag IDs as a comma-separated list. • You can only apply a single Record Category to a file at any time. Use caution when assigning Record Categories to files, as they determine the file's retention period, which determines how long it will be retained before it is deleted from ZL UA and the source server.
cacheId	Integer	Leave empty.
action	String	Specify **tagApply** to apply the Static Tags or Record Categories to the specified files, or specify **tagRemove** to remove them from the specified files. Use the **tagType** field described previously to determine whether Static Tags or Record Categories will be applied.
mapUserParams	String	Leave empty.
fAll	Boolean	Set to True.

Response Schema Fields

The endpoint returns a string indicating whether the recategorization task was submitted successfully.

POST: Recategorize Search View (globalTags)

Apply Static Tags or Record Categories to the files returned by a search filter operation. A filter search is a search that was conducted on the contents of a workspace.

Path

http://localhost:8080/ps/api/v1/ws/recategorization/workspace/{workspaceId}/search/globalTags

Request Parameters

Parameter	Type	Description
workspaceId	Integer	Enter the ID of the workspace containing the filter search to be updated. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation. You can set the **searchStoreType** to 53 (In-Place files) when executing the endpoint to ensure that the workspaces returned by the search include In-Place files.

Request Schema Fields

Schema Field	Type	Description
tagType	String	Specify **static** to apply or remove Static Tags, or **record** to apply or remove Record Categories.
tagIds	Integer	Specify the ID of the Static Tags or Record Category to be applied or removed. You can use the POST: Get Record Categories (getRecordCategories) endpoint described on page 199 to retrieve Record Category ID values and other Record Category configuration data. You can also retrieve Record Category IDs from the **FilePlan** database table. You can retrieve Static Tag IDs from the **InplaceTag** database table. Please take note of the following: • You can apply multiple Static Tags to files in a single operation. When doing so, enter the Static Tag IDs as a comma-separated list. • You can only apply a single Record Category to a file at any time. Use caution when assigning Record Categories to files, as they determine the file's retention period, which determines how long it will be retained before it is deleted from ZL UA and the source server.
cacheId	Integer	Leave empty.

Schema Field	Type	Description
action	String	Specify **tagApply** to apply the Static Tags or Record Categories to the specified files, or specify **tagRemove** to remove them from the specified files. Use the **tagType** field described previously to determine whether Static Tags or Record Categories will be applied.
mapUserData	String	Leave empty.
viewId	String	Enter the view ID of the filter search on which re-categorization is to be performed. The view ID for a filter search is returned after you run the search with the POST: Workspace Filter Search endpoint described on page 494.
fAll	Boolean	Set to True.

Response Schema Fields

The endpoint returns a string indicating whether the recategorization task was submitted successfully.

Workspace/Roles

This section describes endpoints you can use to grant or revoke system roles to users. These roles determine which operations the users will be able to perform on workspaces within the Enterprise Analytics module:

- *PUT: Grant Roles (grantroles)*
- *PUT: Revoke User Roles (revokeroles)*

PUT: Grant Roles (grantroles)

Grant workspace system roles to a user. Each system role defines a set of privileges that allow users to perform various operations within the workspace (via the Enterprise Analytics module UI or the workspace endpoints in the REST API). If the user who logged into ZL UA while authenticating a REST API application attempts to perform a workspace operation for which they do not have the appropriate system roles, an Insufficient Privileges exception will be generated.

Path

http://localhost:8080/ps/api/v1/ws/roles/grantroles

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to grant roles to. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
userRoles: The remaining input is an array of fields defining the roles to be granted. Specify the following for each **userRole** entry:		
roleId	Integer	The role ID. You can assign the following roles with this endpoint: • **1500:** Workspace Administrator • **1501:** Workspace Reviewer • **1502:** Workspace Exporter • **1503:** Workspace Auditor • **1504:** Workspace User Administrator For descriptions of the workspace operations that these roles provide access to, refer to the *Granting System Roles* section of the *ZL UA Workspace Administrator's Guide.*
roleName	String	Not required. Leave empty.

Schema Field	Type	Description
scope	String	Specify whether the role should be granted globally so that it is applicable to all departments, or if it should be granted to specific departments only: • **Global**: Global • **InclRecur**: On selected departments recursively • **Incl**: On selected departments only
allScopeDomainIds	Integer	An array of domain IDs to specify the departments the role is applicable to for roles that are only granted on selected departments. These can be retrieved from the **ArchiveServer** database table. Specify the domain IDs as a comma-separated list. For example: `"AllScopeDomainIds": [ 0,1,2,3 ]`

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each role specified in the request.		
success	Boolean	Indicates whether the role was granted successfully (True) or not (False).
result	String	The result of the request. The string will indicate how the role has been applied (on which departments, scope, role ID, etc).
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

PUT: Revoke User Roles (revokeroles)

Revoke workspace roles that have been previously assigned to a user.

Path

http://localhost:8080/ps/api/v1/ws/roles/revokeroles

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to modify. The REST API includes several endpoints you can use to retrieve user configuration data such as the ZLP user ID. For more information, refer to *UAA/Users* on page 406.
roleIds	Integer	An array of role IDs to specify the roles to be revoked. Specify the role IDs as a comma-separated list. For example: `"RoleIds": [ 1500,1501,1502,1503 ]` You can revoke the following roles with this endpoint: • **1500:** Workspace Administrator • **1501:** Workspace Reviewer • **1502:** Workspace Exporter • **1503:** Workspace Auditor • **1504:** Workspace User Administrator For descriptions of the workspace operations that these roles provide access to, refer to the *Granting System Roles* section of the *ZL UA Workspace Administrator's Guide*.

Response Schema Fields

Schema Field	Type	Description
additionalProp: Includes the following fields for each role specified in the request.		
success	Boolean	Indicates whether the role was revoked successfully (True) or not (False).
result	String	The result of the request. If successful, a message will display indicating that the role has been revoked.
error	Array	If errors occurred, the **message** and **exception** strings provide information describing them.

Workspace/Audit

This section describes endpoints you can use to manage workspace audits. This includes endpoints to download and view audit trail data for operations that have been performed within a workspace:

- *POST: Download Workspace Audit Report (downloadAuditReport)*
- *POST: Get Workspace Audit Trails (getAudits)*
- *GET: Get Workspace Audit Action Params (getWsAuditActionParams)*

POST: Download Workspace Audit Report (downloadAuditReport)

Download an audit trail report for a workspace. The audit trail report summarizes the operations that have been performed within the workspace. You can customize the request data to create an audit trail report for all workspace operations, or for specific types of operations. You can also filter the request to include actions that occurred within a specific date range.

Path

http://localhost:8080/ps/api/v1/ws/audits/downloadauditreport

Request Parameters

None.

Request Body Schema Fields

Field	Type	Description
cacheId	Integer	If you want to download a report for a set audit trail data that you have viewed previously with the POST: Get Workspace Audit Trails (getAudits) endpoint described on page 443, enter the cache ID returned by the endpoint here. This is the endpoint you would use to view the audit trail data. When you specify a cache ID, you still must specify the remaining request body schema fields to generate a report for a new set of audit trail data. **Make sure these values match those that were originally used to generate the specified cache ID**. To generate a report for a new set of audit trail data, you can leave this field empty and specify the remaining request body schema fields as desired. The response time may be slightly longer when generating audit trail data without specifying a cache ID.
actionCode	String	Specify an action code to identify the action sub-type to be included in the report. This is used in conjunction with the **actionType** field to allow you to generate audit trail data for specific sets of operations. For example, if **actionType** is set to **workspace**, you could specify an action code to view audit trail data for workspace creations or workspace deletions. Use the GET: Get Audit Action Params endpoint described on page 446 to view the valid action codes for each action type. You can also enter **-1** as the action code to view all operations for the selected action type.

Field	Type	Description
actionType	String	Specify the action type for the report: • **any:** All Enterprise Analytics operations. • **workspace:** Workspace management operations (e.g.. workspace created, workspace edited, workspace deleted). • **tag:** Tag operations (e.g., tag created, tag deleted). • **audit:** Audit operations (e.g., audit trail viewed). • **preservation:** Preservation operations (e.g., preservation viewed). • **item:** Operations affecting files and messages (e.g., tag applied or tag removed). • **search:** Search operations (e.g., saved search viewed). • **recategorization:** Recategorization actions (i.e., applying Record Categories). • **export:** Export operations (e.g., search exported). You can specify an action sub-type with the **actionCode** parameter to view a more specific set of operations.
dateStart **dateEnd**	String	These fields are applicable when the **mode** field is set to **custom,** meaning that the report will include actions performed within a specific date range. Use them to specify the start date and end date (inclusive) for the report.
dateMode	String	Set this to any of the following values to determine the date range for the report: • **1day**: Actions performed within the last day. • **1week**: Actions performed within the last week. • **1month**: Actions performed within the last 30 days. • **custom**: Actions performed within a specific date range (i.e., the range specified by the **dateStart** and **dateEnd** parameters).
workspaceId	Integer	Specify the ID of the workspace for which you want to view the audit trail data. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.

Response Schema Fields

The response includes a link you can use to download the audit trail report.

POST: Get Workspace Audit Trails (getAudits)

View audit trail data for a workspace. You can customize the request data to create an audit trail report for all Enterprise Analytics operations, for specific types of operations, or for operations that occurred within a specific date range.

Path

http://localhost:8080/ps/api/v1/ws/audits/getaudits

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
workspaceId	Integer	Specify the ID of the workspace for which you want to view the audit trail data. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.
dateEnd **dateMode** **dateStart**	String	Set the **dateMode** field to any of the following values to determine the date range for the report: • **1day**: Actions performed within the last day. • **1week**: Actions performed within the last week. • **1month**: Actions performed within the last 30 days. • **custom**: Actions performed within a specific date range (i.e., the range specified by the **dateStart** and **dateEnd** parameters). If you set **dateMode** to **custom,** use the **dateStart** and **dateEnd** fields to specify the start date and end dates (inclusive) for the report. Otherwise, leave these fields blank.
start **itemsPerPage**	Integer	Some operations return too many values for the UI to display. You can use the **start** parameter to indicate the page number on which these values should be displayed, and the **itemsPerPage** parameter to indicate the maximum number of items that can be displayed on the page.

Parameter	Type	Description
actionCode	String	Specify an action code to identify the action sub-type to be included in the report. This is used in conjunction with the **actionType** field so you can generate audit trail data for specific sets of operations. For example, if **actionType** is set to **workspace**, you could specify an action code to view audit trail data for workspace creations or workspace deletions. Use the GET: Get Audit Action Params endpoint described on page 446 to view the valid action codes for each action type. You can also enter **-1** as the action code to view all operations for the selected action type.
actionType	String	Specify the action type for the report: • **any:** All Enterprise Analytics operations. • **workspace:** Workspace management operations (e.g.. workspace created, workspace edited, workspace deleted). • **tag:** Tag operations (e.g., tag created, tag deleted). • **audit:** Audit operations (e.g., audit trail viewed). • **preservation:** Preservation operations (e.g., preservation viewed). • **item:** Operations affecting files and messages (e.g., tag applied or tag removed). • **search:** Search operations (e.g., saved search viewed). • **recategorization:** Recategorization actions (i.e., applying Record Categories). • **export:** Export operations (e.g., search exported). You can specify an action sub-type with the **actionCode** parameter to view a more specific set of operations.

Response Schema Fields

The response includes an entry for each action taken upon the workspace that matches the input specified in the request schema. Each entry includes the following information.

Parameter	Type	Description
total	Integer	The total number of operations that match the specified input.
cacheId	String	The cache ID assigned to the audit trail data. You can provide this as an input to the POST: Download Workspace Audit Report (downloadAuditReport) endpoint described on page 441 to download a report for the audit trail data returned here.
totalRecords **totalDisplayRecords**	Integer	The number of audit trail entries included and displayed in the results.
rows	Integer	The number of rows included in the results.
auditTrails	Array	The **auditTrails** array includes an entry for each operation that matches the request input. Each entry includes the following fields:
workspaceName	String	The name of the workspace upon which the action was taken.
workspaceId	Integer	The ID of the workspace upon which the action was taken.
action	String	The action taken upon the workspace.
date	String	The date and time the action was performed.
username	String	The user who performed the action.
itemId	Integer	The ID of the object upon which the action was taken. The object could be a file, a message, a workspace, a search, or a tag, depending on which action types you are viewing.
sourceIp	String	The IP address of the machine from which the user performed the action.
destIp	String	The IP address of the ZL application server running the Enterprise Analytics module.
comments	String	Additional information about the action. This varies depending on the action. For example, an action affecting a search would include the name of the search here.

GET: Get Workspace Audit Action Params (getWsAuditActionParams)

Retrieve action codes for the Enterprise Analytics module. You will use these codes to specify the **actionCode** parameter when using the POST: Get Workspace Audit Trails and POST: Download Workspace Audit Trails endpoints to view audit trail data or download an audit trail report. This will allow you to generate audit trail data for specific sets of operations.

Path

http://localhost:8080/ps/api/v1/ws/audits/getwsauditactionparams/{auditType}

Request Parameters

Parameter	Type	Description
auditType	String	Specify the type of action you want to view the action codes for: • **any:** All workspace operations. • **workspace:** Workspace management operations (e.g.. workspace created, workspace edited, workspace deleted). • **tag:** Tag operations (e.g., tag created, tag deleted). • **audit:** Audit operations (e.g., audit trail viewed). • **preservation:** Preservation operations (e.g., preservation viewed). • **item:** Operations affecting files and messages (e.g., tag applied or tag removed). • **search:** Search operations (e.g., saved search viewed). • **recategorization:** Recategorization actions (i.e., applying Record Categories). • **export:** Export operations (e.g., search exported).

Request Body Schema Fields

None.

Response Schema Fields

Parameter	Type	Description
actionDescription	String	A description of the action sub-type.
actionCode	Integer	The action code for the sub-type. Use this to specify the **actionCode** parameter when using the POST: Download Workspace Audit Trails and POST: Get Workspace Audit Trails endpoints to generate audit trail data.

Workspace/Export

This section describes endpoints you can use to export workspace data:

- *POST: Export Workspace Data Source (dataSource)*
- *POST: Export Workspace View (view)*

POST: Export Workspace Data Source (dataSource)

Export a data source from a workspace. The export file will be generated in the directory specified by the **Default Export Location** or **Enterprise Analytics Export Directory** setting in the ZL UA SysAdmin registry. For more information on these registry settings, contact your System Administrator or refer to the *ZL UA Workspace Administrator's Guide.*

This endpoint supports the **top export** feature. When you use the top export feature, you will specify how many items to include in the export and a sort order. The items to include in the export will be selected based on these settings.

For example, you could sort the items within the data source by relevance in descending order. If you chose to export the top 200 items with this sorting in place, the export would include the 200 items that are assigned the highest relevance scores. Alternatively, you could reverse the default sorting so that the items with the lowest relevance are listed first. In this case, the export would include the 200 items that are assigned the lowest relevance scores. The same method will be used to select the items if you sort the search results by other columns.

Path

http://localhost:8080/ps/api/v1/ws/export/exports/datasource

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
workspaceId	Integer	The ID of the workspace containing the data source to be exported. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.
exportFileName	String	Specify the name of the export file.

Schema Field	Type	Description
sortField	String	If you plan to use the top export feature, specify the field by which the export data should be sorted. This is used to determine which items will be included in the export. For files, use any of the following options: • **lastModified:** Sort by item last modified date. • **createDate:** Sort by item creation date. • **lastAccessed:** Sort by item last accessed date. • **relevancy:** Sort by item relevancy, a rating of how closely each file matched the search parameters. • **size:** Sort by item size. • **sort_dummy:** Do not sort the data. For messages, use any of the following options: • **date:** Sort by date sent. • **relevancy:** Sort by item relevancy, a rating of how closely each file matched the search parameters. • **size:** Sort by item size. • **sort_dummy:** Do not sort the data. Leave empty if you do not want to use the top export feature. **Please note that the sort options for this endpoint are not supported for full workspaces in ZL UA 11.1.**
ascending	Boolean	If you plan to use the top export feature, set to True to sort the export data in ascending order by **sortField**, or False to sort the export data in descending order by **sortField**. This is used to determine which items will be included in the export. Leave empty if you do not want to use the top export feature.

Schema Field	Type	Description
exportCount	Integer	If you plan to use the top export feature, specify how many items to include in the export, or leave blank to export all items that are included in the workspace. If you specify an **exportCount**, the items that the export will include will be selected based on how the workspace data is sorted. This is determined by the **sortField** and **ascending** fields described previously. For example, if you set **exportCount** to 200, **sortField** to **relevancy** and **ascending** to False, the export would include the 200 messages that are assigned the highest relevance. Leave empty if you do not want to use the top export feature.
removeBasicHeaders	Boolean	Set to True to remove items with duplicate basic headers from the export.
removeEnvHeaders	Boolean	Set to True to remove items with duplicate content envelope headers from the export.
notifyUser	Boolean	Set to True to send an e-mail notifications when the export completes. You can configure the email addresses to which notifications will be sent in the ZL system registry. For more information, contact your System Administrator or refer to the *ZL UA Workspace Administrator's Guide.*
exportSelection	String	Specify the export file format (i.e., **json** or **apf**.
exportFields	String	Enter a comma-separated list of fields to include in the export file. Leave empty to include all fields. For files, you can include the following fields, regardless of export type: **size**, **name**, **createDate**, **lastAccess**, **lastModified**. For mails, you can include the following for JSON exports: **zlid, body, aliases, tags, attachment** For mails, you include the following for APF exports: **messageID, date, size, type, from, to, cc, bcc, subject, body, aliases, tags**

Schema Field	Type	Description
datasourceId	String	Specify the ID of the data source you want to export. You can use the POST: Get Workspace Data Sources Using Workspace ID endpoint described on page 457 to retrieve the ID values for all data sources that have been added to a particular workspace in a single operation.

Response Schema Fields

The response schema includes the task ID assigned to the export.

POST: Export Workspace View (view)

Export a workspace view. A workspace view is the search result set generated when you run a filter search or a Data Set Manipulation on a workspace.

The file will be generated in the directory specified by the **Default Export Location** or **Enterprise Analytics Export Directory** setting in the ZL UA SysAdmin registry. For more information on this registry setting, contact your System Administrator or refer to the *ZL Enterprise Analytics Administrator's Guide.*

This endpoint provides access to the **top export** feature. When you use the top export feature, you will specify how many items to include in the export and a sort order. The items to include in the export will be selected based on these values.

For example, you could sort the items within the search result set by relevance in descending order. If you chose to export the top 200 items with this sorting in place, the export would include the 200 items that are assigned the highest relevance scores. Alternatively, you could reverse the default sorting so that the items with the lowest relevance are listed first. In this case, the export would include the 200 items that are assigned the lowest relevance scores. The same method will be used to select the items if you sort the search results by other columns.

Path

http://localhost:8080/ps/api/v1/ws/export/exports/view

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
workspaceId	Integer	The ID of the workspace containing the search or data source to be exported. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.
exportFileName	String	Specify the name of the export file.

Schema Field	Type	Description
sortField	String	If you plan to use the top export feature, specify which field the export data should be sorted by. This is used to determine which items will be included in the export: • **lastModified:** Sort by item last modified date. • **createDate:** Sort by item creation date. • **relevancy:** Sort by item relevancy, a rating of how closely each file matched the search parameters. • **size:** Sort by item size. Leave empty if you do not want to use the top export feature.
ascending	Boolean	If you plan to use the top export feature, set to True to sort the export data in ascending order by **sortField**, or False to sort the export data in descending order by **sortField**. This is used to determine which items will be included in the export. Leave empty if you do not want to use the top export feature.
exportCount	Integer	If you plan to use the top export feature, specify how many items to include in the export. If you specify an **exportCount**, the items that the export will include will be selected based on how the workspace data is sorted. This is determined by the **sortField** and **ascending** fields described previously. For example, if you set **exportCount** to 200, **sortField** to **relevancy** and **ascending** to False, the export would include the 200 messages that are assigned the highest relevance. Leave empty if you do not want to use the top export feature.
removeBasicHeaders	Boolean	Set to True to remove items with duplicate basic headers from the export.
removeEnvHeaders	Boolean	Set to True to remove items with duplicate content envelope headers from the export.
notifyUser	Boolean	Set to True to send an e-mail notifications when the export completes. You can configure the email addresses to which notifications will be sent in the ZL system registry. For more information, contact your System Administrator or refer to the *ZL UA Workspace Administrator's Guide.*

Schema Field	Type	Description
exportSelection	String	Specify the export file format (i.e., **json** or **csv**).
exportFields	String	Enter a comma-separated list of fields to include in the export. Leave empty to include all fields.
viewId	String	Specify the view ID of the filter search or Data Set Manipulation you want to export from the workspace. View IDs are returned after you perform a search on a workspace with the POST: Workspace Dataset Manipulation Search and POST: Workspace Filter Search endpoints. For more information, refer to *Workspace/Search* on page 490.

Response Schema Fields

The response schema includes the task ID assigned to the export.

Workspace/Workspace

This section describes endpoints you can use to manage workspaces in the Enterprise Analytics module. This includes endpoints to create workspaces, add data sources to workspaces, view workspace status information and retrieve workspace configuration information:

- *POST: Get All Data Sources From a Workspace (datasources)*
- *POST: Create Data Sources (createDataSources)*
- *GET: Get All Workspaces (getWorkspaces)*
- *POST: Create Workspace (createWorkspace)*
- *POST: Download Summarization Report for Document (summaryDownload)*
- *GET: Get All Background Task Statuses (getAllBackgroundTaskStatus)*
- *POST: Run All Background Tasks (runAllBackgroundTasks)*
- *GET: Retrieve Data Source Items (getAllItemsInDataSource)*
- *GET: Get Specific Background Task Status (getBackgroundTaskStatus)*
- *GET: Get GSR Task Status (gsrStatus)* ***(Internal Use Only)***
- *GET: Get Workspace System Registry Configuration (getWorkspaceAppConfigurations)*
- *GET: Get Combined Task Status (workspaceStatus)*
- *POST: Get Workspace Using ID (getWorkspaceUsingId)*
- *PUT: Update Workspace (updateWorkspaceNameAndDescription)*
- *POST: Run Summarization on Document (summary)*

POST: Get All Data Sources From a Workspace (datasources)

Retrieve a list of the data sources that have been added to a workspace. Each data source is a search that was used to build and add data to the workspace.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/{workspaceId}/datasources

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	
workspaceId	String	The ID of the workspace whose configuration is to be retrieved. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.

Request Body Schema Fields

None.

Response Schema Fields

The response schema includes an entry for each data source included in the specified workspace.

Schema Field	Type	Description
dataSourceId	Integer	The data source ID.
workspaceId	Integer	The workspace ID.
searchName	String	The name of the search conducted to build the data source.
createDate **updateDate** **lastExportDate**	String	The date and time the data source was created, last updated and last exported.
totalRecords	Integer	The number of items included in the data source.

Schema Field	Type	Description
purpose	String	The purpose of the data source, as specified when it was created with the POST: Create Data Source endpoint.
rawQuery	String	The raw query represents the search that was used to build the data source.
datatype	String	The type of data source: • **JournalMails:** Journaled mails • **ArchiveMails:** Archived mails • **InplaceMails:** In-Place mails • **ArchiveFiles:** Archived files • **InplaceFiles:** In-Place files • **DeletedArchiveFiles:** Archived files from deleted search store • **DeletedInplaceFiles:** In-Place files from deleted search store

POST: Create Data Sources (createDataSources)

Create a data source (or group of data sources) and add it to a workspace.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/{workspaceId}/datasources

Request Parameters

Parameter	Type	Description
workspaceId	Integer	The ID of the workspace to add the data source to. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.

Request Body Schema Fields

Parameter	Type	Description
datasources: The **datasources** array includes an entry for each data source you plan to add to the workspace. The fields you should include in each entry are described below.		
searchType	Integer	A value identifying the type of search store that will be used to build the data source: • **1: Journaled Mails** • **2: Archived Mails** • **9: In-Place Mails** • **50: Archived Files** • **53: In-Place Files** • **55: Deleted Archive Files** • **56: Deleted In-Place Files**
dataSourceName	String	Enter the name to assign to the data source.

Parameter	Type	Description
viewId	Integer	The view ID of the search result set to add to the data source. The view IDs you can specify here are returned after you perform Global Searches on the files and messages included in your ZL UA installation. For more information on the endpoints you can use to perform Global Searches, refer to *Global Search* on page 290. **Please note that the Global Search REST API includes separate endpoints for searches for archived and In-Place messages and files. In addition, the endpoints used to search messages include inputs you can use to specify whether to search for journaled mails or archived mails. Please make sure to specify a view ID for a Global Search that is of the same data type as that specified for the searchType parameter described earlier in this table.**
purpose	String	A description of the data source's purpose.
searchStoreId	Integer	The search store ID assigned to the store used to create the data source. You can retrieve search store IDs from the **ssid** column of the **SearchStore** database table.

Response Schema Fields

The response scheme include the following fields for each data source specified in the request schema.

Schema Field	Type	Description
dataSourceId	Integer	The data source ID.
workspaceId	Integer	The workspace ID.
searchName	String	The name of the search conducted to build the data source.
createDate **updateDate** **lastExportDate**	String	The date and time the data source was created, last updated and last exported.
totalRecords	Integer	The number of items included in the data source.
purpose	String	The purpose of the data source, as specified when it was created with the POST: Create Data Source endpoint described on page 459.

Schema Field	Type	Description
rawQuery	String	The raw query represents the search that was used to build the data source.
searchStoreId	Integer	The ID of the search store used to build the data source.

GET: Get All Workspaces (getWorkspaces)

Retrieve the configurations of the workspaces that have been created with a specific search store.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	
searchStoreType	Integer	A value identifying the search store to use to build the workspaces to be viewed: • **-1** or **Empty: All Search Stores** • **1: Journaled Mails** • **2: Archived Mails** • **9: In-Place Mails** • **50: Archived Files** • **53: In-Place Files** • **55: Deleted Archive Files** • **56: Deleted In-Place Files**
workspaceMode	String	Enter **lite** to return Lite Workspace configurations, or **full** to return Full Workspace configurations. Leave empty to return all workspace configurations.
workspaceType	String	Enter **mail** or **file.**

Request Body Schema Fields

None.

Response Schema Fields

The response schema includes an entry for each workspace that has been added with the specified search store. Each entry includes the following fields.

Schema Field	Type	Description
workspaceName	String	The name of the workspace.
workspaceDescription	String	A description of the workspace.
workspaceType	String	The workspace type (**mail** or **file**).
workspaceMode	String	The workspace mode: **lite** for a Lite Workspace, or **full** for a Full Workspace.
workspaceSearchType	Integer	A value identifying the search store to use to build the workspace: • **1: Journaled Mails** • **2: Archived Mails** • **9: In-Place Mails** • **50: Archived Files** • **53: In-Place Files** • **55: Deleted Archive Files** • **56: Deleted In-Place Files**
workspaceId	Integer	The ID assigned to the workspace.
createDate **lastAccessedDate**	String	The dates the workspace was created and last accessed.
itemCount	Integer	The number of items included in the workspace.
status	String	A summary of the workspace status.

POST: Create Workspace (createWorkspace)

Create a new workspace.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces

Request Parameters

None.

Request Schema Fields

Schema Field	Type	Description
workspaceName	String	The name of the workspace.
workspaceDescription	String	A description of the workspace.
workspaceType	String	The data type for the workspace. Specify **mail** or **file**.
workspaceMode	String	The workspace mode. Specify **lite** or **full**: • **Lite:** Lite workspaces can be created quickly. You should use Lite Workspaces when you want to quickly perform a search and create a workspace to view the data. • **Full:** Full workspaces include advanced filtering and data wrangling capabilities. Please note that to create a full workspace, the **Enable Full Workspace in the EA Module** system registry setting must be enabled. You can check this setting with the GET: Get Workspace System Registry Configuration (getWorkspaceAppConfigurations) described on page 479.

Schema Field	Type	Description
workspaceSearchType	Integer	A value identifying the search store to use to build the workspace: • **1: Journaled Mails** • **2: Archived Mails** • **9: In-Place Mails** • **50: Archived Files** • **53: In-Place Files** • **55: Deleted Archive Files** • **56: Deleted In-Place Files**

Response Schema Fields

Schema Field	Type	Description
workspaceName	String	The name assigned to the workspace.
workspaceDescription	String	The description of the workspace.
workspaceType	String	The workspace type (**mail** or **file**).
workspaceMode	String	The workspace mode: **lite** or **full**.
workspaceSearchType	Integer	A value identifying the search store to use to build the workspace: • **1: Journaled Mails** • **2: Archived Mails** • **9: In-Place Mails** • **50: Archived Files** • **53: In-Place Files** • **55: Deleted Archive Files** • **56: Deleted In-Place Files**
workspaceId	Integer	The ID value assigned to the workspace.

Schema Field	Type	Description
createDate **lastAccessedDate**	String	The dates that the workspace was created, and last accessed.
itemCount	Integer	The number of items included in the workspace.
status	String	The status of the workspace.

POST: Download Summarization Report for Document (summaryDownload)

Download the summarization report for a document. This includes an AI-generated summary of the document's contents. Use of the summarization feature has the following requirements:

- The **Enable Summarization Using AI for Inplace Files in Full Workspace** system registry setting must be enabled. You can check this status with the GET: Get Workspace System Registry Configuration endpoint described on page 479.
- The ZL Summary Service must be installed and started on your system. This requires additional system configuration. For further details, contact your System Administrator or refer to the *Enabling the ZL Summary Service* section of the *ZL UA Workspace Administrator's Guide.*
- The summarization feature is only available for files in full workspaces.

Please note that as of the release of ZL UA 11.1.2, the summarization feature is a Beta Level feature. It is not intended for use in production environments. Instructions are provided if you would like to try it in a demo or test system.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/{workspaceId}/item/{documentId}/summary/download

Request Parameters

Parameter	Type	Description
workspaceId	Integer	The ID of the full workspace containing the file to be summarized. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation. Make sure to check the **workspaceMode** and **workspaceType** fields in the endpoint's response data to make sure that you specify an ID for a full workspace for files. The summarization feature is not supported for lite workspaces or for mail documents.
documentId	String	The ID assigned to the file to be summarized. You can use the GET: Retrieve Data Source Items endpoint described on page 472 to retrieve a list of the items that have been added to each data source within a workspace. Use the **refItemId** response parameter as the document ID. Make sure that the **type** response parameter for the file is set to 12 (In-Place file). The summarization feature is not supported for archived files or mail documents.

Request Schema Fields

None.

Response Schema Fields

The response schema includes a link you can use to download the summarization report.

GET: Get All Background Task Statuses (getAllBackgroundTaskStatus)

Retrieve the statuses of background tasks that have been run on a workspace.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/{workspaceId}/status

Request Parameters

None.

Request Schema Fields

Schema Field	Type	Description
workspaceId	Integer	The ID of the workspace whose background task statuses are to be retrieved. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.

Response Schema Fields

The response schema includes the following fields for each background task that has been executed upon the workspace.

Schema Field	Type	Description
key	String	The task key assigned to the task.
taskName	String	The task name.
startTime **endTime**	Integer	The time that the task started and ended. These times are delivered as epoch timestamps. For more information on epoch timestamps, refer to: https://www.epochconverter.com/
elapsedTime	Integer	The amount of time the task has taken, in milliseconds.
taskSuccess	Boolean	If True, it indicates that the task has completed successfully.
taskMessage	String	Additional information about the task.
taskErrorMessage	String	Error information for the task (if applicable).
status	Integer	A numeric code identifying the task status.
totalPhases **currentPhase**	Integer	The total number of phases included in the task, and the phase that the task is currently performing.

Schema Field	Type	Description
statusMsg	String	Additional status information about the task.
counters	Array	The **counters** array includes the following information regarding the items affected by the task. It may have one or more entries, depending on which task is performed.
key	String	A description of the counter. For example, the Workspace Preservation task includes an **Items Found** key which indicates how many items were found by the task.
value	Integer	The value associated with the task key.
phases	Array	The **phases** array includes an entry for each phase included within the task. Each entry includes the following information:
phase	String	The phase ID represents the index assigned to the phase, corresponding to the start and end timestamps of the phase.
status	String	The phase status includes information such as a timestamp indicating when the task started, and the overall status of the phase.
props	Array	Each **prop** entry is map of key value pairs corresponding to the name and value of the corresponding properties of a background task.
isInProgess	Boolean	If True, it indicates that the task is in progress.
isDead	Boolean	It True, it indicates that the task has stopped.
known	Boolean	If True, it indicates that it is a known task.
pid	Integer	The process ID assigned to the task.

POST: Run All Background Tasks (runAllBackgroundTasks)

Run all background tasks on a workspace. This includes the **ItemImporter**, **IndexOfIndex** and **WorkspaceMapper** background tasks. You should execute this endpoint after creating a full workspace or adding data to it, as these background tasks are used to import the data into the workspace and index it.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/{workspaceId}/status

Request Parameters

None.

Request Schema Fields

Schema Field	Type	Description
workspaceId	Integer	The ID of the workspace to be updated. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.

Response Schema Fields

The response schema includes the following information for each background task.

Schema Field	Type	Description
additionalProp: Includes the following fields:		
success	Boolean	Indicates whether the background task was completed successfully (True) or not (False).
result	String	A string describing the result of the background task.
error	If errors occurred, the **message** and **exception** strings provide information describing them.	

GET: Retrieve Data Source Items (getAllItemsInDataSource)

Retrieve a list of the items (e.g., files or messages) that were added to a data source within a particular workspace.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/{workspaceId}/datasources/{sourceId}

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	
sort	String	Specify which field the response data should be sorted by. For files, you can choose from the following options: **size** **name** **createDate** **lastModified** For mails, you can choose from the following options: **sender** **to** **subject** **date** **size** Leave empty if you do not want the data sorted.
ascending	Boolean	Set to True to sort the export data in ascending order by **sort**, or False to sort the export data in descending order by **sort**. This is used to determine which items will be included in the response data.

Parameter	Type	Description
updateItemCount	Boolean	For full workspaces, this parameter is not required. For lite workspaces, set to True to update the item count assigned to the data source. The item count may change as items are added and removed from the search index. ZL recommends setting this to True the first time you access a data source with this endpoint, and False for subsequent requests.
workspaceId	Integer	The ID of the workspace containing the data source whose configuration is to be retrieved. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.
sourceId	Integer	The ID of the data source you want to view. You can use the POST: Get Workspace Data Sources Using Workspace ID endpoint described on page 457 to retrieve the ID values for all data sources that have been added to a particular workspace in a single operation.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
pageNumber	Integer	The current page.
pageSize	Integer	The page size specified for the operation.
viewId	Integer	The view ID assigned to the operation's result set.
dataSource	Array	The **dataSource** array include fields describing the data source being viewed. It includes the following fields:
dataSourceId	Integer	The data source ID.
workspaceId	Integer	The workspace ID.
searchName	String	The name assigned to the search used to build the data source.
createDate **updateDate** **lastExportDate**	String	The dates and times the data source was created, last updated, and last exported.

Schema Field	Type	Description
totalRecords	Integer	The number of items included in the data source.
purpose	String	The purpose of the data source.
rawQuery	String	The raw query representing the search used to build the data source.
datatype	String	The type of data source: • **JournalMails:** Journaled mails • **ArchiveMails:** Archived mails • **InplaceMails:** In-Place mails • **ArchiveFiles:** Archived files • **InplaceFiles:** In-Place files • **DeletedArchiveFiles:** Archived files from deleted search store • **DeletedInplaceFiles:** In-Place files from deleted search store
items	Array	The **items** array includes an entry for each file or message included within the workspace specified in the input. Each entry includes the following fields:
rowID	Integer	Internal use only.
viewItemId	Integer	The view ID assigned to the item.
refItemId	Integer	The reference ID assigned to the item.
size	Integer	The size of the item, in KB or MB
relevancy	Integer	The relevancy rating assigned to the item. The relevancy rating is an indication of how well the item matched the search criteria used to build the data source. A low relevancy rating indicates the message matched a greater part of the search criteria (e.g., items that are assigned a relevancy rating of 100.0 matched the most search criteria). **Please note that relevancy scores are a Beta Level feature as of the release of ZL UA 11.1.2.**

Schema Field	Type	Description
type	Integer	The type of the file or message: • **1: Mails** • **2: Archived files** • **12: In-place files**
deletedSearchItem	Boolean	If True, it indicates that the item was retrieved from a search store for deleted items.
totalItems	Integer	The number of items included in the data source.

GET: Get Specific Background Task Status (getBackgroundTaskStatus)

Retrieve the status of a background task that has been run on a workspace.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/{workspaceId}/status/{taskName}

Request Parameters

Parameter	Type	Description
workspaceId	Integer	The ID of the workspace to be viewed. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.
taskName	String	The name of the background task for which the status is to be retrieved. Enter any of the following values: **globalSearch** **wsItemImport** **uContextIndex** **workspaceMapperTask** The endpoint will return status information for the most recent execution of the specified background task.

Request Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
key	String	The task key assigned to the task.
taskName	String	The task name.
startTime **endTime**	Integer	The time that the task started and ended. These times are delivered as epoch timestamps. For more information on epoch timestamps, refer to: https://www.epochconverter.com/
elapsedTime	Integer	The amount of time the task has taken, in milliseconds.
taskSuccess	Boolean	Set to True if the task has completed successfully.

Schema Field	Type	Description
taskMessage	String	Additional information about the task.
taskErrorMessage	String	Error information for the task (if applicable).
status	String	The task status.
totalPhases **currentPhase**	Integer	The total number of phases included in the task, and the phase that the task is currently performing.
statusMsg	String	Additional status information about the task.
counters	Array	The **counters** array includes the following information regarding the items affected by the task. It may have one or more entries, depending on which task is performed:
key	String	A description of the counter. For example, the Workspace Preservation task includes an **Items Found** key which indicates how many items were found by the task.
value	Integer	The value associated with the task key.
phases	Array	The **phases** array includes an entry for each phase included within the task. Each entry includes the following information:
phase	String	The phase ID represents the index assigned to the phase, corresponding to the start and end timestamps of the phase.
status	String	The phase status includes information such as a timestamp indicating when the task started, and the overall status of the phase.
props	Array	Each **prop** entry is map of key value pairs corresponding to the name and value of the corresponding properties of a background task.
isInProgess	Boolean	If True, it indicates that the task is in progress.
isDead	Boolean	It True, it indicates that the task has stopped.
known	Boolean	If True, it indicates that it is a known task.
pid	Integer	The process ID assigned to the task.

GET: Get GSR Task Status (gsrStatus)

As of the release of ZL UA 11.1.2, this endpoint is reserved for internal use.

GET: Get Workspace System Registry Configuration (getWorkspaceAppConfigurations)

Retrieve the configuration of system registry settings that affect workspace features and operations. For instructions to follow when configuring the registry settings described here, refer to the *Enabling and Configuring the Enterprise Analytics Module* section of the *ZL UA Workspace Administrator's Guide* or the *System Configuration* chapter of the *ZL System Administrator's Guide.*

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/config

Request Parameters

None.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
jsonExportOn	Boolean	If True, it indicates that JSON exports are enabled. This is determined by the **Enable JSON File Format Export** registry setting.
apfExportOn	Boolean	If True, it indicates that APF exports are enabled. This is determined by the **Enable Apache Parquet File Format Export** registry setting.
ocrEnabled	Boolean	If True, it indicates that OCR conversion is enabled. This is determined by the **Enable OCR for Attachments** registry setting.
preservationEnabled	Boolean	If True, it indicates that the Preservation Workflow is enabled. This is determined by the **Enable Preservations in the EA Module for Lite Workspace** registry setting.
efmEnabled	Boolean	If True, it indicates that the Enterprise Files Management module is enabled. This is determined by the **ZL Enterprise Files Management** system registry setting.

Schema Field	Type	Description
zlSynonymSearchEnabled	Boolean	If True, it indicates that the ZL Synonyms Service is enabled. When this feature is enabled, you can optionally click **See Similar Keywords** after specifying some search filter settings to view similar keywords that may also be suitable for your search. This is determined by the **Enable ZL Synonyms Search in the EA Module** registry setting.
zlSynonymsUrl	String	If the ZL synonyms search feature is enabled, this is the URL of the ZL server that is running the ZL Synonyms Service. This is determined by the **ZL Synonyms URL** registry setting.
fullWorkspaceEnabled	Boolean	If True, it indicates the Full Workspaces are enabled. This is determined by the **Enable Full Workspace in the EA Module** registry setting.
recategorizationEnabled	Boolean	If True, it indicates that recategorization of files via the application of Record Categories and Static Tags is enabled. This is determined by the **Enable Recategorization in the EA Module for InPlace Files** registry setting.
deletedSearchStoreFilesEnabled	Boolean	If True, it indicates that search stores for deleted files are enabled. This is determined by the **Enable Deleted Search Store in EA module for Workspace (Files)** registry setting.
summarizationEnabled	Boolean	If True, it indicates that the **Enable Summarization Using AI for Inplace Files in Full Workspace** registry setting is enabled. When this is enabled, you can use the POST: Run Summarization on Document endpoint describe on page 488 to run summarization on a document. The endpoint will return an AI-generated summary of the document's contents. **As of the release of ZL UA 11.1.2, summarization is a Beta Level feature.**
tabulationEnabled	Boolean	If True, it indicates that the **Enable Tabulation Feature in EA Module for Files** registry setting is enabled. When this setting is enabled, you can use the Workspace/Stats endpoints described on page 502 to view tabulation data for your workspaces. **As of the release of ZL UA 11.1.2, tabulation is a Beta Level feature.**

Schema Field	Type	Description
workspacePrivileges	Array	Lists the privileges that have been granted to the user who logged into ZL UA when authenticating the REST application. A True value indicates that the permission has been granted to the user, and a False value indicates that it has not. These privileges are determined by the roles that have been granted to the user. For information on granting or revoking roles that affect workspace operations, refer to *Workspace/Roles* on page 435.

GET: Get Combined Task Status (workspaceStatus)

Retrieve the status of a workspace.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/{workspaceId}/workspaceStatus

Request Parameters

Parameter	Type	Description
workspaceId	String	The ID of the workspace whose task status is to be retrieved. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
workspaceName	String	The name of the workspace.
workspaceDescription	String	The description entered when the workspace was created.
workspaceType	String	The workspace type (**mail** or **file**).
workspaceMode	String	Indicates whether the workspace is a full workspace or a lite workspace.
workspaceSearchType	Integer	A value identifying the search store to use to build the workspace: • **1: Journaled Mails** • **2: Archived Mails** • **9: In-Place Mails** • **50: Archived Files** • **53: In-Place Files** • **55: Deleted Archive Files** • **56: Deleted In-Place Files**

Schema Field	Type	Description
workspaceId	Integer	The workspace ID.
createDate **lastAccessedDate**	String	The date and time the workspace was created and last accessed.
itemCount	Integer	The number of items included in the workspace.
status	String	The status of the workspace (e.g., **Done**, **Failed, Unknown, In Progress** or **N/A**). **Unknown** indicates that an error occurred while reading the status. **In Progress** indicates that the workspace is being created, or a data source is being added to it. **N/A** is returned for al lite workspaces, as lite workspaces are created instantaneously.

POST: Get Workspace Using ID (getWorkspaceUsingId)

Retrieve the configuration of a specific workspace.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/{workspaceId}/workspaceStatus

Request Parameters

Parameter	Type	Description
workspaceId	String	The ID of the workspace whose configuration is to be retrieved. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.

Request Body Schema Fields

None.

Response Schema Fields

Schema Field	Type	Description
workspaceName	String	The name of the workspace.
workspaceDescription	String	The description entered when the workspace was created.
workspaceType	String	The workspace type (**mail** or **file**).
workspaceMode	String	Indicates whether the workspace is a full workspace or a lite workspace.
workspaceSearchType	Integer	A value identifying the search store to use to build the workspace: The following values may be returned: • **1: Journaled Mails** • **2: Archived Mails** • **9: In-Place Mails** • **50: Archived Files** • **53: In-Place Files** • **55: Deleted Archive Files** • **56: Deleted In-Place Files**

Schema Field	Type	Description
workspaceId	Integer	The workspace ID.
createDate **lastAccessedDate**	String	The date and time the workspace was created and last accessed.
itemCount	Integer	The number of items included in the workspace.
status	String	The status of the workspace.

PUT: Update Workspace (updateWorkspaceNameAndDescription)

Update the name or description assigned to a workspace.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/{workspaceId}

Request Parameters

Parameter	Type	Description
workspaceId	Integer	The ID of the workspace to be updated. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.

Request Schema Fields

Schema Field	Type	Description
workspaceName	String	The new name to assign to the workspace.
workspaceDescription	String	The new description to assign to the workspace.

Response Schema Fields

Schema Field	Type	Description
workspaceName	String	The name of the workspace.
workspaceDescription	String	The description entered when the workspace was created.
workspaceType	String	The workspace type (**mail** or **file**).
workspaceMode	String	Indicates whether the workspace is a full workspace or a lite workspace.

Schema Field	Type	Description
workspaceSearchType	Integer	A value identifying the search store to use to build the workspace: • **1: Journaled Mails** • **2: Archived Mails** • **9: In-Place Mails** • **50: Archived Files** • **53: In-Place Files** • **55: Deleted Archive Files** • **56: Deleted In-Place Files**
workspaceId	Integer	The workspace ID.
createDate **lastAccessedDate**	String	The date and time the workspace was created and last accessed.
itemCount	Integer	The number of items included in the workspace.
status	String	The status of the workspace.

POST: Run Summarization on Document (summary)

Run summarization on a document. The endpoint will return an AI-generated summary of the document's contents. Use of the summarization feature has the following requirements:

- The **Enable Summarization Using AI for Inplace Files in Full Workspace** system registry setting must be enabled. You can check this status with the GET: Get Workspace System Registry Configuration endpoint described on page 479.
- The ZL Summary Service must be installed and started on your system. This requires additional system configuration. For further details, contact your System Administrator or refer to the *Enabling the ZL Summary Service* section of the *ZL UA Workspace Administrator's Guide*.
- The summarization feature is only available for files in full workspaces.

Please note that as of the release of ZL UA 11.1.2, the summarization feature is a Beta Level feature. It is not intended for use in production environments. Instructions are provided if you would like to try it in a demo or test system.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/{workspaceId}/item/{documentId}/summary

Request Parameters

Parameter	Type	Description
workspaceId	Integer	The ID of the full workspace containing the file to be summarized. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation. Make sure to check the **workspaceMode** and **workspaceType** fields in the endpoint's response data to make sure that you specify an ID for a full workspace for files. The summarization feature is not supported for lite workspaces or for mail documents.
documentId	String	The ID assigned to the file to be summarized. You can use the GET: Retrieve Data Source Items endpoint described on page 472 to retrieve a list of the items that have been added to each data source within a workspace. Use the **refItemId** response parameter as the document ID. Make sure that the **type** response parameter for the file is set to 12 (In-Place file). The summarization feature is not supported for archived files or mail documents.

Request Schema Fields

None.

Response Schema Fields

The response scheme in includes the **summary** string, which summarizes the contents of the specified file.

Workspace/Search

Use the Workspace/Search endpoints to run searches on your workspaces:

- *POST: Workspace Dataset Manipulation Search (dsmSearch)*
- *POST: Workspace Filter Search (filterSearch)*

POST: Workspace Dataset Manipulation Search (dsmSearch)

Use this endpoint apply a dataset manipulation to a workspace. You can use the Dataset Manipulation feature to apply logical operations to the messages or files that are returned by specific searches. When you create a Dataset Manipulation, you will select two or more data sources (i.e., a search that was used to create or add data to a workspace). You will also specify an operation to specify how the selected sources will be combined:

- **Union:** Returns the superset of items included in the data sources.
- **Intersect:** Returns common items included in the data sources.
- **Not**: Returns items included in the first data source, but not in the second.

For example, you could create a Dataset Manipulation representing the Intersection of 3 different searches to view the messages that were returned by **each** of the 3 searches. You could also create a Dataset Manipulation representing the Union of these 3 searches to view the messages that were returned by **any** of the 3 searches.

Path

http://localhost:8080/ps/api/v1/ws/workspace/workspaces/{workspaceId}/item/{documentId}/summary

Request Parameters

Parameter	Type	Description
Page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Schema Fields

Parameter	Type	Description
workspaceId	Integer	The ID of the workspace to be updated. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.
mergeOperation	String	Specify an operation to specify how the selected sources will be combined: • **Union:** Returns the superset of items included in the data sources. • **Intersect:** Returns common items included in the data sources. • **Not**: Returns items included in the first data source, but not in the second, third, fourth, etc.

Parameter	Type	Description
firstDataSource	String	Specify the ID of the first data source to be included. This setting is only required when **mergeOperation** is set to **Not**.
dataSources	String	Specify the IDs of the other data sources to be included as a comma-separated list.
sortParameter	String	Specify which field the items included in the search results should be sorted by. For files, you can choose from the following options: **size** **name** **createDate** **lastModified** **lastAccessed** **relevancy** For mails, you can choose from the following options: **sender** **to** **subject** **date** **size** **relevancy** Use the **sortDirection** field to specify whether the items included in the search results should be sorted ascending or descending order. Leave the **sortParameter** field empty if you do not want the data sorted.
sortDirection	String	Specify the sort order for the items listed in the search results. Specify **ascending** or **descending**.
start	Integer	The starting index for the search results.
itemsPerPage	Integer	The number of search results to be returned per page.

Response Schema Fields

Schema Field	Type	Description
items	Array	The **items** array includes an entry for each file or message returned by the search. Each entry includes the following fields:
rowID	Integer	Internal use only.

Schema Field	Type	Description
viewItemId	Integer	The view ID assigned to the item.
refItemId	Integer	The reference ID assigned to the item.
size	Integer	The size of the item, in KB or MB.
relevancy	Integer	The relevancy rating assigned to the item. The relevancy rating is an indication of how well the item matched the search criteria used to build the data source. A low relevancy rating indicates the message matched a greater part of the search criteria (e.g., items that are assigned a relevancy rating of 100.0 matched the most search criteria). **Please note that relevancy scores are a Beta Level feature as of the release of ZL UA 11.1.2.**
type	Integer	The type of the file or message.
deletedSearchItem	Boolean	If True, it indicates that the item was retrieved from a search store for deleted items.
totalRecords	Integer	The number of items returned by the search.
viewId	String	The view ID assigned to the search result set.
displayedItemCount	Integer	The number of items displayed in the search results.
missCacheId	String	Reserved for future use.

POST: Workspace Filter Search (filterSearch)

Search the files or messages included in a workspace.

Path

http://localhost:8080/ps/api/v1/ws/search/search/filtersearch

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Schema Fields

By default, all string-type fields in the request schema are passed as empty strings, and all other fields are passed as null values. The search results will be filtered by any schema fields for which you specify input – meaning that if you were to specify a workspace ID and none of the other request schema fields, the search would return every item in the workspace.

Schema Field	Type	Description
workspaceId	Integer	The ID of the workspace to be viewed. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation.

Schema Field	Type	Description
bodyCondType **body**	String	**These fields are only applicable when searching messages. Leave them empty when searching files.** Use these fields to search a message body for specific words or phrases: • **bodyCondType:** Specify the how the keywords specified for the **body** field will be used to search the workspace. Specify **anyPhrase** to search for messages that include any of the keywords, or **allPhrase** to search for messages that include all the keywords. Examples are provided below. • **body:** Specify the words or phrases to search for. For example, if you set **body** to *guaranteed full rebate* and **bodyCondType** to **anyPhrase,** the operation would return messages that includes any instance of *rebate*, *full*, or *guaranteed* in the message body. If you set **bodyCondType** to **allPhrase,** the operation would return messages that include all these words in the message body.
enableBodyExclusion **excludeBodyCondType** **excludeBody**	Boolean String String	**These fields are only applicable when searching messages. Leave them empty when searching files.** Set the **enableBodyExclusion** field to True to filter the search results by body content exclusion. In this case, messages that include certain keywords or phrases will be **excluded from** the search results. Use the remaining fields to configure the filter settings. • **excludeBodyCondType:** Specify how the keywords specified for the **excludeBody** field will be used to search the workspace. Specify **anyPhrase** to exclude messages that include any of the keywords, or **allPhrase** to exclude messages that include all of the keywords. Examples are provided below. • **excludeBody:** Specify the words or phrases to search for. For example, if you set **excludeBody** to *guaranteed full rebate* and **excludeBodyCondType** to **anyPhrase,** the search results would exclude messages that includes any instance of *rebate*, *full*, or *guaranteed* in the message body. If you set **excludeBodyCondType** to **allPhrase,** the search results would exclude messages that include all these words in the message body.

Schema Field	Type	Description
sortParameter	String	Specify which field the items included in the search results should be sorted by. For files, you can choose from the following sort options: **size** **name** **createDate** **lastModified** **lastAccessed** **relevancy** For messages, you can choose from the following sort options: **sender** **to** **subject** **date** **size** **relevancy** Use the **sortDirection** field to specify whether the search results should be sorted ascending or descending order. Leave the **sortParameter** field empty if you do not want the data sorted.
sortDirection	String	Specify the order for the search results. Specify **ascending** or **descending**.
start	Integer	Starting index of the search results (for pagination).
itemsPerPage	Integer	Number of search results to be returned per page.

Schema Field	Type	Description
anyUserFlag **anyUserField** **rawFrom** **rawTo** **rawCc** **rawBcc**	Boolean String String String String String	**These fields are only applicable when searching messages. Leave them empty when searching files.** Use the **anyUserFlag** field to determine how to search the email addresses included in each message's **To**, **From**, **CC** or **BCC** lists: • **True:** Set **anyUseFlag** to True to search for messages sent to or received by a specific user or group of users. In this case, use the **anyUserField** to specify the email address of the users you want to search for. Enter the email addresses as a comma-separated list. The response data will include messages where the specified address was included in any of the **To**, **From**, **CC** or **BCC** address lists for the message. Leave the **rawFrom, rawTo, rawCc** and **rawBcc** fields empty when the **anyUserFlag** field is set to True. • **False:** Set **anyUserFlag** to False if you want to search the message's **To**, **From**, **CC** or **BCC** address lists separately, or for different email addresses. In this case, use the **rawFrom, rawTo, rawCc** and **rawBcc** fields to specify which email addresses to search for within each address list. Enter the email addresses for each field as a comma-separated list. The response data will include messages where the specified addresses were included in the **To**, **From**, **CC** or **BCC** lists for the message, as appropriate. Leave the **anyUserField** field empty when **anyUserFlag** is set to False.

Schema Field	Type	Description
enableUserExclusion **excludeAnyUserFlag** **excludeAnyUserField** **exlcludeRawFrom** **excludeRawTo** **excludeRawCc** **excludeRawBcc**	Boolean Boolean String String String String String	**These fields are only applicable when searching messages. Leave them empty when searching files.** Set the **enableUserExclusion** field to True to filter the search results by message exclusion. In this case, messages sent to or from specific users will be **excluded from** the search results. Use the remaining fields to configure the filter settings. Use the **excludeAnyUserFlag** field to determine how to search the email addresses included in each message's **To**, **From**, **CC** or **BCC** lists: • **True:** Set **excludeAnyUserFlag** to True to search for messages sent to or received by a specific user or group of users. In this case, use the **excludeAnyUserField** to specify the email address of the users you want to search for. Enter the email addresses as a comma-separated list. The response data will exclude any message where the specified address was included in any of the **To**, **From**, **CC** or **BCC** address lists for the message. Leave the **excludeRawFrom, excludeRawTo, excludeRawCc** and **excludeRawBcc** fields empty when the **excludeAnyUserFlag** field is set to True. • **False:** Set **excludeAnyUserFlag** to False if you want to search the message's **To**, **From**, **CC** or **BCC** address lists separately, or for different email addresses. In this case, use the **excludeRawFrom, excludeRawTo, excludeRawCc** and **excludeRawBcc** fields to specify which email addresses to search for within each address list. Enter the email addresses for each field as a comma-separated list. The response data will exclude messages where the specified addresses were included in the **To**, **From**, **CC** or **BCC** lists for the message, as appropriate. Leave the **anyUserField** field empty when **anyUserFlag** is set to False.
datasetType	String	Set this field to **allData** to search all the data in the workspace. Other values are not supported.

Schema Field	Type	Description
nameCondType **name**	String	**These fields are only applicable when searching files. Leave them empty when searching mail documents.** Use them to search the file names for a specific phrase: • **nameCondType:** Specify **anyPhrase** to search for files that include any of the specified phrases, or **allPhrase** to search for files that include all of the specified phrases. • **name:** Specify the words or phrases to search for within the file names. For example, if you set **name** to *one specific item* and **nameCondType** to **anyPhrase,** the operation would return files that include any of those words in the file name. If you set **nameCondType** to **allPhrase,** the operation would return files that include all the words in the name.
srchMsgSizeMode	Integer	To filter the search by message size, follow these steps: 1. Specify the mode to use to filter messages based on size: • -1 for **any size** • 3 for **less than** • 4 for **greater than** • 5 for **between** 2. Use the **searchMsgSizeLow** and **searchMsgSizeLowUnit** fields described below to specify the lower limit of the size range. 3. Use the **searchMsgSizeHigh** and **searchMsgSizeHighUnit** fields described below to specify the upper limit of the size range. Steps 2 and 3 may not be necessary depending on the **searchMsgSizeMode** selected. For example, if you choose **4** for **greater than**, you would only need to specify the lower limit of the size range. The search would return all messages that are larger than the lower limit. **Leave these fields empty when searching files.**
srchMsgSizeLow	String	Lower limit of message size for filtering.
srchMsgSizeLowUnit	Integer	Unit for the lower size limit: • **2**: Kilobytes • **3**: Megabytes

Schema Field	Type	Description
srchMsgSizeHigh	String	Upper limit of message size for filtering.
srchMsgSizeHighUnit	Integer	Unit for the upper size limit (e.g., KB, MB).
dateStart **dateEnd** **dateMode**	String	Set **dateMode** to **between** to return messages that were sent (or files that were created) during a specific date range. In this case, use the **dateStart** and **dateEnd** fields to specify the date range. The date range is inclusive. Set **dateMode** to **any** to return all messages or files, regardless of when the messages were sent or the files were created. In this case, leave the **dateStart** and **dateEnd** fields empty.
msgFlags	String	Flags to include messages with specific properties. Specify **MF_ATTACH** to only include messages that have attachments. Other values are not supported as of the release of ZL UA 11.1.
excludeMsgFlags	String	Flags to exclude messages with specific properties. Specify **MF_ATTACH** to exclude messages that have attachments. Other values are not supported as of the release of ZL UA 11.1.
lastModifiedDateStart **lastModifiedDateEnd** **lastModifiedDateMode**	String	**These fields are only applicable when searching files. Leave them empty when searching messages.** Set **lastModifiedDateMode** to **between** to return files that were last modified during a particular date range. In this case, use the **lastModifiedDateStart** and **lastModifiedDateEnd** fields to establish the date range. The start and end dates are inclusive. Set **lastModifiedDateMode** to **any** to return all files, regardless of when they were last modified. In this case, leave the **lastModifiedDateStart** and **lastModifiedDateEnd** fields empty.
updateItemCount	Boolean	For full workspaces, this parameter is not required. For lite workspaces, set to True to update the item count assigned to the workspace. The item count may change as items are added and removed from the search index. ZL recommends setting this to True the first time you access a data source with this endpoint, and False for subsequent requests.

Response Schema Fields

Schema Field	Type	Description
items	Array	The **items** array includes an entry for each file or message returned by the search. Each entry includes the following fields:
rowID	Integer	Internal use only.
viewItemId	Integer	The view ID assigned to the item.
refItemId	Integer	The reference ID assigned to the item.
size	Integer	The size of the item, in KB or MB.
relevancy	Integer	The relevancy rating assigned to the item. The relevancy rating is an indication of how well the item matched the search criteria used to build the data source. A low relevancy rating indicates the message matched a greater part of the search criteria (e.g., items that are assigned a relevancy rating of 100.0 matched the most search criteria). **Please note that relevancy scores are a Beta Level feature as of the release of ZL UA 11.1.2.**
type	Integer	The type of the file or message.
deletedSearchItem	Boolean	If True, it indicates that the item was retrieved from a search store for deleted items.
totalRecords	Integer	The number of items returned by the search.
viewId	String	The view ID assigned to the search result set.
displayedItemCount	Integer	The number of items displayed in the search results.
missCacheId	String	Reserved for future use.

Workspace/Stats

This section describes endpoints you can use to view cross-tabulation data from the workspaces that have been added to the Enterprise Analytics module. Use these endpoints to generate cross-tabulation matrices comparing groups of files in full workspaces based on file attributes such **File Type**, **File Size**, **Date Created** and **Last Modified Date.** **These endpoints are not applicable to messages or to files in lite workspaces.**

For example, you could create a cross-tabulation matrix that indicates how many files that match various file size ranges were created during each calendar month, or how many files of certain file types fit into each size range.

This includes the following endpoints:

- *POST: Create Cross-Tabulation Matrix (crosstab)*: Use this endpoint to create a cross-tabulation matrix for a workspace.
- *POST: View Cross-Tabulation Items (crosstabItems):* Use this endpoint to view files included in the cross-tabulation matrix that was most recently created within the workspace.

As of the release of ZL UA 11.1.2, the tabulation feature is a Beta Level feature. It is not intended for use in production environments. Instructions are provided if you would like to try it in a demo or test system.

POST: Create Cross-Tabulation Matrix (crosstab)

Use this endpoint to create a cross-tabulation matrix comparing groups of files in full workspaces based on file attributes such as **File Type**, **File Size**, **Date Created** and **Last Modified Date.** For example, you could create a cross-tabulation matrix that indicates how many files that match various file size ranges were created during each calendar month, or how many files of certain file types fit into each size range.

After creating the cross-tabulation matrix, you can view the files it included with the *POST: View Cross-Tabulation Items (crosstabItems)* described on page 512.

As of the release of ZL UA 11.1.2, the tabulation feature is a Beta Level feature. It is not intended for use in production environments. Instructions are provided if you would like to try it in a demo or test system.

Path

http://localhost:8080/ps/api/v1/workspacestats/crosstab

Request Parameters

None.

Request Schema Fields

Schema Field	Type	Description
searchParams: Use the **searchParams** fields to identify the workspace to be viewed, and the files for which you would like to create a cross-tabulation matrix.		
workspaceId	Integer	The ID of the workspace for which you would like to generate the cross-tabulation matrix. You can use the GET: Get All Workspaces endpoint described on page 462 to retrieve the ID values for all workspaces in a single operation. The tabulation feature is only available for files in full workspaces. Make sure that the **workspaceType** is set to **file** and the **workspaceMode** is set to **full** for the workspace you choose.
Use the remaining **searchParams** fields as search filters to identify the files for which you want to create the cross-tabulation matrix. By default, all string-type fields are passed as empty strings, and all other fields are passed as null values. The search results will be filtered by any schema fields for which you specify input – meaning that if you were to specify a workspace ID and none of the other **searchParams** fields, the cross-tabulation matrix would return every item in the workspace. Please note that some **searchParams** fields displayed in the Swagger file in ZL UA 11.1.2 are not used by the endpoint. These are marked as reserved in the table below. Do not modify their values when executing this endpoint.		

Schema Field	Type	Description
bodyCondType **body**	String	**These fields are only applicable when searching messages. Leave them empty when searching files.** Use these fields to search a message body for specific words or phrases: • **bodyCondType:** Specify the how the keywords specified for the **body** field will be used to search the workspace. Specify **anyPhrase** to return messages that include any of the keywords, or **allPhrase** to return messages that include all of the keywords. Examples are provided below. • **body:** Specify the words or phrases to search for. For example, if you set **body** to *guaranteed full rebate* and **bodyCondType** to **anyPhrase,** the operation would return messages that includes any instance of *rebate*, *full*, or *guaranteed* in the message body. If you set **bodyCondType** to **allPhrase,** the operation would return messages that include all these words in the message body.
enableBodyExclusion **excludeBodyCondType** **excludeBody**	Boolean String String	**These fields are only applicable when searching messages. Leave them empty when searching files.** Set the **enableBodyExclusion** field to True to filter the search results by body content exclusion. In this case, messages that include certain keywords or phrases will be **excluded from** the search results. Use the remaining fields to configure the filter settings. • **excludeBodyCondType:** Specify how the keywords specified for the **excludeBody** field will be used to search the workspace. Specify **anyPhrase** to exclude messages that include any of the keywords, or **allPhrase** to exclude messages that include all of the keywords. Examples are provided below. • **excludeBody:** Specify the words or phrases to search for. For example, if you set **excludeBody** to *guaranteed full rebate* and **excludeBodyCondType** to **anyPhrase,** the search results would exclude messages that includes any instance of *rebate*, *full*, or *guaranteed* in the message body. If you set **excludeBodyCondType** to **allPhrase,** the search results would exclude messages that include all these words in the message body.
sortParameter	String	Reserved.
sortDirection	String	Reserved.

Schema Field	Type	Description
start	Integer	Reserved.
itemsPerPage	Integer	Reserved.
anyUserFlag **anyUserField** **rawFrom** **rawTo** **rawCc** **rawBcc**	Boolean String String String String String	**These fields are only applicable when searching messages. Leave them empty when searching files.** Use the **anyUserFlag** field to determine how to search the email addresses included in each message's **To**, **From**, **CC** or **BCC** lists: • **True:** Set **anyUseFlag** to True to search for messages sent to or received by a specific user or group of users. In this case, use the **anyUserField** to specify the email address of the users you want to search for. Enter the email addresses as a comma-separated list. The response data will include messages where the specified address was included in any of the **To**, **From**, **CC** or **BCC** address lists for the message. Leave the **rawFrom, rawTo, rawCc** and **rawBcc** fields empty when the **anyUserFlag** field is set to True. • **False:** Set **anyUserFlag** to False if you want to search the message's **To**, **From**, **CC** or **BCC** address lists separately, or for different email addresses. In this case, use the **rawFrom, rawTo, rawCc** and **rawBcc** fields to specify which email addresses to search for within each address list. Enter the email addresses for each field as a comma-separated list. The response data will include messages where the specified addresses were included in the **To**, **From**, **CC** or **BCC** lists for the message, as appropriate. Leave the **anyUserField** field empty when **anyUserFlag** is set to False.

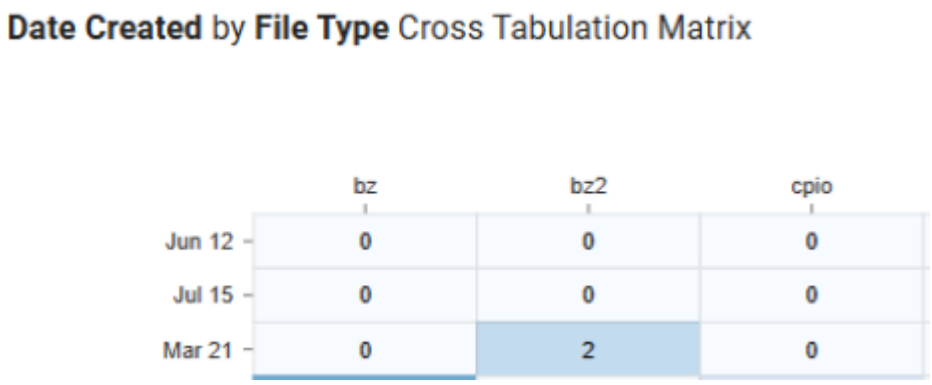
Schema Field	Type	Description
enableUserExclusion **excludeAnyUserFlag** **excludeAnyUserField** **exlcludeRawFrom** **excludeRawTo** **excludeRawCc** **excludeRawBcc**	Boolean Boolean String String String String String	**These fields are only applicable when searching messages. Leave them empty when searching files.** Set the **enableUserExclusion** field to True to filter the search results by message exclusion. In this case, messages sent to or from specific users will be **excluded from** the search results. Use the remaining fields to configure the filter settings. Use the **excludeAnyUserFlag** field to determine how to search the email addresses included in each message's **To**, **From**, **CC** or **BCC** lists: • **True:** Set **excludeAnyUserFlag** to True to search for messages sent to or received by a specific user or group of users. In this case, use the **excludeAnyUserField** to specify the email address of the users you want to search for. Enter the email addresses as a comma-separated list. The response data will exclude any message where the specified address was included in any of the **To**, **From**, **CC** or **BCC** address lists for the message. Leave the **excludeRawFrom, excludeRawTo, excludeRawCc** and **excludeRawBcc** fields empty when the **excludeAnyUserFlag** field is set to True. • **False:** Set **excludeAnyUserFlag** to False if you want to search the message's **To**, **From**, **CC** or **BCC** address lists separately, or for different email addresses. In this case, use the **excludeRawFrom, excludeRawTo, excludeRawCc** and **excludeRawBcc** fields to specify which email addresses to search for within each address list. Enter the email addresses for each field as a comma-separated list. The response data will exclude messages where the specified addresses were included in the **To**, **From**, **CC** or **BCC** lists for the message, as appropriate. Leave the **anyUserField** field empty when **anyUserFlag** is set to False.
datasetType	String	Set this field to **allData** to search all the data in the workspace. Other values are not supported.

Schema Field	Type	Description
nameCondType **name**	String	**These fields are only applicable when searching files. Leave them empty when searching mail documents.** Use them to search the file names for a specific phrase: • **nameCondType:** Specify **anyPhrase** or **allPhrase**. • **name:** Specify the words or phrases to search for within the file names. For example, if you set **name** to *one specific item* and **nameCondType** to **anyPhrase,** the operation would return files that include any of those words in the file name. If you set **nameCondType** to **allPhrase,** the operation would return files that include all the words in the name.
srchMsgSizeMode	Integer	To filter the search by message size, follow these steps: 1. Specify the mode to use to filter messages based on size: • -1 for **any size** • 3 for **less than** • 4 for **greater than** • 5 for **between** 2. Use the **searchMsgSizeLow** and **searchMsgSizeLowUnit** fields described below to specify the lower limit of the size range. 3. Use the **searchMsgSizeHigh** and **searchMsgSizeHighUnit** fields described below to specify the upper limit of the size range. Steps 2 and 3 may not be necessary depending on the **searchMsgSizeMode** selected. For example, if you choose **4** for **greater than**, you would only need to specify the lower end of the size range. The search would include messages that are larger than the lower limit. **Leave these fields empty when searching files.**
srchMsgSizeLow	String	Lower limit of message size for filtering.
srchMsgSizeLowUnit	Integer	Unit for the lower size limit: • **2**: Kilobytes • **3**: Megabytes
srchMsgSizeHigh	String	Upper limit of message size for filtering.

Schema Field	Type	Description
srchMsgSizeHighUnit	Integer	Unit for the upper size limit (e.g., KB, MB).
dateStart **dateEnd** **dateMode**	String	Set **dateMode** to **between** to return messages that were sent (or files that were created) during a specific date range. In this case, use the **dateStart** and **dateEnd** fields to specify the date range. The date range is inclusive. Set **dateMode** to **any** to return all messages or files, regardless of when the messages were sent or the files were created. In this case, leave the **dateStart** and **dateEnd** fields empty.
msgFlags	String	Flags to include messages with specific properties. Specify **MF_ATTACH** to only include messages that have attachments. Other values are not supported as of the release of ZL UA 11.1.
excludeMsgFlags	String	Flags to exclude messages with specific properties. Specify **MF_ATTACH** to exclude messages that have attachments. Other values are not supported as of the release of ZL UA 11.1.
lastModifiedDateStart **lastModifiedDateEnd** **lastModifiedDateMode**	String	**These fields are only applicable when searching files. Leave them empty when searching messages.** Set **lastModifiedDateMode** to **between** to return files that were last modified during a particular date range. In this case, use the **lastModifiedDateStart** and **lastModifiedDateEnd** fields to establish the date range. The start and end dates are inclusive. Set **lastModifiedDateMode** to **any** to return all files, regardless of when they were last modified. In this case, leave the **lastModifiedDateStart** and **lastModifiedDateEnd** fields empty.
updateItemCount	Boolean	Reserved.

Schema Field	Type	Description
Use the remaining fields to specify how the cross-tabulation matrix will be created (i.e., what file attributes it will include, and how they will be presented).		
rowCategory **columnCategory**	String	The **rowCategory** and **columnCategory** fields define the file attributes that will be used to create the cross-tabulation matrix. Specify any of the following values: • **createDate** • **lastModifiedDate** • **fileType** • **size** For example, you could specify **fileSize** as the **rowCategory** and **fileType** as the **colCategory** to create a cross-tabulation matrix that indicates how many files of various file types fit into various size ranges.

Response Schema Fields

Schema Field	Type	Description
values	Integer	The **values** array presents the values that are included in the cross-tabulation matrix. These are listed in the order in which they would appear in the cross-tabulation matrix moving from left-to-right across each row, from the top row to the bottom. For example, in the cross-tabulation matrix below, the first three values would represent the **June 2012** values for **bz**, **bz2** and **cpio**. The next three values would represent the July 2015 values for these file types. The final three would represent the August 2021 values for these file types.  In this example, the **values** array would be presented as: `[ [0,0,0], [0,0,0], [0,2,0] ]` You can view information included for each file or message included within specific cells of the cross-tabulation matrix with the *POST: View Cross-Tabulation Items (crosstabItems)* described on page 512.
minValue	Integer	The minimum value present.
maxValue	String	The maximum value present.
colSearchKeys **rowSearchKeys**	String	After creating the cross-tabulation matrix, you can view the specific files it includes with the *POST: View Cross-Tabulation Items (crosstabItems)* endpoint described on page 512. These values list the search keys you can provide as input to this endpoint. For example, if **rowCategory** was set to **size** in the request input, the **rowSearchKeys** field would include a series of strings you could use to view files that fit the different size ranges. Or if it were set to **fileType,** the **rowSearchKeys** field would include a series of strings you could use to view different file types.

Schema Field	Type	Description
rowCategories	String	Lists the rows added to the cross-tabulation matrix. For example, if **rowCategory** is set to **size** in the request**,** this would include values identifying the file size ranges for the files returned by the search filter criteria specified in the request. The cross-tabulation matrix would include a row for each of these file size ranges. This could include values such as: **0B – 10KB** **10KB – 100KB** **500K – 1MB** **1MB – 5MB** **5+ MB**
colCategories	String	Lists the columns added to the cross-tabulation matrix. For example, if **colCategory** is set to **fileType** in the request**,** this would include values identifying the file types of the files returned by the search filter criteria specified in the request. The cross-tabulation matrix would include a column for each of these file types. This could include values such as: **csv** **docx** **html** **pptx**

POST: View Cross-Tabulation Items (crosstabItems)

Use this endpoint to view files included within the cross-tabulation matrix most recently created for a workspace with the POST: Create Cross-Tabulation Matrix (crosstab) endpoint. This endpoint includes input parameters you can use to specify a cell within the cross-tabulation matrix. The response data will include information regarding the files included with that cell. It also provides the option to specify the same search filter options if you would like to view a limited set of data within the cross-tabulation matrix.

For example, you could create a cross-tabulation matrix based on file size (**size**) and (**createDate)**. The cross-tabulation matrix would include information regarding how many files that match various file size ranges were created during each calendar month. In this example, you could use this endpoint to view all files 5MB or greater that were created during a given calendar month.

As of the release of ZL UA 11.1.2, the tabulation feature is a Beta Level feature. It is not intended for use in production environments. Instructions are provided if you would like to try it in a demo or test system.

Path

http://localhost:8080/ps/api/v1/workspacestats/crosstab/items

Request Parameters

Request Parameters

Parameter	Type	Description
page	Integer	Some operations return too many values for the UI to display. You can use the **page** parameter to indicate the page number on which these values should be displayed, and the **pageSize** parameter to indicate the maximum number of items that can be displayed on the page. The **pageSize** parameter defaults to 200 and has a maximum size of 1,000.
pageSize	Integer	

Request Schema Fields

Schema Field	Type	Description
searchParams: Specify the same set of input that was used to create the cross-tabulation matrix with the POST: Create Cross-Tabulation Matrix (crosstab) endpoint. Make sure not to alter or omit any values, as the original request is not cached and cannot be referenced by this call.		
Use the remaining fields to specify which cell within the cross-tabulation matrix you would like to view.		
rowCategory **rowValue**	String	For the **rowCategory** field, specify the row you want to view: • **createDate** • **lastModifiedDate** • **fileType** • **size** For the **rowValue**, specify the category value you want to view. For example, if you set **rowCategory** to **fileType**, you could set **rowValue** to **html** to view data for HTML files. The available strings you can supply as input were included in the response data returned when you created the cross-tabulation matrix with the POST: Create Cross-Tabulation Matrix (crosstab) endpoint described on page 503. The **rowCategories** field includes the string you can use to specify the **rowCategory**, and the **rowSearchKeys** field includes the strings you can use to specify the **rowValue**.

Schema Field	Type	Description
colCategory **colValue**	String	For the **colCategory** field, specify the column you want to view: • **createDate** • **lastModifiedDate** • **fileType** • **size** For the **colValue**, specify the value you want to view. For example, if you set **colCategory** to **size**, you could enter **0B,10KB** here to view data for files between 0-10Kb. If you used these settings in conjunction with the previous example (where **rowCategory** is set to **filetype** and **rowValue** is set to **html**), the endpoint would return information for HTML files that are sized between 0-10Kb. The available strings you can supply as input were included in the response data returned when you created the endpoint with the POST: Create Cross-Tabulation Matrix (crosstab) endpoint described on page 503. The **colCategories** field includes the string you can use to specify the **colCategory**, and the **colSearchKeys** field includes the strings you can use to specify the **colValue**.

Response Schema Fields

Schema Field	Type	Description
items	Array	The **items** array includes an entry for each file or message included in the cross-tabulation matrix cell specified in the request data. Each entry includes the following fields:
rowID	Integer	Internal use only.
viewItemId	Integer	The view ID assigned to the item.
refItemId	Integer	The reference ID assigned to the item.
size	Integer	The size of the item, in KB or MB.

Schema Field	Type	Description
relevancy	Integer	The relevancy rating assigned to the item. The relevancy rating is an indication of how well the item matched the search criteria used to build the data source. A low relevancy rating indicates the message matched a greater part of the search criteria (e.g., items that are assigned a relevancy rating of 100.0 matched the most search criteria). **Please note that relevancy scores are a Beta Level feature as of the release of ZL UA 11.1.2.**
type	Integer	The type of the file or message.
deletedSearchItem	Boolean	If True, it indicates that the item was retrieved from a search store for deleted items.
totalRecords	Integer	The number of items returned by the search.
viewId	String	The view ID assigned to the search result set.
displayedItemCount	Integer	The number of items displayed in the search results.
missCacheId	String	Reserved for future use.