



ZL UA REST API Kit Reference Guide

ZL UA Version 10.3.2 **PRELIMINARY BETA DRAFT**

Last Updated: May 29, 2024

Document Revision 1.0



Copyright © 2024 ZL Technologies, Inc.

All rights reserved.

ZL Technologies, Inc. (ZLTI, formerly known as “ZipLip”) and its licensors retain all ownership rights to the software programs offered by ZLTI (referred to herein as “Software”) and related documentation. Use of the Software and related documentation is governed by the license agreement accompanying the Software and applicable copyright law.

Your right to copy this documentation is limited by copyright law. Making unauthorized copies, adaptations, or compilation works is prohibited and constitutes a punishable violation of the law. ZLTI may revise this documentation occasionally without notice.

Disclaimer

THIS DOCUMENTATION IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND. IN NO EVENT SHALL ZLTI BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND ARISING FROM ANY ERROR IN THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION ANY LOSS OR INTERRUPTION OF BUSINESS, REVENUE, USE, OR DATA.

Other product and company names appearing in ZLTI products and materials are used for identification purposes only and may be trademarks or registered trademarks of their respective companies. Registered and unregistered trademarks used in any ZLTI products and materials are the exclusive property of their respective owners.

Contents

Introduction to REST API	1
Accessing the REST API	6
Discovery/CaseDataSource.....	7
GET: Get All Case Data Source Runs (getallcasedatasourceruns).....	8
GET: Get Case Data Source Run Schedule (getcasedatasourcerunschedule)	10
GET: Get Case Data Source Using Name (getcasedatasourceusingname)	11
GET: Get Latest Case Data Source Run (getlatestcasedatasourcerun).....	13
POST: Schedule Case Data Source (scheduleDataSource).....	15
DELETE: Unschedule Data Source (unscheduleDataSource)	16
Discovery/Cases.....	17
POST: Create a Tag (addtag)	27
POST: Create a Case (createcase).....	29
DELETE: Delete Case (deleteCase)	36
DELETE: Delete Tag (deleteTag).....	38
GET: Get All Case Data Sources (getallcasedatasources)	39
GET: Get Case Info (getallcaseusingname)	41
GET: Get Case Info (getallcaseusingnamepattern).....	42
GET: Get All Case Tags (getallcasetags)	43
GET: Get All Case User Privileges (getallcaseuserprivileges)	46
GET: Get All Child Tags (getallchildtags)	48
GET: Get Case Data Source Using ID (getcasedatasourceusingsrcid).....	52
GET: Get Case Using ID (getcaseusingid).....	56
GET: Get Case Items (getcaseitem)	58
PUT: Update Tag (updatetag)	62
Discovery/Privileges.....	69
DELTE: Delete All Privileges Within a Case (deleteallprivileges)	70
DELTE: Delete a User's Privileges Within a Case (deleteprivilege).....	71
GET: Get User Privileges (getuserprivileges)	72
PUT: Update User Privileges (updateuserprivileges).....	74

Discovery/Tasks	76
POST: Run Background Task	77
GET: Get Task Status (gettaskstatus)	79
Discovery/Custodians	81
POST: Add Custodian (addcustodian)	82
PUT: Add Custodian Alias (addalias)	84
GET: Get All Aliases (getallaliases)	86
GET: Get All Custodians (getallcustodians)	88
GET: Get Custodian Preservation Using ID (getusingid)	90
GET: Get All Custodian Preservations (getallcustodianpreservation)	94
GET: Get Custodian Using Address (getusingaddress)	96
GET: Get Custodian Using Address (getusingid)	98
DELETE: Delete Custodian (delete)	100
DELETE: Delete Custodian Alias (deletealias)	101
PUT: Update Custodian (updatecustodian)	103
UAA/Departments	105
POST: Create Child Department (createchilddepartment)	106
DELETE: Delete Department (deleteDepartment)	109
GET: Get Child Departments (getallchilddepartment)	110
GET: Get All Departments Using Name (getalldeptsusingnamepattern)	111
GET: Get Users of a Department (getallusers)	112
GET: Get Department Using ID (getdepartmentusingid)	115
GET: Get Department By Name (getdepartmentusingname)	116
GET: Get Department Parent Information By ID (getparentdepartment)	117
GET: Get Department Policy Information Using ID (getpolicy)	118
PUT: Update Department (updatedepartment)	119
PUT: Update Parent Department (updateparentdepartment)	121
Discovery/Roles	137
PUT: Grant User Roles (grantrroles)	138
PUT: Revoke User Roles (revokeroles)	140
FAM/Roles	142

PUT: Grant User Roles (grantroles)	142
PUT: Revoke User Roles (revokeroles)	144
UAA/Data Sources	146
POST: Create An EWS Mail Server (createmailserver/ews).....	147
POST: Create File Server (createFileServer).....	149
POST: Create a Google Drive Server (creategoogledriveserver)	151
POST: Create a Google Mail Server (creategooglemailserver)	153
POST: Create a Lotus Domino Mail Server (createlotusdominomailserver)	155
POST: Create an Exchange Mail Server (createmailserver/microsoftexchange).....	157
POST: Create a OneDrive Server (createonedriveserver).....	159
POST: Create a SharePoint Server (createsharepointserver)	161
DELETE: Delete Server (deletemailserver).....	163
GET: Get All Servers (getallmailservers)	164
GET: Get Server Using ID (getmailserverusingid)	166
GET: Get Server Using Name (getmailserverusingname)	168
PUT: Update an EWS Mail Server (updateewsmailserver)	169
PUT: Update File Server (updatefileservice).....	171
PUT: Update a Google Drive Server (updategoogledriveserver).....	173
PUT: Update a Google Mail Server (updategooglemailserver)	175
PUT: Update a Lotus Domino Mail Server (updatelotusdominoserver).....	177
PUT: Update an Exchange Mail Server (updatemailserver/exchange)	179
PUT: Update a OneDrive Server's Configuration (updateonedriveserver)	181
PUT: Update a SharePoint Server (updatesharepointserver).....	183
FAM/Dashboards	185
POST: Create Dashboard (createdashboard).....	185
DELETE: Delete Dashboard (deletedashboard)	187
POST: Download Dashboard Report (downloaddashboardreport).....	188
GET: Get All Dashboards (getalldashboards).....	190
GET: Get All Dashboards of a User (getalldashboards)	191
GET: Get All Dashboards of a User Based on Status (getalldashboards).....	192
GET: Get Dashboard Using ID (getdashboardusingid)	193

GET: Get Default Dashboard (getdefaultdashboard)	194
PUT: Update Dashboard (updatedashboard)	195
FAM/Disposition	197
PUT: Disable Disposition (disabledisposition)	199
PUT: Enable Disposition (enabledisposition)	200
GET: Get Latest Disposition Run (getlatestrun)	201
GET: Get Projects Pending Approval (getprojectspendingforapproval)	203
GET: Get Projects Ready For Disposition (getprojectsreadyfordisposition)	204
POST: Start Disposition (startdisposition)	205
FAM/Reports	206
POST: Download Audit Trail Report (downloadaudittrailreport)	207
POST: Download File Analysis Report (downloadfileanalysisreport)	210
POST: Download File Manifest Report (downloadmanifestreport)	211
POST: Download PII Report (downloadpiireport)	212
POST: Download Remediation Report (downloadremediationreport)	213
POST: Generate File Tree Disposition Report (generatedispositionreport)	214
POST: Generate File Tree Pre-Disposition Report (generatepredispositionreport)	215
FAM/Tasks	216
POST: Update Content Index (updatecontentindex)	217
GET: Get Task Status (getTaskStatus)	218
POST: Run All Searches (runallsearches)	220
POST: Run Content Tagger (runcontenttagger)	221
POST: Run Crawl (runcrawl)	222
POST: Run Metadata Tagger (runmetadatatagger)	223
POST: Run PII Tagger (runpiitagger)	224
POST: Run Remediation Task (runremediation)	225
POST: Update Index (updateindex)	226
FAM/Project Privileges	227
PUT: Add or Grant Group Project Privileges (addgroupprojectprivileges, grantgroupprivileges)	228
PUT: Add or Grant User Project Privileges (adduserprojectprivileges, grantuserprojectprivileges)	230
GET: Get Project Entity IDs (getprojectentityids)	232

GET: Get Project Entity Privileges (getprojectentityprivileges)	233
GET: Get Project Entity Types (getprojectentitytypes)	234
GET: Get Project Privileges Using Project ID (getprojectprivileges)	235
PUT: Revoke All Project Privileges (revokeallprojectprivileges)	236
PUT: Revoke Group Project Privileges (revokegroupprojectprivileges)	237
PUT: Revoke User Project Privileges (revokeuserprojectprivileges)	238
UAA Projects	239
POST: Create Google Drive Project (creategoogledriveproject)	240
POST: Create OneDrive Project (createonedriveproject)	244
POST: Create Project - Supports File Share Projects Only (create)	247
POST: Create SharePoint Project	251
DELETE: Remove Project (deleteusingid)	257
GET: Get All Projects (getallprojects)	258
GET: Get Sub-Folders (subfolders)	259
GET: Get Project Info by ID (getprojectusingid)	261
GET: Get Project Info by Name (getprojectusingname)	262
GET: Get Projects of File Server (getprojectsusingidfileserver)	263
GET: Get Projects Using Search (getprojectusingpatternsearch)	264
PUT: Update OneDrive Project (updateonedriveproject)	265
PUT: Update Project - File Share Projects Only (update)	266
PUT: Update SharePoint Project (updatesharepointproject)	267
UAA/Roles	268
GET: Get All Custom Roles (getallcustomroles)	268
GET: Get All System Roles (getallsystemroles)	270
GET: Get User Roles (getroleofuser)	272
PUT: Grant User Roles (grantries)	274
PUT: Revoke User Roles (revokeroles)	276
FAM/Security Groups	278
PUT: Ensure Security Group (ensuresecuritygroup)	278
GET: Get All Security Groups (getallsecuritygroups)	280
PUT: Remove Security Group (removesecuritygroup)	281

UAA/Agents	282
POST: Create A Lotus Domino Journal Server Agent (createlotusjournalagent).....	283
POST: Create Lotus Domino Mailbox Crawl Server Agent (createlotusmailboxcrawlagent)	285
POST: Create a Server Agent (createserveragent)	287
POST: Create a User Synchronization Mail Server Agent (createusersyncserveragent)	290
DELETE: Delete a Server Agent (deleteserveragent)	293
GET: Get Server Agent Using Function (getagentusingfunction)	294
GET: Get Server Agent Using Name (getagentusingname)	296
GET: Get All Agents of Mail Server (getagentusingidmailserver)	298
PUT: Update A Lotus Domino Journal Server Agent (updatelotusjournalagent)	300
PUT: Update Lotus Domino Mailbox Crawl Server Agent (updatelotusmailboxcrawlagent).....	302
PUT: Update a Server Agent (updateserveragent).....	304
PUT: Update a User Synchronization Mail Server Agent (updateusersyncserveragent).....	306
FAM/Projects	309
POST: Create OneDrive Project (createonedriveproject)	310
POST: Create Project - File Share Projects Only (create).....	314
POST: Create SharePoint Project (createsharepointproject)	318
DELETE: Remove Project (deleteusingid)	324
GET: Get All Projects (getallprojects).....	325
GET: Get Sub-Folders (subfolders).....	326
GET: Get Project Info by ID (getprojectusingid)	328
GET: Get Project Info by Name (getprojectusingname)	329
GET: Get Projects of File Server (getprojectsusingidfilesserver).....	330
GET: Get Projects Using Search (getprojectusingpatternsearch)	331
PUT: Update OneDrive Project (updateonedriveproject)	332
PUT: Update Project - File Share Projects Only (update)	333
PUT: Update SharePoint Project (updatesharepointproject).....	334
FAM/Tags.....	335
DELETE: Delete Tag (deletetag)	336
GET: Get All Tags of a Project (getalltagsofaproject).....	337
GET: Get Tag Using ID (gettagusingid).....	339

GET: Get Tag Using Name (gettagusingname).....	341
POST: Upload Content Spec (uploadcontenttagspec)	343
POST: Upload MetaData Spec (uploadmetadataspec)	345
POST: Upload PII Tags (uploadPIIspec)	348
POST: Upload Tag Definition File (uploadtags).....	350
FAM/Task Sets	352
POST: Add Task Set Entries (addentries)	353
DELETE: Delete Task Set Entry (deleteentry).....	356
GET: Get All Task Set Entries (getallentries)	357
GET: Get All Entries of a Task Set (getallentriesofset)	358
GET: Get All Entries Using Pattern (getallentriesusingpattern).....	359
GET: Get a Task Set Using ID (getusingid).....	360
POST: Update Task Set Entry (updateentry).....	361
UAA/Users	363
PUT: Add User Alias (adduseralias).....	364
POST: Create User (createuser)	366
DELETE: Delete User (deleteuser).....	369
GET: Get All Department Users (getalldepartmentusers)	370
GET: Get All User Aliases (getalluseraliases)	371
GET: Get User Using Address (getuserusingaddress)	373
GET: Get User Using Alias (getuserusingalias)	374
GET: Get User Using External Reference (getuserusingexternalreference)	375
GET: Get User Using Owner (getuserusingowner)	376
GET: Get User Using ZLP ID (getuserusingid).....	377
PUT: Move User to a New Department (moveusertonewdepartment).....	378
PUT: Remove User Alias (removealias).....	379
PUT: Restore Terminated User (restoreterminateduser).....	380
PUT: Terminate User (terminateuser)	381
PUT: Update User Account Information (updateuseraccountinfo).....	382
PUT: Update User Email Address (updateuseremailaddress)	383
PUT: Update User Mail Server Information (updateusermailserverinfo)	384

PUT: Update User Owner (updateuserownerfield) 385

PUT: Update User Sync Status (updateusersyncexclude)..... 386

Introduction to REST API

NOTE: As of the release of this preliminary document, the REST API is in Beta level and this document is in a preliminary state. As a result, the contents of this document and the descriptions of the REST API endpoints it contains are preliminary and subject to change.

The REST API has been added to ZL Unified Archive (UA) for ZL UA 10.1.0 and higher, providing integration of the ZL UA High Performance API Kit for compatibility with REST architecture and Web services. This results in several advantages once fully implemented:

- 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.

In ZL UA 10.3, endpoints have been provided for the following purposes:

- **Discovery/Case Data Sources:** In the ZL Discovery Manager, 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*.

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 7.

- **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 17.

- **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 69.

- **Discovery/Tasks:** The Discovery Manager includes several background tasks you must execute at different times. For example, after running a search to add data to a case, you must the Run Case Importer background task. This imports the data from any new collections and preservations into the case, updates the case content index (if content indexing is enabled), and builds the parallel index.

Use these endpoints to execute the different background tasks or view the status of the background tasks. For more information, see *Discovery/Tasks* on page 76.

- **Discovery/Custodians:** Custodians are targets of an eDiscovery investigation who are likely to possess or 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 81.

- **UAA/Departments:** A department is defined by ZL UA as a hierarchical grouping of users. Each user created in ZL UA is grouped into a department that has been added to the system. The structure of 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 105.

- **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 122.
- **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 137.

- **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 142.

- **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 and update mail servers in ZL UA. For more information, see *UAA/Data Sources* on page 146.

- **FAM/Dashboards:** When using the FAM module, it is important to understand the difference between a project and a dashboard. A project is a set of folders and files that has been grouped together. Projects

are also used in other ZL UA modules. A dashboard is a set of projects that has been grouped together for analysis within the FAM module.

Use these endpoints to create and update dashboards for use in the FAM module. For more information, refer to *FAM/Dashboards* on page 185.

- **FAM/Disposition:** Manage disposition within FAM projects. 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. For more information, see *FAM/Disposition* on page 197.

- **FAM/Reports:** Use these endpoints to generate reports containing information relevant to audit trail data, file and folder analysis data, PII date, remediation tasks, and the disposition status of FAM projects. For more information, see *FAM/Reports* on page 206.
- **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 required background tasks on FAM projects. For more information, see *FAM/Tasks* on page 216.

- **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 227.
- **UAA/Projects:** A project is essentially 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.

Use these endpoints to create and manage projects on various server types. For more information, refer to *UAA Projects* on page 239.

- **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. A case's department can be defined during case setup.

Use these endpoints to grant and revoke roles within the UAA module and ZL UA. For more information, refer to *UAA/Roles* on page 268.

- **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 278.

- **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 282.

- **FAM/Projects:** A project represents a collection of folders and directories on a server that is grouped together for use within the FAM module. Use these endpoints to create and manage file projects. Use these endpoints to create and manage File Share, SharePoint and OneDrive projects: For more information, see *FAM/Projects* on page 309.
- **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 335.

- **FAM/Task Sets:** 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.

You can create task sets to run background tasks simultaneously, on an on-demand basis or at regularly scheduled intervals. Each task set configuration includes a group of background tasks and a list of projects and dashboards that the task set is applicable to. When the task set is executed, all of the background tasks will be executed upon the specified projects and upon all projects included in the specified dashboards. You can create task sets to run background tasks simultaneously, on an on-demand basis or at regularly scheduled intervals. Each task set configuration includes a group of background tasks and a list of projects and dashboards that the task set is applicable to. When the task set is executed, all of the background tasks will be executed upon the specified projects and upon all projects included in the specified dashboards.

Use these endpoints to create and manage task sets. For more information, refer to *FAM/Task Sets* on page 352.

- **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 363.

Accessing the REST API

Access the Rest API 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>

ZL Technologies API Beta OAS3

/ps/api/openapi.json

Provides Data Analytics Services

[Terms of service](#)
[Tech Support - Website](#)
[Send email to Tech Support](#)
[Apache 2.0](#)

Servers

/ps/api

Authorize

Filter by tag

Authentication

Discovery/CaseDataSource

Discovery/Cases

Discovery/Privilege

Discovery/Tasks

Discovery/Custodians

UAA/Departments

Discovery/Roles

Figure 1: ZL UA Rest API

Discovery/CaseDataSource

In the ZL Discovery Manager, 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*.

When you save the search to the case, you can specify options to automatically re-run the search later. New search results returned during each re-run 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:

- **POST: Create Child Department (*createchilddepartment*):** 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 Case Data Source Using Name (*getcasedatasourceusingname*):** View the details 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.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
dataSourceId	Integer	Specify the ID of the data source you want to view. For information on retrieving data source IDs, refer to <i>GET: Get Case Data Source Using Name (getcasedatasourceusingname)</i> on page 11.
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 Codes

- 200: Successful
- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

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 for each run:		
id	Integer	The data source run ID.
caseDataSourceId	Integer	The ID of the data source.
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.
searchPID	String	The PID 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/time 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.
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 runs of a data source.

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. For information on retrieving data source IDs, refer to <i>GET: Get Case Data Source Using Name (getcasedatasourceusingname)</i> on page 11.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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 dateStart and dateExpiry .
dateExpiry	String	The date and time that the data source run schedule ends.

GET: Get Case Data Source Using Name (getcasedatasourceusingname)

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

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the data source you want to view. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
dataSourceName	String	Specify the name of the data source you want to view. This is the name assigned to the search when it was initially created in Discovery Manager.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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.

Schema Field	Type	Description
parentId	Integer	If the data source is a copy, this represents the ID of the parent data source. This defaults to -1 otherwise.
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 Latest Case Data Source Run (getlatestcasedatasourcerun)

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

Request Parameters

Parameter	Type	Description
dataSourceId	Integer	Specify the ID of the data source you want to view. Information for the most recent data source run for this source will be returned. For information on retrieving data source IDs, refer to <i>GET: Get Case Data Source Using Name (getcasedatasourceusingname)</i> on page 11.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
id	Integer	The data source run ID.
caseDataSourceId	Integer	The ID of the data source.
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 PID 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	

Schema Field	Type	Description
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/time 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.
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 runs of a particular data source.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
dataSourceId	Integer	Specify the ID of the data source to be scheduled. For information on retrieving data source IDs, refer to <i>GET: Get Case Data Source Using Name (getcasedatasourceusingname)</i> on page 11.
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 dateStart and dateExpiry .
dateExpiry	String	The date and time to end the data source runs.

Response Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable.
500: Internal Server Error

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 dateStart and dateExpiry .
dateExpiry	String	The date and time to end that the future data source runs will end.

DELETE: Unschedule Data Source (unscheduleDataSource)

Delete the scheduling for a data source, so that no further runs are executed for the data source.

Request Parameters

Parameter	Type	Description
dataSourceId	Integer	Specify the ID of the data source to be updated. For information on retrieving data source IDs, refer to <i>GET: Get Case Data Source Using Name (getcasedatasourceusingname)</i> on page 11.

Request Body Schema Fields

None.

Response Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not acceptable.
500: Internal Server Error

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 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)*
- *PUT: Apply Case Tags (applyCaseTags)*

Apply tags to a case item.

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. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
caseItemId	Integer	Specify the ID of the case item you want to apply tags to.

Schema Field	Type	Description
tagIds	Integer	<p>An array of tag IDs identifying the tags you want to apply to the case item. Specify the tag IDs as a comma-separated list. For example:</p> <pre>"tagIds": [0, 1, 2, 3]</pre> <p>For information on obtaining case IDs, refer to <i>GET: Get All Case Tags (getallcasetags)</i> on page 43.</p>

Response Codes

206: Partial Content
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

Response Schema Fields

A string indicating whether the tags were applied successfully.

- POST: Create a Case (createcase)
- POST: Create a Schema (createSchema)

Create a schema. Custom metadata 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.

Request Parameters

Parameter	Type	Description
schemaName	String	The name of the schema.

Request Body Schema Fields

None.

Response Codes

201: Created
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
name	String	The name of the schema.
displayName	String	The display name of the schema.
schemald	Integer	The ID assigned to the schema.
nSchemaFields	Integer	The number of fields the schema contains.

- DELETE: Delete Case (deleteCase)
- *DELETE: Delete Tag (deleteTag)*
- GET: Get All Case Data Sources (getallcasedatasources)
- GET: Get Case Info (getallcaseusingname)
- GET: Get Case Info (getallcaseusingnamepattern)
- GET: Get All Case Tags (getallcasetags)
- GET: Get All Case User Privileges (getallcaseuserprivileges)
- GET: Get All Child Tags (getallchildtags)
- GET: Get All Cases Using Domain ID (getAllCasesUsingDomain)
- GET: Get List of All Schema (getAllSchema)
- Retrieve a list of all metadata schema that has been added. Custom metadata 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.

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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
SchemaName	String	The name of the schema.
Schema Fields List: 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).

Schema Field	Type	Description
Mandatory	Boolean	Indicates whether the field is mandatory (True) or not.

- GET: Get Case Data Source Using ID (*getcasedatasourceusingsrcid*)
- GET: Get Case Using ID (*getcaseusingid*)
- GET: Get Case Item (*getcaseitem*)
- GET: Get Case Schema Using Case ID and Field Item Schema (*getCaseSchema*)
- GET: Get Case Schema Metadata (*GetCaseSchemaMetaData*)

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

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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		

Schema Field	Type	Description
additionalProp2		
additionalProp3		

PUT: Update Case Item Schema (updatecaseitemschema)

Update the schema.....

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
schemaName	String	Enter the name of the schema.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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 a schema. Custom metadata 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.

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
schemaName	String	Enter the name of the schema.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

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)

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

Request Parameters

Schema Field	Type	Description
idCase	Integer	Enter the case ID. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
FieldData Entries: Includes an entry for each field that belongs to the specified file schema. Each entry includes the following information:		
additionalProp1		
additionalProp2		
additionalProp3		

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

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		
additionalProp2		

Schema Field	Type	Description
additionalProp3		

- 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.

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. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
parentTagId	Integer	The ID of the parent tag that the new tag will be grouped under in the tag hierarchy. For information on obtaining tag IDs, refer to <i>GET: Get All Case Tags (getallcasetags)</i> on page 43.

Response Codes

201: Created
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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.

Schema Field	Type	Description
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.
root_node	Boolean	Indicates the tag is the root tag.
read_only	Boolean	Indicates the tag is read-only.
enduser_tag	Boolean	Indicates that the tag can be applied manually.
auto_tag	Boolean	Indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag	Boolean	Indicates the tag is a mutually exclusive tag.
tag_32	Boolean	Indicates the tag is a PII tag.
tag_64	Boolean	Indicates 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.

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. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
caseItemId	Integer	Specify the ID of the case item you want to apply tags to.
tagIds	Integer	An array of tag IDs identifying the tags you want to apply to the case item. Specify the tag IDs as a comma-separated list. For example: <pre>"tagIds": [0,1,2,3]</pre> For information on obtaining case IDs, refer to <i>GET: Get All Case Tags (getallcasetags)</i> on page 43.

Response Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

Response Schema Fields

A string indicating whether the tags were applied successfully.

POST: Create a Case (createcase)

Create a new case.

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 privilege's role, he/she 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: <ul style="list-style-type: none">1: GENERAL LITIGATION2: ANTITRUST3: LABOR AND EMPLOYMENT4: BREACH OF CONTRACT5: IP/PATENT6: REAL ESTATE7: GOVERNMENT INVESTIGATION8: INTERNAL MATTER
state	Integer	Specify the current state of the case. This field value does not affect the behavior of the case; it is for informational purposes only.
fileDate	String	Specify the filing date for the case.

Schema Field	Type	Description
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 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 Codes

201: Created
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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,

Schema Field	Type	Description
lastUpdate	String	The date and time the case was last updated,
state	Integer	The current state of the case.
ownerZipUserId	Integer	The ZLP user ID of the user who created the case.
journalDomainId	Integer	The case's journal domain ID.
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

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.

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case the custodian should be added to. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
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 omit these fields.		
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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.
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.

POST: Create a Schema (createSchema)

Create a schema. Custom metadata 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.

Request Parameters

Parameter	Type	Description
schemaName	String	The name of the schema.

Request Body Schema Fields

None.

Response Codes

201: Created
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
name	String	The name of the schema.
displayName	String	The display name of the schema.
schemald	Integer	The ID assigned to the schema.
nSchemaFields	Integer	The number of fields the schema contains.

DELETE: Delete Case (deleteCase)

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

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case to be deleted. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.

Request Body Schema Fields

None.

Response Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.

Request Parameters

Parameter	Type	Description
tagId	Integer	Specify the ID of the tag to be deleted. For information on obtaining tag IDs, refer to <i>GET: Get All Case Tags (getallcasetags)</i> on page 43.
caseId	Integer	Specify the ID of the case containing the tag to be deleted. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.

Request Body Schema Fields

None.

Response Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case whose data sources you want to view. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
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 Codes

200: Successful
206: Partial Content
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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.

Schema Field	Type	Description
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.
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 (getallcaseusingname)

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

Request Parameters

Parameter	Type	Description
caseName	String	The name of the case you want to view.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

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.
ownerZipUserId	Integer	The ZLP user ID of the user who created the case.
journalDomainId	Integer	Reserved.

GET: Get Case Info (getallcaseusingnamepattern)

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

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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

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	Reserved.

GET: Get All Case Tags (getallcasetags)

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

Request Parameters

Parameter	Type	Description
caseId	Integer	The ID of the case containing the tags you want to view. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
stTagPurpose	String	Specify the purpose of the tags you want to view (e.g., Review, AutoProcess, etc). The endpoint will return information for this tag and the child tags that have been added beneath it.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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.

Schema Field	Type	Description
root_node	Boolean	Indicates the tag is the root tag.
read_only	Boolean	Indicates the tag is read-only.
enduser_tag	Boolean	Indicates that the tag can be applied manually.
auto_tag	Boolean	Indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag	Boolean	Indicates the tag is a mutually exclusive tag.
tag_32	Boolean	Indicates the tag is a PII tag.
tag_64	Boolean	Indicates 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.
tagFlags	Array	An array of Boolean values indicating the status of various tag attributes.
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.

Schema Field	Type	Description
tag_32	Boolean	Indicates the child tag is a PII tag.
tag_64	Boolean	Indicates the child tag is a content tag.
description	String	A description of the child tag.
createDate	String	The date and time that the child tag was created.

GET: Get All Case User Privileges (getallcaseuserprivileges)

Retrieve information for the user privileges that have been assigned within a particular case. Specify the case by its ID.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case whose user privileges you want to view. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
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 Codes

200: Successful
206: Partial Content
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
User: The response includes an entry for user that has access to the specified case. Each entry includes the following fields:		
zlpUserID	Integer	The ZLP ID of the user.
fullName	String	The name of the user.

Schema Field	Type	Description
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.

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.

Request Parameters

Parameter	Type	Description
caseId	Integer	The ID of the case that the tags you want to view were added to. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
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 Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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.

Schema Field	Type	Description
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	Indicates the tag is the root tag.
read_only	Boolean	Indicates the tag is read-only.
enduser_tag	Boolean	Indicates that the tag can be applied manually.
auto_tag	Boolean	Indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag	Boolean	Indicates the tag is a mutually exclusive tag.
tag_32	Boolean	Indicates the tag is a PII tag.
tag_64	Boolean	Indicates the tag is a content tag.
description	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.

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 Codes

206: Partial Content

400: Bad Request

401: Unauthorized

403: Forbidden

404: Not Found

406: Not Applicable

500: Internal Server Error

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.

Schema Field	Type	Description
ownerZipUserId	Integer	The ZLP user ID of the user who created the case.
journalDomainId	Integer	Reserved.

GET: Get List of All Schema (getAllSchema)

Retrieve a list of all metadata schema that has been added. Custom metadata 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.

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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
SchemaName	String	The name of the schema.
Schema Fields List: 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.

Schema Field	Type	Description
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.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
srcid	Integer	Specify the ID of the data sources you want to view. For information on obtaining data source IDs, refer to <i>GET: Get All Case Data Sources (getallcasedatasources)</i> on page 39.

Response Codes

200: Successful

400: Bad Request

401: Unauthorized

403: Forbidden

404: Not Found

406: Not Applicable

500: Internal Server Error

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.
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.

Schema Field	Type	Description
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.

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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	Reserved.

GET: Get Case Item (getcaseitem)

Retrieve information for an item that has been added to a case. Specify the case and case item by their ID values.

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
caseItemId	Integer	Enter the case item ID.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
id	Integer	The case item ID.
caseId	Integer	The case ID.
type	String	The type of item.
referenceItemId	String	Reserved.
subject	String	If the item is an e-mail message, this is the e-mail message's subject.
flags		
long	Integer	Reserved/

Schema Field	Type	Description
allFlagDesc	String	Reserved.
size	Integer	The size of the item, in bytes.
dateCreate	String	The date and time the item was created.
dateUpdate	String	The date and time the item was last updated,
dateImport	String	The date and time the item was imported into ZL UA.
tagId	Integer	A list of tag IDs. These are the tags that have been applied to the item.
runIds	Integer	A list of data source run IDs. These are the data source runs where the item was found in the search.
srcIds	Integer	A list of data source IDs. These are the data sources that include the item.
tagType	Integer	Reserved.

**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.

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
fltemSchema	Boolean	Select either True or False.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

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).

Schema Field	Type	Description
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.

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

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		
additionalProp2		
additionalProp3		

PUT: Update Case Item Schema (updatecaseitemschema)

Update the schema.....

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
schemaName	String	Enter the name of the schema.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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 a schema. Custom metadata 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.

Request Parameters

Parameter	Type	Description
caseId	Integer	Enter the case ID. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
schemaName	String	Enter the name of the schema.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

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)

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

Request Parameters

Schema Field	Type	Description
idCase	Integer	Enter the case ID. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
FieldData Entries: Includes an entry for each field that belongs to the specified file schema. Each entry includes the following information:		
additionalProp1		
additionalProp2		
additionalProp3		

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

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		
additionalProp2		

Schema Field	Type	Description
additionalProp3		

PUT: Update Tag (updatetag)

Update a tag's configuration.

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. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
tagId	Integer	The ID of the tag to be updated. For information on obtaining tag IDs, refer to <i>GET: Get All Case Tags (getallcasetags)</i> on page 43.

Response Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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.

Schema Field	Type	Description
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	Indicates the tag is the root tag.
read_only	Boolean	Indicates the tag is read-only.
enduser_tag	Boolean	Indicates that the tag can be applied manually.
auto_tag	Boolean	Indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag	Boolean	Indicates the tag is a mutually exclusive tag.
tag_32	Boolean	Indicates the tag is a PII tag.
tag_64	Boolean	Indicates 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 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).

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 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 (getuserprivileges)*
- *PUT: Update User Privileges (updateuserprivileges)*

DELTE: Delete All Privileges Within a Case (deleteallprivileges)

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

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. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.

Request Body Schema Fields

None.

Response Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

Response Schema Fields

A string indicating whether the user privileges were removed 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.

Request Parameters

Parameter	Type	Description
idZipUser	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. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.

Request Body Schema Fields

None.

Response Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

Response Schema Fields

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

GET: Get User Privileges (getuserprivileges)

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.

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. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.

Response Codes

- 200: Successful
- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Applicable
- 500: Internal Server Error

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.

Schema Field	Type	Description
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.

PUT: Update User Privileges (updateuserprivileges)

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.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
reviewPrivileges	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.
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. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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/Tasks

The Discovery Manager includes several background tasks you must execute at different times. For example, after running a search to add data to a case, you must the Run Case Importer background task. This imports the data from any new collections and preservations into the case, updates the case content index (if content indexing is enabled), and builds the parallel index.

The following sections describe the endpoints available within Discovery/Tasks. Use these endpoints to execute the different background tasks or view the status of the background tasks:

- *POST: Run Background Task*
- *GET: Get Task Status (gettaskstatus)*

POST: Run Background Task

This section describes the request and response data for the POST endpoints available under Discovery/Tasks. Although there is a separate endpoint provided for each background task, the request and response data is the same for each one. The following table lists the available POST endpoints and describes the background tasks they can be used to execute. Descriptions of the request and response data follow.

Endpoint Name	Description
leagalholdimport	Use these background tasks to import items into a case and build the case parallel index. Generally, it is not necessary to run these background tasks separately, as the Run Case Importer background tasks added in ZL UA 9.2 accomplishes all these objectives. For more information, refer to the <i>Background Tasks and Indexing</i> chapter of the <i>ZL Discovery Manager Guide</i> .
buildcaseitemparallel	
buildlegalholdparallel	
caseitemimporter	
runcaseimporter	Execute the Run Case Importer background task. Execute this background task after creating a collection or a preservation via ECA search (or re-running a previously saved ECA search), you must execute the Run Case Importer background task. This imports the data from any new collections or preservations into the case, updates the case content index (if content indexing is enabled), and builds the parallel index.
runinplacetransport	Execute the Run In-Place Transport background task. This moves any in-place data (MIP files) returned by a search into the ZL Vault. Because MIP files have not been physically added to the ZL Vault, they cannot be tagged or placed on legal hold. You must first transport the MIP files into the ZL Vault with the Run In-Place Transport task (in addition to running the other background tasks required when adding data to a case).
runsavedsearches	Execute the Run Saved Searches background task. Use this background task to re-run all searches that have been saved to the case.
runtagger	Execute the Run Tagger background task. This should be executed after a tag specification rule is uploaded to a case. The Run Tagger task will automatically apply the tag to documents that contain the keyword or phrase within the specified scope of the message.
updateContentIndex	Execute the Update Content Index background task. This updates the content index within the task. This updates the case's content index, which is used to enable content-based searching of the items added to the case.

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case you want to run the background task on. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
stBuildMode	String	Only applicable to the buildcaseitemparallel and buildlegalholdparallel endpoints. Specify the parallel index task mode. Generally, you should choose Incremental as the parallel index task mode. Running the task in Full_Build mode will re-build the index entirely, which is comparatively time-consuming and only required in very specific scenarios.

Request Body Schema Fields

None.

Response Codes

202: Accepted
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

Response Schema Fields

A string indicating whether the background task was initiated successfully.

GET: Get Task Status (gettaskstatus)

Obtain the status of a background task that is being run on a case.

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case the task was executed upon. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
stTask	String	Select the task whose status you want to obtain.

Request Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

Parameter	Type	Description
message	String	Information regarding the background task status.
startTime	String	The date and time the background task began.
taskName	String	The name of the background task.
pid	String	The server host name.
currentPhases		
totalPhases		
doneDate	String	The date and time the task completed.

Parameter	Type	Description
status	String	Each status entry includes additional information about the status of the background task.

Discovery/Custodians

Custodians are targets of an eDiscovery investigation who are likely to possess or 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 Custodian Preservation Using Custodian ID (getcustodianpreservationusingid)*
- *GET: Get All Custodians Using ZLP User ID (getallcustodiansusingzlpid)*
- *GET: Get All Custodian Preservations*
- *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.

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case the custodian should be added to. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
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 omit these fields.		
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 Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.
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.

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodian to be updated. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
custodianId	Integer	Specify the ID of the custodian to be updated. For information on obtaining custodian IDs, refer to <i>GET: Get All Custodians (getallcustodians)</i> on page 88.
aliasType	Integer	Specify the alias type: <ul style="list-style-type: none">• 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 Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found

406: Not Acceptable
500: Internal Server Error

Response Schema Fields

Parameter	Type	Description
caseld	Integer	The ID of the case containing the custodian that has been updated. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
custodianId	Integer	The ID of the custodian that has been updated.
aliasType	Integer	The alias type: <ul style="list-style-type: none">• 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.

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodian to be viewed. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
custodianId	Integer	Specify the ID of the custodian to be viewed. For information on obtaining custodian IDs, refer to <i>GET: Get All Custodians (getallcustodians)</i> on page 88.
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 Codes

200: Successful
206: Partial Content
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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:		

Parameter	Type	Description
caseId	Integer	The case ID.
custodianId	Integer	The custodian ID.
aliasType	Integer	The alias type: <ul style="list-style-type: none">• 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.

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodians to be viewed. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
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 Codes

- 200: Successful
- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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	Specify the ID of the case the custodian should be added to.

Parameter	Type	Description
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.
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.

Request Parameters

Parameter	Type	Description
idCustodianPreservation	Integer	Specify the preservation ID. For information on obtaining preservation IDs, refer to <i>GET: Get All Custodian Preservations (getallcustodianpreservation)</i> on page 94.

Request Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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).

Parameter	Type	Description
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 All Custodians Using ZLP User ID (getallcutodiansusingzlpid)

Retrieve information for a custodian. Specify the custodian by his or her user ID.

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to grant roles to.
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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.
zlpUserId	Integer	The ZLP user ID of the custodian.
isTerminated	Boolean	If True, the user has been terminated.

Parameter	Type	Description
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.

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 all custodian preservations that have been added to a case.

Request Parameters

Parameter	Type	Description
idCase	Integer	Specify the ID of the case whose custodian preservations you want to view. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.

Parameter	Type	Description
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.
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.

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodian to be viewed. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
address	String	Specify the custodian's primary email address.

Request Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.

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 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.

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodian to be viewed. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
custodianId	Integer	Specify the custodian ID. For information on obtaining custodian IDs, refer to <i>GET: Get All Custodians (getallcustodians)</i> on page 88.

Request Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.

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.

DELETE: Delete Custodian (delete)

Remove a custodian from a case.

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodian to be removed. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
custodianId	Integer	Specify the ID of the custodian to be removed. For information on obtaining custodian IDs, refer to <i>GET: Get All Custodians (getallcustodians)</i> on page 88.

Request Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

A string indicating whether the custodian was removed successfully.

DELETE: Delete Custodian Alias (deletealias)

Remove a custodian alias.

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case containing the custodian to be updated. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
custodianId	Integer	Specify the ID of the custodian to be updated. For information on obtaining custodian IDs, refer to <i>GET: Get All Custodians (getallcustodians)</i> on page 88.
aliasType	Integer	Specify the alias type: <ul style="list-style-type: none">• 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 Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found

406: Not Acceptable

500: Internal Server Error

Response Schema Fields

A string indicating whether the custodian alias was removed successfully.

PUT: Update Custodian (updatecustodian)

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

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the ID of the case the custodian to be updated belongs to. For information on obtaining case IDs, refer to <i>GET: Get Case Info (getallcaseusingname)</i> on page 41 or <i>GET: Get Case Info (getallcaseusingnamepattern)</i> on page 42.
custodianId	Integer	Specify the custodian ID of the custodian to be updated. For information on obtaining custodian IDs, refer to <i>GET: Get All Custodians (getallcustodians)</i> on page 88.
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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.

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

UAA/Departments

A department is defined by ZL UA as a hierarchical grouping of users. Each user created in ZL UA is grouped into a department that has been added to the system. The structure of the departmental hierarchy 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/Mail Servers endpoints available in the REST API. Use these endpoints to create and update servers:

- *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 Parent Information By ID (getparentdepartment)*: Retrieve the configuration of department's parent department.
- *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.

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	Reserved.

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 Codes

201: Created
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Applicable
500: Internal Server Error

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 department's 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 name assigned to the department.

Schema Field	Type	Description
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	Reserved.
miscField1 miscField2	String	Any miscellaneous information entered when the department was created.
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.

Request Parameters

Parameter	Type	Description
deptId	Integer	Specify the ID of the department to be deleted. UAA/Departments includes several GET endpoints you can use to retrieve department IDs, as described in the following sections.

Request Body Schema Fields

None.

Response Codes

- 200: OK
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not acceptable.
- 500: Internal Server Error

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.

Request Parameters

Parameter	Type	Description
deptId	Integer	Specify the ID of the department whose child department configurations you want to view.
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 Codes

206: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable.
500: Internal Server Error

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 106.

GET: Get All Departments Using Name (getalldeptsusingnamepattern)

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

Request Parameters

Parameter	Type	Description
pattern	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 Codes

206: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable.
500: Internal Server Error

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 106.

GET: Get Users of a Department (getallusers)

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

Request Parameters

Parameter	Type	Description
departmentName	Integer	Specify the name of the department whose users you want to view.
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 Codes

- 206: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable.
- 500: Internal Server Error

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 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.

Schema Field	Type	Description
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.
deptName	String	The department the user is assigned to.
altReviewDepts	String	The Alternate Review Department(s) 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	
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.
dateHired	String	The date and time the user was hired.
dateTerminated	String	The date and time the user was terminated, if applicable.
terminated	Boolean	Indictes whether the user has been Terminated (True) or not.
dateIsterStart	String	These fields indicate that date and time that the last user synchronization process started and ended, and the date
dateIsterEnd	String	

Schema Field	Type	Description
dateIsterUpdate	String	and time that the user’s information was updated during synchronization.
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.

Request Parameters

Parameter	Type	Description
deptId	Integer	Enter the ID of the department to view.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable.
500: Internal Server Error

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 106.

GET: Get Department By Name (getdepartmentusingname)

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

Request Parameters

Parameter	Type	Description
deptName	String	Enter the name of the department to view.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable.
500: Internal Server Error

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 106.

GET: Get Department Parent Information By ID (getparentdepartment)

Retrieve the configuration of a parent department. Identify the department whose parent is to be retrieved by its ID.

Request Parameters

Parameter	Type	Description
deptId	Integer	Enter the ID of the department whose parent is to be retrieved.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable.
500: Internal Server Error

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 106.

GET: Get Department Policy Information Using ID (getpolicy)

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

Request Parameters

Parameter	Type	Description
deptId	Integer	Enter the ID of the department to view.
stPolicyType	String	Select the policy you want to view (e.g., Archiving, Stubbing, etc).

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable.
- 500: Internal Server Error

Response Schema Fields

Parameter	Type	Description
id	Integer	The department ID.
name	String	The policy name.
policyType	String	The policy type.
description	String	A description of the policy.
createDate	String	The date and time the policy was created.

PUT: Update Department (updatedepartment)

Update a department's configuration.

Request Parameters

Parameter	Type	Description
deptId	Integer	Specify the ID of the department to be updated. UAA/Departments includes several GET endpoints you can use to retrieve department IDs, as described in the previous sections.

Request Body Schema Fields

Schema Field	Type	Description
name	String	Type 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 in the Domain name text box to create a new domain. Leave the Domain name text box 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	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.
externalReference	String	Use this optional field to store any unique IDs used outside of ZL UA.
deptTags	String	Reserved.

Schema Field	Type	Description
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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.

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 that should be assigned as its parent ID as the parentId value. UAA/Departments includes several GET endpoints you can use to retrieve department IDs, as described in the previous sections.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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 106.

Discovery/Reports

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

- **Download Audit Trail Report:** This report lists the actions taken upon a case and indicates which user performed each action.
- **Download Preservation Hits Report:** This 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:** This report presents details of all the custodians added to the case within their assigned departmental scope.
- **Download Export Task Report:** This report presents detailed information on every export action taken within the case.
- **Download Global Case Report:** This 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:** This 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:** This 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:** This report summarizes the collections, preservations and analysis searches that have been added to a case.
- **Download Custodian Preservation Report:** This report summarizes the custodian preservations that have been added to cases.
- **Download Global Survey Report:** This report is a consolidated report of all the surveys across all the cases accessible to you in ZL Discovery Manager.
- **Download 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.
- **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:** This report summarizes the search terms used to conduct a search.
- **Download Workflow Audit Term Report:** This 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 taken upon the case, and indicates which user performed each action. The report will be downloaded to the **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the case ID.
action subAction	String	Reserved.
dateMode	String	Reserved.
startDate endDate	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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 500: Internal Server Error

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 **Downloads** directory.

Request Parameters

Parameter	Type	Description
caseId	Integer	The ID of the case the preservation belongs to.
storeId	Integer	The ID of the search store that was used to create the preservation (i.e., the store that was searched).

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 500: Internal Server Error

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 **Downloads** directory.

Request Parameters

Parameter	Type	Description
caseId	Integer	Specify the case ID.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 500: Internal Server Error

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 **Downloads** directory.

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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 500: Internal Server Error

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 **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
500: Internal Server Error

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 **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
500: Internal Server Error

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 **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
500: Internal Server Error

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 **Downloads** directory.

Request Parameters

Parameter	Type	Description
caseld	Integer	Specify the case ID.
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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 500: Internal Server Error

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 **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
caseMode	Integer	Specify the case mode.
caseIds	Integer	An array of case IDs identifying 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]
srchField	String	Reserved.
srchPattern	String	Enter a search pattern to locate the custodians of interest.
dateEnd	String	Specify the end date for the report.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 500: Internal Server Error

Response Schema Fields

None.

POST: Download Global Survey Report (downloadGlobalSurveyReport)

Download a global survey report, a consolidated report of all the surveys across all the cases accessible to you in ZL Discovery Manager. The report will be downloaded to the **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
caseMode	Integer	Specify the case mode.
type	String	Enter Surveys to view the details of the surveys across your accessible cases in ZL Discovery Manager.
caseId	Integer	Specify the case ID.
surveyIds	Integer	<div>An array of survey IDs you want to view. Specify the case IDs as a comma-separated list. For example: <pre>"surveyIds": [0,1,2,3]</pre> Leave this blank to view all survey data for the specified case.</div>

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 500: Internal Server Error

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 **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
caseId	Integer	The ID of the case that you want to search.
searchStoreId	Integer	The ID of the search store.
fields	String	An array of fields to search. Specify the field names as a comma-separated list. For example: "fields": [0,1,2,3] 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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 500: Internal Server Error

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 **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
500: Internal Server Error

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 **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
caseId	Integer	Specify the case ID.
searchID	Integer	Specify the search (data source) ID. For information on retrieving data source IDs, refer to <i>GET: Get Case Data Source Using Name (getcasedatasourceusingname)</i> on page 11.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
500: Internal Server Error

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 **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
taskType subTaskType		Reserved.
idCase	Integer	Specify the case ID.
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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 500: Internal Server Error

Response Schema Fields

None.

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.

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. A case's department can be 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 (granroles)

Grant roles to a user.

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.
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 obtaining role IDs, refer to the following sections: <ul style="list-style-type: none">• <i>GET: Get All Custom Roles</i> on page 268• <i>GET: Get All System Roles</i> on page 270
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: <ul style="list-style-type: none">• Global: Global• InclRecur: On selected departments recursively• Incl: On selected departments only
allScopeDomainIds	Integer	An array of domain IDs identifying the departments the role is applicable to for roles that are only granted on selected departments. These can be retrieved from the ArchiveServer Department Table. Specify the domain IDs as a comma-separated list. For example: <pre>"AllScopeDomainIds": [0,1,2,3]</pre>

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden

404: Not Found
406: Not Acceptable
500: Internal Server Error

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.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zipUserId	Integer	Specify the ZL user ID of the user you want to modify.
roles	Integer	<p>An array of role IDs identifying the roles to be revoked. Specify the role IDs as a comma-separated list. For example:</p> <pre>"Roles": [0, 1, 2, 3]</pre> <p>For information on obtaining role IDs, refer to the following sections:</p> <ul style="list-style-type: none">• <i>GET: Get All Custom Roles</i> on page 268• <i>GET: Get All System Roles</i> on page 270

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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/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. A case's department can be defined during case setup.

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.

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.
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 obtaining role IDs, refer to the following sections: <ul style="list-style-type: none">• <i>GET: Get All Custom Roles</i> on page 268• <i>GET: Get All System Roles</i> on page 270
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: <ul style="list-style-type: none">• Global: Global• InclRecur: On selected departments recursively• Incl: On selected departments only

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

Response Codes

- 200: Successful
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.

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.
roles	Integer	<p>An array of role IDs identifying the roles to be revoked. Specify the role IDs as a comma-separated list. For example:</p> <pre>"Roles": [0, 1, 2, 3]</pre> <p>For information on obtaining role IDs, refer to the following sections:</p> <ul style="list-style-type: none"><i>GET: Get All Custom Roles</i> on page 268<i>GET: Get All System Roles</i> on page 270

Response Codes

- 200: Successful
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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 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.

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 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)*
- *GET: Get Server Using ID (getmailserverusingid)*
- *GET: Get Server Using Name (getmailserverusingname)*
- *PUT: Update an EWS Mail Server (updateewsmailserver)*
- *PUT: Update File Server (updatefileservice)*
- *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 An EWS Mail Server (createmailserver/ews)

Create a Microsoft EWS mail server.

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. For detailed instructions on this and other aspects of MS Teams server administration, refer to the <i>ZL UA MS Teams Administrator's Guide</i> .
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	Select this check box 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. For detailed instructions on this and other aspects of MS Teams server administration, refer to the <i>ZL UA MS Teams Administrator's Guide</i> .

Response Codes

201: Created

400: Bad Request

401: Unauthorized

403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.
discovery Name	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.

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 Codes

- 201: Created
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.

Schema Field	Type	Description
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.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	This field is not required.
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 has to be crawled.
consumerCertFile	String	<p>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.</p> <p>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:</p> <p>D:\file.txt</p> <p>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:</p> <p>\\server2\D\$\file.txt</p>
scopes	String	Specify the service account scopes. Use this to set the change level of Google Drive API permission that can be assigned to ZL.

Response Codes

201: Created

400: Bad Request

401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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 Google Mail Server (creategooglemailserver)

Create a Google Mail server.

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	<p>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.</p> <p>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:</p> <p>D:\file.txt</p> <p>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:</p> <p>\\server2\D\$\file.txt</p>

Response Codes

- 201: Created
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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 Lotus Domino Mail Server (createlotusdominomailserver)

Create a Lotus Domino mail server.

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		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		Type the password of the administrative service account you created in Notes.
remoteRouterHost		Type the name of the remote router if there is no local router.
discoveryMethod		Enter the name of the Discovery object to use with the server. This must be created with the UAA module.

Response Codes

201: Created
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
id	String	The server ID.

Schema Field	Type	Description
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 an Exchange Mail Server
(createmailserver/microsoftexchange)**

Create a Microsoft Exchange mail server.

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 Codes

- 201: Created
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.
discovery Name	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.

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. For more information, refer to the <i>ZL UA OneDrive Archiving Administrator's Guide</i> .

Response Codes

201: Created
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.
discovery Name	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.

POST: Create a SharePoint Server (createsharepointserver)

Create a SharePoint server.

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 obtained 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. For more information, refer to the <i>ZL UA SharePoint Archiving Administrator's Guide</i> .
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 Codes

201: Created
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.
discovery Name	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.

Request Parameters

Parameter	Type	Description
serverId	Integer	Specify the ID of the server to be deleted. You can obtain server IDs with the Get > GetAllMailServers endpoint, as described in <i>GET: Get All Servers</i> on page 164.

Request Body Schema Fields

None.

Response Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

Response Schema Fields

A string indicating whether the server was deleted successfully.

GET: Get All Servers (getallmailservers)

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

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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.

Schema Field	Type	Description
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 ID (getmailserverusingid)

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

Request Parameters

Parameter	Type	Description
serverId	Integer	Specify the ID of the server to be retrieved. You can obtain server IDs with the Get > GetAllMailServers endpoint, as described in <i>GET: Get All Servers</i> on page 164.

Request Body Schema Fields

None.

Response Codes

200: OK

400: Bad Request

401: Unauthorized

403: Forbidden

404: Not Found

406: Not Acceptable

500: Internal Server Error

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.

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

GET: Get Server Using Name (getmailserverusingname)

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

Request Parameters

Parameter	Type	Description
serverName	String	Specify the name of the server to be retrieved.

Request Body Schema Fields

None.

Response Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.

PUT: Update an EWS Mail Server (updateewsmailserver)

Update the configuration of a Microsoft EWS mail server.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the Microsoft EWS server to be updated.
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. For detailed instructions on this and other aspects of MS Teams server administration, refer to the <i>ZL UA MS Teams Administrator's Guide</i> .

Response Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.
discovery Name	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.

Request Parameters

Parameter	Type	Description
serverId	Integer	Specify the ID of the file server to be updated. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.

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 Codes

- 200: OK
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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

Schema Field	Type	Description
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.

PUT: Update a Google Drive Server (updategoogledriveserver)

Update a Google Drive server's configuration.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the Google Drive server to be updated.
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 has to be crawled.
consumerCertFile	String	<p>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.</p> <p>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:</p> <p>D:\file.txt</p> <p>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:</p> <p>\\server2\D\$\file.txt</p>
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 Codes

200: OK

400: Bad Request

401: Unauthorized

403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the Google Mail server to be updated.
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	<p>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.</p> <p>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:</p> <p>D:\file.txt</p> <p>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:</p> <p>\\server2\D\$\file.txt</p>

Response Codes

- 200: OK
- 400: Bad Request

401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the server to be updated.
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 Codes

200: OK

400: Bad Request

401: Unauthorized

403: Forbidden

404: Not Found

406: Not Acceptable

500: Internal Server Error

Response Schema Fields

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

Schema Field	Type	Description
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 Exchange Mail Server (updatemailserver/exchange)

Update a Microsoft Exchange mail server's configuration.

Request Parameters

Parameter	Type	Description
serverId	Integer	Specify the ID of the Microsoft Exchange mail server to be updated. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.

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 Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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	Addtiional information about the server.
discovery Name	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.

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. For more information, refer to the <i>ZL UA OneDrive Archiving Administrator’s Guide</i> .

Response Codes

- 200: OK
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.

Schema Field	Type	Description
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.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the SharePoint server to be updated.
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 obtained 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. For more information, refer to the <i>ZL UA SharePoint Archiving Administrator's Guide</i> .
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 Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.
discovery Name	String	The name of the Discovery Settings object on the server.
group	String	The server group.
dateCreate	String	The date the server was created.

FAM/Dashboards

When using the FAM module, it is important to understand the difference between a project and a dashboard:

- A project is a set of folders and files that has been grouped together. Projects are also used in other ZL UA modules.
- A dashboard is a set of projects that has been grouped together for analysis within the FAM module.

The following sections describe the FAM/Dashboards endpoints available in the REST API. Use these endpoints to create and update dashboards:

- **POST: Create Dashboard (createdashboard):** Create a dashboard.
- **DELETE: Delete Dashboard (deletedashboard):** Delete a dashboard.
- **POST: Download Dashboard Report (downloaddashboardreport):** Download the Dashboard Report, an Excel spreadsheet summarizing the contents of the dashboard.
- **GET: Get All Dashboards (getalldashboards):** Retrieve the configuration of all dashboards.
- **GET: Get All Dashboards of a User (getalldashboards):** Retrieve the configuration of all dashboards associated with a specific user.
- **GET: Get All Dashboards of a User Based on Status (getalldashboards):** Retrieve the configuration of all dashboard assigned a specific status.
- **GET: Get Dashboard Using ID (getdashboardusingid):** Retrieve the configuration of a dashboard specified by its ID,
- **GET: Get Default Dashboard (getdefaultdashboard):** Retrieve the configuration of the default dashboard assigned to a particular user.
- **PUT: Update Dashboard (updatedashboard):** Update the configuration of a dashboard.

POST: Create Dashboard (createdashboard)

Create a dashboard.

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
Id	Integer	Specify the dashboard ID.
zlpUserId	Ineger	Specify the ZLP user ID of the user that the dashboard will be available to.
isDefault	Boolean	Set to True if this dashboard should be used as the default dashboard that is displayed when the FAM module opens for the assigned user.

Parameter	Type	Description
name	String	Specify the name of the dashboard.
projectIds	Integer	An array of project IDs identifying the projects that should be added to the dashboard. Specify the project IDs as a comma-separated list. For example: <pre>"projectIds": [0,1,2,3]</pre>
mapParam	String	Not required.

Response Codes

201: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

Parameter	Type	Description
Id	Integer	The dashboard ID.
zlpUserId	Ineger	The ZLP user ID of the user the dashboard is assigned to.
isDefault	Boolean	If True, then this dashboard will be used as the default dashboard that is displayed when the assigned user logs into the FAM module.
name	String	The name of the dashboard.
projectIds	Integer	An array of project IDs identifying the projects that are included in the dashboard. The project IDs will appear as a comma-separated list. For example: <pre>"projectIds": [0,1,2,3]</pre>
mapParam	String	A key including the IDs of the projects included in the dashboard.

DELETE: Delete Dashboard (deletedashboard)

Delete a dashboard.

Request Parameters

Parameter	Type	Description
dashId	Integer	Specify the ID of the dashboard to be deleted.

Request Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

None.

POST: Download Dashboard Report (downloaddashboardreport)

Download the Dashboard Report: an Excel spreadsheet summarizing the contents of the dashboard. The report will be downloaded to the **Downloads** directory. It includes several tabs you can use to view different types of data:

- **Dashboard Summary:** The header indicates how many projects the dashboard includes, the total size of these projects, and the total file and folder count. The Dashboard Summary also includes basic details about each project included in the dashboard, such as the project ID, the project name, the project type, and the number of files included in the project.
- **Project Distribution:** Indicates the total size and file count of each project included in the dashboard.
- **File Extension Distribution:** Indicates how many files of each extension type (e.g. BIN or SETUP) are included in each project.
- **Size Category Distribution:** Indicates how many files of different size categories are included in each project. For example, it lists how many small-sized (10KB-100KB) files there are in the data set, how many medium-sized files (100KB-1MB) there are, and so on.
- **Tag Distribution:** Indicates how many files each tag has been applied to.
- **ACL Distribution:** List each user that is an ACL owner for files within the currently selected dashboard, and indicates how many files the user is ACL owner for.
- **Create Date:** Lists the number of files created and the size of those files for each date on which a file within the currently selected dashboard was created.
- **Last Modified Date:** Lists the number of files modified and the size of those files for each date on which a file within the currently selected dashboard was modified.
- **Last Accessed Date:** Lists the number of files accessed and the size of those files for each date on which a file within the currently selected dashboard was accessed.

Request Parameters

Parameter	Type	Description
dashId	Integer	Specify the ID of the dashboard you want to view in the report.

Request Schema Fields

None.

Response Codes

- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

None.

GET: Get All Dashboards (getalldashboards)

Retrieve a list of the dashboards that have been created.

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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
Dashboards: Includes an entry for each dashboard that is found. The fields included for each dashboard are the same as those required in the request schema when creating the dashboard. They represent the basic configuration of the dashboard. For descriptions of these fields, refer to <i>POST: Create Dashboard</i> on page 185.		

GET: Get All Dashboards of a User (getalldashboards)

Retrieve a list of the dashboards that have been created by a specific user.

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ZLP user ID of the user whose dashboards you want to view.
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 Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
Dashboards: Includes an entry for each dashboard that was created by the specified user. The fields included for each dashboard are the same as those required in the request schema when creating the dashboard. They represent the configuration of the dashboard. For descriptions of these fields, refer to <i>POST: Create Dashboard</i> on page 185.		

GET: Get All Dashboards of a User Based on Status (getalldashboards)

Retrieve a list of the dashboards that are available to a specific user that have a specific status.

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ZLP user ID of the user whose dashboards you want to view.
status	Integer	0: No Status 1: Default Status
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 Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
Dashboards: Includes an entry for each dashboard that is found for the user. The fields included for each dashboard are the same as those required in the request schema when creating the dashboard. They represent the configuration of the dashboard. For descriptions of these fields, refer to <i>POST: Create Dashboard</i> on page 185.		

GET: Get Dashboard Using ID (getdashboardusingid)

Retrieve the configuration of a specific dashboard by specifying its dashboard ID.

Request Parameters

Parameter	Type	Description
dashboardId	Integer	Specify the ID of the dashboard you want to view.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
Dashboards: The fields included for the dashboard are the same as those required in the request schema when creating the dashboard. They represent the configuration of the dashboard. For descriptions of these fields, refer to <i>POST: Create Dashboard</i> on page 185.		

GET: Get Default Dashboard (getdefaultdashboard)

Retrieve the configuration of the default dashboard that is assigned to a particular user.

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ID of the user whose default dashboard you want to view.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
Dashboards: The fields included for the default dashboard are the same as those required in the request schema when creating the dashboard. They represent the configuration of the default dashboard. For descriptions of these fields, refer to <i>POST: Create Dashboard</i> on page 185.		

PUT: Update Dashboard (updatedashboard)

Update the configuration of an existing dashboard.

Request Parameters

None.

Request Schema Fields

Parameter	Type	Description
Id	Integer	Specify the dashboard ID.
zlpUserId	Ineger	Specify the ZLP user ID the dashboard should be assigned to.
isDefault	Boolean	Set to True if this dashboard should be used as the default dashboard that is displayed when the assigned user opens the FAM module.
name	String	Specify the name of the dashboard.
projectIds	Integer	An array of project IDs identifying the projects that should be added to the dashboard. Specify the project IDs as a comma-separated list. For example: <pre>"projectIds": [0,1,2,3]</pre>
mapParam	String	Not required.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

Parameter	Type	Description
ID	Integer	The dashboard ID.
zlpUserId	Ineger	The ZLP user ID of the user the dashboard is assigned to.

Parameter	Type	Description
isDefault	Boolean	If True, then this dashboard will be used as the default dashboard that is displayed when the assigned user opens the FAM module.
name	String	The name of the dashboard.
projectIds	Integer	An array of project IDs identifying the projects that are included in the dashboard. The project IDs will appear as a comma-separated list. For example: <pre>"projectIds": [0,1,2,3]</pre>
mapParam	String	Reserved.

FAM/Disposition

Disposition is the process by which files whose records management lifecycle has expired are deleted and removed from the ZL UA system. 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 (approvedisposition)

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

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be updated. For information on retrieving FAM project names and project IDs, refer to <i>GET: Get All Projects</i> on page 325.
runId	Integer	Specify the Disposition Run ID. You can obtain Disposition Run IDs with the Get Latest Disposition Run endpoint. For more information, refer to <i>GET: Get Latest Disposition Run</i> on page 201.
approvalComment	String	Enter any comments or descriptive information needed for the Disposition Run approval.

Request Body Schema Fields

None.

Response Codes

200: OK
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

A string indicating whether the disposition run was approved successfully.

PUT: Disable Disposition (disabledisposition)

Disable disposition for a project.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be updated. For information on retrieving FAM project names and project IDs, refer to <i>GET: Get All Projects</i> on page 325 .

Request Body Schema Fields

None.

Response Codes

- 200: OK
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

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

PUT: Enable Disposition (enabledisposition)

Enable disposition for a project.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
idProj	Integer	Specify the ID of the project to be updated. For information on retrieving FAM project names and project IDs, refer to <i>GET: Get All Projects</i> on page 325
dateStart dateEnd	String	<p>Specify a date range during which Disposition Runs can 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 End Date blank to run the task indefinitely.</p> <p>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.</p> <p>Specify the dates using the following format: YYYY-MM-DD</p>

Response Codes

- 200: OK
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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 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.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be viewed. For information on retrieving FAM project names and project IDs, refer to <i>GET: Get All Projects</i> on page 325

Request Schema Fields

None.

Response Codes

- 200: Accepted
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
id	Integer	The Disposition Run ID.
idProj	Integer	The project ID.
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.
approvalComment	String	Comments entered by the user who approved the Disposition Run.

Schema Field	Type	Description
dateStart dateEnd	String	The start date and end date for the Disposition Run.
isApproved	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.

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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

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

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 and the Disposition Task can be run next.

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 Codes

206: Partial Content

400: Bad Request

401: Unauthorized

403: Forbidden

404: Not Found

406: Not Acceptable

500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
Project IDs: Lists the project ID for each project that is ready for Disposition.		

POST: Start Disposition (startdisposition)

Execute a Disposition Task on a project. The Disposition Task This deletes the files included in the most recently approved Disposition Approval Request and removes them from the ZL UA system.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be updated. For information on retrieving FAM project names and project IDs, refer to <i>GET: Get All Projects</i> on page 325
runId	Integer	Specify the Disposition Run ID. You can obtain Disposition Run IDs with the Get Latest Disposition Run endpoint. For more information, refer to <i>GET: Get Latest Disposition Run</i> on page 201.

Request Schema Fields

None.

Response Codes

- 200: Accepted
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

A string indicating whether the Disposition Run was executed successfully.

FAM/Reports

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

- **Audit Trail Report:** This lists the actions taken upon the project (e.g., Project Viewed, Search Ran, etc.) and indicates which user performed each action.
- **File Analysis Report:** This lists information for a specified folder, such as the number of sub-folders and files, as well as information related to tagging and retention codes. It also includes details for each project directory path, such as the total size of the directory, the number of files it contains and the retention code that has been applied.
- **File Manifest Report:** This lists the file name, folder name and path of the files included in the data set. The file names and folder paths are presented as hyperlinks which you can click to open the file (if you have access to the server machine). Excel allows a maximum of 30,000 active hyperlinks per document, so large sets of data may not include hyperlinks for every file.
- **PII Report:** This lists the files that included terms matching the PII patterns defined for your installation. This includes the file name, folder name and path of each file that included PII terms. It also includes the specific PII terms the file included, and how many times each term was found.
- **Remediation Report:** Lists remediation actions that have been carried out within a project.
- **File Tree Disposition Report:** This lists the file name, folder name and path of the files that were disposed of during the Disposition Run, as well as other file attributes such as the name, extension and type, creation and last modification date, size, record category, and disposition eligibility date.
- **File Tree Pre-Disposition Report:** This lists the files that are eligible for disposition within the project.

POST: Download Audit Trail Report (downloadaudittrailreport)

Download an audit report for a project. This lists the actions taken upon the project (e.g., Project Viewed, Search Ran, etc) and indicates which user performed each action. The report will be downloaded to the **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
projectId	Integer	Specify the project ID.
startDate endDate	String	The start date and end date for the report (inclusive). Audit trail data for this date range will be included in the report.
actionType subType	String	<p>Specify the action type and sub-type you want to view. Each action type has a group of action sub-types you can view in the audit report.</p> <p>For example, you could set actionType to search and set subtype to ACTION_SAVED_TO_EDISCOVERY to see when search results were saved to the Discovery Manager in the audit report.</p> <p>Refer to the following section, <i>Action Types and Sub-Types</i>, for possible combination values.</p>

Action Types and Sub-Types

Action Type	Available Sub-Types
entry	ACTION_ENTRY_DOWNLOADED ACTION_MANUAL_TAG_APPLIED ACTION_MANUAL_TAG_REMOVED ACTION_MANUAL_RECORD_APPLIED
retentionActions	ACTION_RETENTION_BACKGROUND_TASK ACTION_RETENTION_CREATE_PURGE ACTION_RETENTION_RETIRE_BIGDB ACTION_RETENTION_DOWNLOAD_REPORT ACTION_RETENTION_APPROVAL

Action Type	Available Sub-Types
	ACTION_RETENTION_ABORT ACTION_RETENTION_CREATE_DISPOSITION_RUN
task	ACTION_CRAWL_SUBMITTED ACTION_UPDATE_CONTENT_INDEX_SUBMITTED ACTION_RUN_METADATA_TAGGER_SUBMITTED ACTION_RUN_CONTENT_TAGGER_SUBMITTED ACTION_RUN_PII_TAGGER_SUBMITTED ACTION_RUN_REMEDIATION_SUBMITTED ACTION_UPDATE_TAG_INDEX_SUBMITTED ACTION_RUN_ALL_SEARCHES
privileges	ACTION_PRIVILEGES_UPDATE
search	ACTION_SAVED_SEARCH_CREATED ACTION_SEARCH_RAN ACTION_SEARCH_TAGGED CROSS_PROJECT_SEARCH_RAN CROSS_PROJECT_SEARCH_HITS_VIEWED ACTION_SAVED_TO_EDISCOVERY ACTION_SAVED_SEARCH_DELETED
report	ACTION_SUMMARY_REPORT_DOWNLOADED ACTION_MANIFEST_REPORT_DOWNLOADED ACTION_PII_REPORT_DOWNLOADED ACTION_DBFTE_REPORT_DOWNLOADED
dashboard	ACTION_DASHBOARD_CREATED ACTION_DASHBOARD_UPDATED ACTION_DASHBOARD_DELETED ACTION_RUN_DASHBOARD_UPDATE_SUBMITTED ACTION_DOWNLOAD_DASHBOARD
project	ACTION_PROJECT_CREATED ACTION_PROJECT_UPDATED ACTION_PROJECT_DELETED ACTION_PROJECT_VIEWED
policy	ACTION_SAMPLING_POLICY_CREATED ACTION_SAMPLING_POLICY_UPDATED ACTION_RUN_SAMPLING_SUBMITTED

Action Type	Available Sub-Types
tagconfig	ACTION_METADATA_SPEC_UPLOADED ACTION_CONTENT_SPEC_UPLOADED ACTION_PII_SPEC_UPLOADED ACTION_TAG_CREATED ACTION_TAG_DELETED ACTION_TAGS_UPLOADED ACTION_TAG_ACTION_APPLIED = 1906
audit	ACTION_AUDIT_TRAIL_VIEWED ACTION_AUDIT_TRAIL_DOWNLOADED ACTION_ENTRY_AUDIT_TRAIL_VIEWED
any	N/A

Response Codes

None.

Response Schema Fields

None.

POST: Download File Analysis Report (downloadfileanalysisreport)

Download a File Analysis Report (also known as the Folder Report in the FAM module UI) for a project. This report lists information for a specified folder, such as the number of sub-folders and files, as well as information related to tagging and retention codes. It also includes details for each project directory path, such as the total size of the directory, the number of files it contains and the retention code that has been applied.

The report will be downloaded to the **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
projectId	Integer	Specify the project ID.
adHocKey	String	Not supported in ZL UA 10.3. Set to null value.
dirId	Integer	Specify the directory ID of the folder directory you want to view. For information on retrieving folder directory IDs, refer to <i>GET: Get Sub-Folders (subfolders)</i> on page 259.

Response Codes

None.

Response Schema Fields

None.

POST: Download File Manifest Report (downloadmanifestreport)

Download a File Manifest Report for a project. This report lists the file name, folder name and path of the files included in the data set. The file names and folder paths are presented as hyperlinks which you can click to open the file (if you have access to the server machine). Excel allows a maximum of 30,000 active hyperlinks per document, so large sets of data may not include hyperlinks for every file.

The report will be downloaded to the **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
projectId	Integer	Specify the project ID.
adHocKey	String	Not supported in ZL UA 10.3. Set to null value.
dirId	Integer	Specify the directory ID of the folder you want to view. For information on retrieving folder directory IDs, refer to <i>GET: Get Sub-Folders (subfolders)</i> on page 259.
folder	Array	Specify the directory IDs of the sub-folders you want to view in the report. These should be sub-folders of the folder references by the dirId field. For information on retrieving folder IDs, refer to <i>GET: Get Sub-Folders (subfolders)</i> on page 259.

Response Codes

None.

Response Schema Fields

None.

POST: Download PII Report (downloadpiireport)

Download a PII Report for a project. The PII Report lists the files that included terms matching the PII patterns defined for your installation. This includes the file name, folder name and path of each file that included PII terms. It also includes the specific PII terms the file included, and how many times each term was found.

The report will be downloaded to the **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
projectId	Integer	Specify the project ID.
dirId	Integer	Specify the directory ID of the folder you want to view. For information on retrieving folder directory IDs, refer to <i>GET: Get Sub-Folders (subfolders)</i> on page 259.
adHocKey	String	Specify the directory ID of the folder you want to view. For information on retrieving folder directory IDs, refer to <i>GET: Get Sub-Folders (subfolders)</i> on page 259.
folder	Array	Specify the directory ID of the folder you want to view. For information on retrieving folder IDs, refer to <i>GET: Get Sub-Folders (subfolders)</i> on page 259.

Response Codes

None.

Response Schema Fields

None.

POST: Download Remediation Report (downloadremediationreport)

Download a Remediation Report for a project. 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. The Remediation Report lists remediation actions that have been carried out within a project.

The report will be downloaded to the **Downloads** directory.

Request Parameters

None.

Request Body Schema Fields

Parameter	Type	Description
projectId	Integer	Specify the project ID.
startDate endDate	String	The start date and end date for the report (inclusive). Remediation actions which occurred during this date range will be included in the report.
remediationType	Array	Specify the type of remediation action you want to view in the report: <ul style="list-style-type: none">ArchiveCopyDeleteMoveLeaveQuarantineRecord

Response Codes

None.

Response Schema Fields

None.

POST: Generate File Tree Disposition Report (generatedispositionreport)

Generate a Disposition Report for a Disposition Run executed upon a project. The Disposition Report is an Excel spreadsheet listing the file name, folder name and path of the files that were disposed of during the Disposition Run, as well as other file attributes such as the name, extension and type, creation and last modification date, size, record category, and disposition eligibility date. The **Disposition Status** column indicates whether each file was disposed of successfully (done) or not (done_error).

The report will be downloaded to the **Downloads** directory.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the project ID.
runId	Integer	Specify the Disposition Run ID.
renderer	String	Set to <code>excel</code> to generate the Disposition Report as an Excel spreadsheet, or <code>csv</code> to generate it as a CSV file.
outputDirectory	String	Specify the directory in which the report should be generated. For example: <code>C:\\Users\\adunna\\Downloads\\fam</code>

Request Body Schema Fields

None.

Response Codes

200: Successful

Response Schema Fields

None.

**POST: Generate File Tree Pre-Disposition Report
(generatepredispositionreport)**

Generate a Pre-Disposition Report for a project. The Pre-Disposition Report lists the files that are eligible for disposition within the project. The report will be downloaded to the **Downloads** directory.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the project ID.
renderer	String	Set to <code>excel</code> to generate the Disposition Report as an Excel spreadsheet, or <code>csv</code> to generate it as a CSV file.
outputDirectory	String	Specify the directory in which the report should be generated. For example: <code>C:\\Users\\adunna\\Downloads\\fam</code>

Request Body Schema Fields

None.

Response Codes

200: Successful

Response Schema Fields

None.

FAM/Tasks

The ZL File Analysis and Management module includes background tasks that must be executed after the completion of certain operations. 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.

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

- *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.
- *GET: Get Task Status (getTaskStatus)*: Obtain the current status of a task.
- *POST: Run All Searches (runallsearches)*: In the FAM module, single project searches include an option to save a search so that you can access it later. Unsaved searches are retained temporarily. Execute the Run All Searches task to re-run these searches and update the search results.
- *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: 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.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
idProj	Integer	Specify the project ID.
filesToIndex	String	Specify which files you want to update: <ul style="list-style-type: none">indexAll: The entire project.indexSavedSearches: A saved search.indexSampledOnly: Sampled files only.
searchNames	Array	<p>If you specified indexSavedSearches as the Files to Index parameter, specify the name(s) of the saved search(es) to update. The files returned by that search will be updated.</p> <p>If you want to update the files returned by multiple saved searches, specify the search names as a comma-separated list. For example:</p> <pre>"SearchNames": ["sa1", "sa2"]</pre>

Response Codes

- 202: Accepted
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

A string indicating whether the task was executed successfully.

GET: Get Task Status (getTaskStatus)

Obtain the status of a task that is being run on a project.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the project ID.
stTask	String	Select the task whose status you want to obtain.
current	String	Specify whether you want to obtain the status of the current invocation of the task, or the previous invocation of the task.

Request Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

Parameter	Type	Description
id	String	The task ID.
name	String	The task name.
pid	String	The server host name.
status	String	The task status.
idZlpUser	Integer	The ZLP user ID assigned to the user who started the task.
dateStart	String	The date and time the task began.

Parameter	Type	Description
mapParam	String	Indicates the status of the task, i.e., whether it was successful or not. If there are any errors, the exception names will be printed here.
dateEnd	String	The date and time the task completed.

POST: Run All Searches (runallsearches)

In the FAM module, single project searches include an option to save a search so that you can access it later. Unsaved searches are retained temporarily. Execute the Run All Searches task to re-run these searches and update the search results.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project whose searches you want to run.

Request Body Schema Fields

None.

Response Codes

- 202: Accepted
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project you want to run the task on.

Request Body Schema Fields

None.

Response Codes

- 202: Accepted
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project you want to crawl.

Request Body Schema Fields

None.

Response Codes

202: Accepted
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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 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.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project you want to run the task on.

Request Body Schema Fields

None.

Response Codes

202: Accepted
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project you want to run the task on.

Request Body Schema Fields

None.

Response Codes

202: Accepted
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project you want to run the task on.

Request Body Schema Fields

None.

Response Codes

- 202: Accepted
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project you want to run the task on.

Request Body Schema Fields

None.

Response Codes

- 202: Accepted
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

A string indicating whether the task was executed successfully.

FAM/Project Privileges

In the FAM module, privileges are granted to users or to security groups to allow them to perform additional actions on a project. Privileges are granted on a project-by-project basis, so a user or security group can have different privilege levels for different projects. An example of a privilege that can be granted to users or security groups is the “Search” privilege.

The following sections describe the FAM/Project Privileges endpoints available in the REST API:

- *PUT: Add or Grant Group Project Privileges (addgroupprojectprivileges, grantgroupprivileges):* Assign project privileges to a security group.
- *PUT: Add or Grant User Project Privileges (adduserprojectprivileges, grantuserprojectprivileges):* Assign project privileges to a user.
- *GET: Get Project Entity IDs (getprojectentityids):* Get project entity IDs for a project. Each project entity represents a user or security group that has been assigned privileges within the project.
- *GET: Get Project Entity Privileges (getprojectentityprivileges):* Get project entity privileges for a project. Each entity represents a user or security group that has been assigned a set of permissions within the project.
- *GET: Get Project Entity Types (getprojectentitytypes):* Get project entity types for a project. Each entity represents a group of users that have been assigned a set of permissions within the project.
- *GET: Get Project Privileges Using Project ID (getprojectprivileges):* Get the project privileges that have been assigned within a project.
- *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.

PUT: Add or Grant Group Project Privileges (addgroupprojectprivileges, grantgroupprivileges)

Assign project privileges to a security group. Please note that **addgroupprojectprivileges** can only be used to assign privileges to a single security group, and **grantgroupprivileges** can be used to assign privileges to multiple security groups.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID.
zlpUserIds	Integer	<p>For addgroupprivileges, specify the ZLP user ID of the security group to grant the privileges to. For example:</p> <pre>"zlpUserIds": [0]</pre> <p>For grantgroupprojectprivileges, specify a list of ZLP user IDs identifying the security groups to grant the privileges to. Specify the group IDs as a comma-separated list. For example:</p> <pre>"zlpUserIds": [0,1,2,3]</pre>
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
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 Codes

- 200: Successful
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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: Add or Grant User Project Privileges (adduserprojectprivileges, grantuserprojectprivileges)

Assign project privileges to a user or group of users.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID.
zlpUserIds	Integer	An array of user IDs identifying 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 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.
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 Codes

- 200: Successful
- 401: Unauthorized
- 403: Forbidden

404: Not Found
500: Internal Server Error

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.

GET: Get Project Entity IDs (getprojectentityids)

Get project entity IDs for a project. Each project entity represents a user or security group that has been assigned privileges within the project.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project from which you want to retrieve the information.
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 Codes

206: Partial Content

400: Bad Request

401: Unauthorized

403: Forbidden

404: Not Found

500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
Entity IDs: Includes the following fields for each project entity that has been created within the project. Each project entity represents a user or security group that has been assigned privileges within the project.		
userName	String	The name of the user or security group.
entityId	Integer	The entity ID assigned to the project entity.

GET: Get Project Entity Privileges (getprojectentityprivileges)

Get project entity privileges for a project. Each entity represents a user or security group that has been assigned a set of permissions within the project.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project from which you want to retrieve the information.
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 Codes

206: Partial Content

400: Bad Request

401: Unauthorized

403: Forbidden

404: Not Found

500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
Storage Project Entity Privileges: Includes an entry for each project entity that has been created within the project. Each project entity represents a user or security group that has been assigned privileges within the project. The fields included in each entry are described below:		
userName	String	The name of the user or security group.
entityPrivileges	Includes the entity ID, entity type, and a list of privileges assigned to the user or security group.	

GET: Get Project Entity Types (getprojectentitytypes)

Get project entity types for a project. Each entity represents a group of users that have been assigned a set of permissions within the project.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project from which you want to retrieve the information.
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 Codes

206: Partial Content

400: Bad Request

401: Unauthorized

403: Forbidden

404: Not Found

500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
Entity Types: Includes an entry for each project entity that has been created within the project. Each project entity represents a user or security group that has been assigned privileges within the project. The fields included in each entry are described below:		
userName	String	The name of the user or security group.
entityType	Integer	The entity type.

GET: Get Project Privileges Using Project ID (getprojectprivileges)

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

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project from which you want to retrieve the information.
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 Codes

206: Partial Content
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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: Revoke All Project Privileges (revokeallprojectprivileges)

Revoke all privileges that have been assigned within a project.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project from which you want to revoke privileges.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID.
zlpUserIds	Integer	An array of group IDs identifying the security groups from which the project privileges should be revoked. Specify the group IDs as a comma-separated list. For example: <pre>"zlpUserIds": [0,1,2,3]</pre>

Response Codes

200: Successful

400: Bad Request

401: Unauthorized

403: Forbidden

404: Not Found

500: Internal Server Error

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.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
projectId	Integer	Specify the project ID.
zlpUserIds	Integer	An array of user IDs identifying the users from whom the project privileges should be revoked. Specify the user IDs as a comma-separated list. For example: <code>"zlpUserIds": [0,1,2,3]</code>

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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

A project is essentially 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.

The following sections describe the UAA/Projects endpoints available in the REST API. Use these endpoints to create and manage Google Drive, OneDrive, SharePoint and File Share projects:

- *POST: Create Google Drive Project (creategoogledriveproject)*
- *POST: Create OneDrive Project (createonedriveproject)*
- *POST: Create Project - Supports File Share Projects Only (create)*
- *POST: Create SharePoint Project*
- *DELETE: Remove Project (deleteusingid)*
- *GET: Get All Projects (getallprojects)*
- *GET: Get Sub-Folders (subfolders)*
- *GET: Get Project Info by ID (getprojectusingid)*
- *GET: Get Project Info by Name (getprojectusingname)*
- *GET: Get Projects of File Server (getprojectsusingidfileserver)*
- *GET: Get Projects Using Search (getprojectusingpatternsearch)*
- *PUT: Update OneDrive Project (updateonedriveproject)*
- *PUT: Update Project - File Share Projects Only (update)*
- *PUT: Update SharePoint Project (updatesharepointproject)*

POST: Create Google Drive Project (creategoogledriveproject)

Create a project on a Google Drive server.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
sharedDrive	Boolean	Set to True if this is a shared drive.
projectInfo: Defines project configuration information.		
id	Integer	Specify the project ID.
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 7 for Google Drive.
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 obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
domainId	Integer	Specify the domain ID.
allowVersioning	Boolean	Reserved
flags	Boolean	<div>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.<ul style="list-style-type: none">allowsAddition: Allow new files to be added to the project during file crawls.</div>

Schema Field	Type	Description
		<ul style="list-style-type: none"> • 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.
folderRoot	These fields are output fields 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. Not required.

Schema Field	Type	Description
dirSpec		Specify which folders and sub-folders will be included in the project. Each entry in the <code>allEntry</code> 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: <code>C:\\Users\\bross\\Downloads\\fileset</code> You could also specify the path as: <code>C:/Users/bross/Downloads/fileset</code>
exclusionPathSet	String	Specify the relative paths of any sub-folders that should not be included when crawling the project. For example: <code>C:\\Users\\adunna\\Downloads\\fam</code> You could also specify the path as: <code>C:/Users/adunna/Downloads/fam</code>
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: <ul style="list-style-type: none"> • fprojectAdminPrivilege: Perform administrative actions on the project. • fReadPrivilege: Read the contents of the project. • fSearchPrivilege: Search the contents of the project. • fSharePrivilege: Not supported. • fAuditPrivilege: View audit trail data for the project. • fWebDavPrivilege: Not supported.

Response Codes

200: Successful
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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 OneDrive Project (createonedriveproject)

Create a project on a OneDrive site.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the project ID.
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 obtain server IDs with the Get > GetAllMailServers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
domainId	Integer	Specify the domain ID.
allowVersioning	Boolean	Not required for FAM operations.
flags	Boolean	<p>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.</p> <ul style="list-style-type: none">• 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.

Schema Field	Type	Description
		<ul style="list-style-type: none"> 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.
folderRoot	These fields are output fields 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. Not required.
dirSpec	Specify which folders and sub-folders will be included in the project. Each entry in the <code>allEntry</code> 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: <code>C:\\Users\\bross\\Downloads\\fileset</code>

Schema Field	Type	Description
		You could also specify the path as: <code>C:/Users/bross/Downloads/fileset</code>
exclusionPathSet	String	Specify the relative paths of any sub-folders that should not be included when crawling the project. For example: <code>C:\\Users\\adunna\\Downloads\\fam</code> You could also specify the path as: <code>C:/Users/adunna/Downloads/fam</code>
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: <ul style="list-style-type: none">• fprojectAdminPrivilege: Perform administrative actions on the project.• fReadPrivilege: Read the contents of the project.• fSearchPrivilege: Search the contents of the project.• fSharePrivilege: Not supported.• fAuditPrivilege: View audit trail data for the project.• fWebDavPrivilege: Not supported.

Response Codes

201: Created
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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 Project - Supports File Share Projects Only (create)

Create a project on a file server.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the project ID.
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 obtain server IDs with the Get > GetAllMailServers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
domainId	Integer	Specify the domain ID.
allowVersioning	Boolean	Not required for FAM operations.
flags	Boolean	<p>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.</p> <ul style="list-style-type: none">• 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

Schema Field	Type	Description
		<p>will automatically have access to the folder in the project in the ZL File Archiving application.</p> <ul style="list-style-type: none"> • 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.
folderRoot	These fields are output fields used 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 <code>allEntry</code> array identifies a project folder and includes the following fields:	

Schema Field	Type	Description
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
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: <ul style="list-style-type: none"> • fprojectAdminPrivilege: Perform administrative actions on the project. • fReadPrivilege: Read the contents of the project. • fSearchPrivilege: Search the contents of the project. • fSharePrivilege: TBD • fWebDavPrivilege: TBD • fAuditPrivilege: View audit trial data for the project.

Response Codes

200: Successful
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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

Create a project on a SharePoint site.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the project ID.
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 obtain server IDs with the Get > GetAllMailServers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
domainId	Integer	Specify the domain ID.
allowVersioning	Boolean	Not required for FAM operations.
flags	Boolean	<p>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.</p> <ul style="list-style-type: none">• 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

Schema Field	Type	Description
		<p>will automatically have access to the folder in the project in the ZL File Archiving application.</p> <ul style="list-style-type: none"> • 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.
folderRoot	These fields are output fields used 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. Not required.
dirSpec	Specify which folders and sub-folders will be included in the project. Each entry in the <code>allEntry</code> array identifies a project folder and includes the following fields:	

Schema Field	Type	Description
symbolicLink	String	The path display name.
rootPath	String	Specify the root path of the folder. For example: <code>C:\\Users\\bross\\Downloads\\fileset</code> You could also specify the path as: <code>C:/Users/bross/Downloads/fileset</code>
exclusionPathSet	String	Specify the relative paths of any sub-folders that should not be included when crawling the project. For example: <code>C:\\Users\\adunna\\Downloads\\fam</code> You could also specify the path as: <code>C:/Users/adunna/Downloads/fam</code>
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: <ul style="list-style-type: none"> • fprojectAdminPrivilege: Perform administrative actions on the project. • fReadPrivilege: Read the contents of the project. • fSearchPrivilege: Search the contents of the project. • fSharePrivilege: TBD • fWebDavPrivilege: TBD • fAuditPrivilege: View audit trial data for the project.
siteTitle	String	Type the site title.
siteURL	String	Type the SharePoint site URL.
guid	String	Type the SharePoint site GUID. Refer to the <i>SharePoint Archiving Administrator's Guide</i> for instructions to follow when retrieving SharePoint site GUID values.

Schema Field	Type	Description
username	String	<p>If using the client-side connector, type the user ID of a site collection administrator who has access to the site URL.</p> <p>When using o365 connector, ensure that you append “o365:” to the user ID/name. For example, if the username is john@demo.onmicrosoft.com, then you specify the value in this field as o365:john@demo.onmicrosoft.com</p>
password	String	<p>If using the Cloud SharePoint Connector, type the password of the user whose ID was specified for the username field.</p>
spServerAPI	Boolean	<p>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.</p> <p>When using the Cloud SharePoint Connector to archive the project files, leave this check box unselected.</p>
skipHidden	Boolean	<p>Set to True to skip archiving lists that are hidden in SharePoint.</p>

Schema Field	Type	Description
inclusive exclusive	String	<p>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.</p> <p>Within the inclusion and exclusion filters there are two different settings:</p> <ul style="list-style-type: none">T: Indicates the List Type. For example, Document Library, Announcements, Events, Calendar etc. Syntax: T:DocumentLibraryL: Indicates the various names of the type of the Lists. For ex: Shared Documents and Site Assets are lists under Document Library. Syntax: L:Shared Documents, L:Lists/WORMProject <p>Home/L:Lists/Delegation Agreement; where Delegation Agreement is the name of the list.</p> <p>For some lists, the name is prefixed with 'List/'. You must check the URL before including it in the necessary fields.</p> <p>Exclusion/Inclusion Parameter Syntax</p> <ul style="list-style-type: none">All document libraries: T:DocumentLibrarySpecific 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 <p>Please note that Exclusion takes a precedence over inclusions. For example, if you have specified inclusion parameters as <i>Home/regulatory/L:Test</i> and exclusion parameter as: <i>Home/regulatory</i>, "regulatory" will not be crawled.</p>

Response Code

- 200: Successful
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be deleted.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

Response Schema Fields

None

GET: Get All Projects (getallprojects)

Retrieve a list of all the projects on the server.

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 Codes

206: Partial Content

400: Bad Request

403: Forbidden

404: Not Found

500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
<p>Project Entries: Includes an entry for each project found on the server. The fields included for each 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:</p> <ul style="list-style-type: none">• Google Drive Projects: <i>POST: Create Google Drive Project (creategoogledriveproject)</i> on page 240• OneDrive Projects: <i>POST: Create OneDrive Project (createonedriveproject)</i> on page 244• File Share Projects: <i>POST: Create Project - Supports File Share Projects Only (create)</i> on page 247• SharePoint Projects: <i>POST: Create SharePoint Project</i> on page 251		

GET: Get Sub-Folders (subfolders)

Retrieve a list of sub-folders for a parent folder.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be viewed.
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 Codes

- 206: Partial Content
- 400: Bad Request
- 404: Not Found

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	Folder name.
displayName	String	The folder display name.
description	String	Description.

Schema Field	Type	Description
relativePath	String	The relative path of the folder.
type	Integer	The project type.
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.

Request Parameters

Parameter	Type	Description
projId	Integer	The ID of the project to be viewed.

Request Body Schema Fields

None.

Response Codes

200: Successful

403: Forbidden

404: Not Found

500: Internal Server Error

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:

- **Google Drive Projects:** *POST: Create Google Drive Project (creategoogledriveproject)* on page 240
- **OneDrive Projects:** *POST: Create OneDrive Project (createonedriveproject)* on page 244
- **File Share Projects:** *POST: Create Project - Supports File Share Projects Only (create)* on page 247
- **SharePoint Projects:** *POST: Create SharePoint Project* on page 251

GET: Get Project Info by Name (getprojectusingname)

Retrieve project's configuration. Specify the project by its name.

Request Parameters

Parameter	Type	Description
projName	String	Specify the name of the project to be viewed.

Request Body Schema Fields

None.

Response Codes

200: Successful

403: Forbidden

404: Not Found

500: Internal Server Error

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:

- **Google Drive Projects:** *POST: Create Google Drive Project (creategoogledriveproject)* on page 240
- **OneDrive Projects:** *POST: Create OneDrive Project (createonedriveproject)* on page 244
- **File Share Projects:** *POST: Create Project - Supports File Share Projects Only (create)* on page 247
- **SharePoint Projects:** *POST: Create SharePoint Project* on page 251

GET: Get Projects of File Server (getprojectsusingidfileserver)

Retrieve a list of projects that have been created on a given file share server, OneDrive server or SharePoint server.

Request Parameters

Parameter	Type	Description
fileServerId	Integer	The ID of the file server to be viewed. You can obtain server IDs with the Get > GetAllMailServers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
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 Codes

206: Partial Content
400: Bad Request
403: Forbidden
404: Not Found
500: Internal Server Error

Response Schema Fields

The response includes an entry for each project found on the server. The schema fields included in each entry are the same as those required in the request schema when creating it. They represent the configuration of the project. For descriptions of these fields, refer to the following sections:

- **OneDrive Projects:** *POST: Create OneDrive Project (createonedriveproject)* on page 244
- **File Share Projects:** *POST: Create Project - Supports File Share Projects Only (create)* on page 247
- **SharePoint Projects:** *POST: Create SharePoint Project* on page 251

GET: Get Projects Using Search (getprojectusingpatternsearch)

Retrieve a project by searching for a specific project name.

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 Codes

206: Partial Content

400: Bad Request

403: Forbidden

404: Not Found

500: Internal Server Error

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:

- **Google Drive Projects:** *POST: Create Google Drive Project (creategoogledriveproject)* on page 240
- **OneDrive Projects:** *POST: Create OneDrive Project (createonedriveproject)* on page 244
- **File Share Projects:** *POST: Create Project - Supports File Share Projects Only (create)* on page 247
- **SharePoint Projects:** *POST: Create SharePoint Project* on page 251

PUT: Update OneDrive Project (updateonedriveproject)

Update a OneDrive project's configuration.

Request Parameters

None.

Request Body Schema Fields

The schema fields required to update the OneDrive project's configuration are the same as those that must be specified when creating the project, as described in *POST: Create OneDrive Project* on page 310.

Response Codes

200: Successful

401: Unauthorized

403: Forbidden

404: Not Found

500: Internal Server Error

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 *POST: Create OneDrive Project* on page 310.

PUT: Update Project - File Share Projects Only (update)

Update a file share project's configuration.

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 *POST: Create Project - Supports File Share Projects Only (create)* on page 247.

Response Codes

200: Successful
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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 *POST: Create Project - Supports File Share Projects Only (create)* on page 247.

PUT: Update SharePoint Project (updatesharepointproject)

Update a SharePoint project's configuration.

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, as described in *POST: Create SharePoint Project* on page 318.

Response Codes

200: Successful

401: Unauthorized

403: Forbidden

404: Not Found

500: Internal Server Error

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 *POST: Create SharePoint Project* on page 318.

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. A case’s department can be defined during case setup.

The following sections describe the UAA/Roles endpoints available in the REST API. You can use these endpoints to grant and revoke roles within the UAA module and 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 (grantrroles)*: Grant roles to a user.
- *PUT: Revoke User Roles (revokerroles)*: 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.

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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized

403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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	Strinf	A description of the role.
fSystemRole	Boolean	Indicates whether the role is a system role (True) or not.
appld	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.

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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.
isSystemRole	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.

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	The ZLP user ID assigned to the user whose roles you want to view.
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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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: <ul style="list-style-type: none">• 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 Department Table.

PUT: Grant User Roles (granroles)

Grant roles to a user.

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.
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 obtaining role IDs, refer to the following sections: <ul style="list-style-type: none"><i>GET: Get All Custom Roles</i> on page 268<i>GET: Get All System Roles</i> on page 270
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: <ul style="list-style-type: none">Global: GlobalInclRecur: On selected departments recursivelyIncl: On selected departments only
allScopeDomainIds	Integer	An array of domain IDs identifying the departments the role is applicable to for roles that are only granted on selected departments. These can be retrieved from the ArchiveServer Department Table. Specify the domain IDs as a comma-separated list. For example: <pre>"AllScopeDomainIds": [0,1,2,3]</pre>

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden

404: Not Found
406: Not Acceptable
500: Internal Server Error

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.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zipUserId	Integer	Specify the ZL user ID of the user you want to modify.
roles	Integer	<div>An array of role IDs identifying the roles to be revoked. Specify the role IDs as a comma-separated list. For example: <pre>"Roles": [0, 1, 2, 3]</pre> For information on obtaining role IDs, refer to the following sections:<ul style="list-style-type: none"><i>GET: Get All Custom Roles</i> on page 268<i>GET: Get All System Roles</i> on page 270</div>

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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, 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.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
aldZlpUser	Integer	An array of ZLP user IDs identifying the users who should be added to the security group. Specify the user IDs as a comma-separated list. For example: <pre>"aIdzlpUser": [0,1,2,3]</pre>

Response Codes

- 200: OK
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Error

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.

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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
Security Groups: Includes an entry for each 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 <i>POST: Create User</i> on page 366.		

PUT: Remove Security Group (removesecuritygroup)

Remove a security group.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
aldZlpUser	Integer	An array of ZLP user IDs identifying the security groups to be removed. Specify the security group IDs as a comma-separated list. For example: "aIdzlpUser": [0,1,2,3]

Response Codes

- 200: OK
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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 server, 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 how the UAA/agents endpoints available in the REST API:

- *POST: Create A Lotus Domino Journal Server Agent (createlotusjournalagent)*: Create a Journal server agent for a Lotus Domino server.
- *POST: Create Lotus Domino Mailbox Crawl Server Agent (createlotusmailboxcrawlagent)*: Create a Mailbox Crawl agent for a Lotus Domino server.
- *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.
- *POST: Create a User Synchronization Mail Server Agent (createusersyncserveragent)*: Create a User Synchronization server agent for a Microsoft Exchange, Microsoft EWS or Google Mail Server.
- *DELETE: Delete a Server Agent (deleteserveragent)*: Delete a server agent.
- *GET: Get Server Agent Using Function (getagentusingfunction)*: Retrieve the configuration of a server agent.
- *GET: Get Server Agent Using Name (getagentusingname)*: Retrieve the configuration of a server agent.
- *GET: Get All Agents of Mail Server (getagentusingidmailserver)*: Retrieve the configurations of the server agents that have been added to a specific server.
- *PUT: Update A Lotus Domino Journal Server Agent (updatelotusjournalagent)*: Update the configuration of a Journal agent that has been added to a Lotus Domino server.
- *PUT: Update Lotus Domino Mailbox Crawl Server Agent (updatelotusmailboxcrawlagent)*: Update the configuration of a Mailbox Crawl agent that has been added to a Lotus Domino server.
- *PUT: Update a Server Agent (updateserveragent)*: Update the configuration of a File Archive server or a Mailbox Crawl agent. These agents can be used with multiple server types, as specified later.
- *PUT: Update a User Synchronization Mail Server Agent (updateusersyncserveragent)*: Update the configuration of a User Synchronization server agent for a Microsoft Exchange, Microsoft EWS or Google Mail Server.

POST: Create A Lotus Domino Journal Server Agent (createlotusjournalagent)

Create a Journal server agent for a Lotus Domino mail server.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
serverAgent:	These fields define the configuration of a a Journal server agent for a Lotus Domino mail server:	
agentName	String	Enter the server agent's name.
agentType	String	Specify lotusJournal as the server agent type.
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.
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 Louts Domino server the agent will run on. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
lotusJournalDbName	String	Specify a name for the Lotus database journal instance.

Response Codes

201: Created
400: Bad Request
401: Unauthorized
403: Forbidden

404: Not Found
406: Not Acceptable
500: Internal Server Error

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 server agent.
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, the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

POST: Create Lotus Domino Mailbox Crawl Server Agent (createlotusmailboxcrawlagent)

Create a Mailbox Crawl server agent for a Lotus Domino mail server.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
agentName	String	Enter the server agent's name.
agentType	String	Specify mbCrawl as the server agent type.
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.
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 Louts Domino server the agent will run on. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
scanChildServers	Boolean	Set to True if this Domino server is a parent server whose child servers should be scanned as well.
threadLimit	Integer	Specify the number of threads the agent should be able to use during the crawling process. This is only applicable if the scanChildServers setting is set to True.

Response Codes

201: Created

400: Bad Request

401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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	Reserved.
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, the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

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.

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: <ul style="list-style-type: none">• 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.
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.

Schema Field	Type	Description
mailServerId	Integer	Specify the ID of the server the agent will run on. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.

Response Codes

201: Created
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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	Reserved.

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, the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

POST: Create a User Synchronization Mail Server Agent (createusersyncserveragent)

Create a User Synchronization server agent for a Microsoft Exchange, Microsoft EWS or Google Mail Server.

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: <ul style="list-style-type: none">For user synchronization on a Microsoft Exchange server, enter exchangeUserSync.For user synchronization on a Microsoft EWS server, enter ewsUserSyncFor user synchronization on a Google Mail server, enter googleUserSync
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.
defaultDepartment	Integer	Specify the ID of the default department to assign users to during user synchronization. The UAA/Departments endpoints provide several ways to retrieve department IDs. For more information, refer to <i>UAA/Departments</i> on page 105.
autoCreate	Boolean	Set to True to create user accounts for any new users found during synchronization.
autoTerminate	Boolean	Set to True to automatically terminate users during synchronization if necessary. This may occur if a user previously added to ZL UA no longer exists, or if the user is disabled.

Schema Field	Type	Description
ignoreDepartmentUpdates	Boolean	Set to True to prevent the user synchronization process from updating department information for existing users that have already been added to ZL UA.
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 Google Mail, Microsoft Exchange or Microsoft EWS server the agent will run on. You can obtain server IDs with the Get > GetAllMailServers endpoint, as described in <i>GET: Get All Servers</i> on page 164.

Response Codes

201: Created
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.

Schema Field	Type	Description
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	Reserved.
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, the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

DELETE: Delete a Server Agent (deleteserveragent)

Delete a server agent.

Request Parameters

Parameter	Type	Description
serverAgentName	String	Specify the name of the server agent to be deleted.
idMailServer	Integer	Specify the ID of the server that the server agent to be deleted belongs to. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.

Request Body Schema Fields

None.

Response Codes

- 200: OK
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

A string indicating whether the server was deleted successfully.

GET: Get Server Agent Using Function (getagentusingfunction)

Retrieve server agents from a server that serve a specific function (e.g., Mailbox Crawl or User Synchronization).

Request Parameters

Parameter	Type	Description
agentFunction	String	Specify the function of the server agent to be retrieved: <ul style="list-style-type: none">For file archiving agents, enter file.For mailbox crawling agents, enter archiving.For user synchronization agents, enter userSync.For journaling agents, enter journaling.
idMailServer	Integer	Specify the ID of the server that the server agent belongs to. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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

Schema Field	Type	Description
		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	Reserved.
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, the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

GET: Get Server Agent Using Name (getagentusingname)

Retrieve server agents from a server. Identify the server agent to be retrieved by its name.

Request Parameters

Parameter	Type	Description
agentName	String	Specify the name of the server agent to be retrieved.
idMailServer	Integer	Specify the ID of the server that the server agent belongs to. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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.

Schema Field	Type	Description
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	Reserved.
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, the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

GET: Get All Agents of Mail Server (getagentusingidmailserver)

Retrieve all server agents that have been added to a particular server.

Request Parameters

Parameter	Type	Description
idMailServer	Integer	Specify the ID of the server that the server agent belongs to. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

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.

Schema Field	Type	Description
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	Reserved.
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, the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

PUT: Update A Lotus Domino Journal Server Agent (`updatelotusjournalagent`)

Update the configuration of a Journal server agent that has been added to a Lotus Domino mail server.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
agentName	String	Enter the server agent's name.
agentType	String	Specify "Journal" as the server agent type.
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.
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 Louts Domino server the agent will run on. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
lotusJournalDbName	String	Specify a name for the Lotus database journal instance.

Response Codes

200: Updated
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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	Reserved.
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, the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

PUT: Update Lotus Domino Mailbox Crawl Server Agent (`updatelotusmailboxcrawlagent`)

Update a Mailbox Crawl server agent that has been added to a Lotus Domino mail server.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
agentName	String	Enter the server agent's name.
agentType	String	Specify "Mailbox Crawl" as the server agent type.
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.
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 Louts Domino server the agent will run on. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
scanChildServers	Boolean	Set to True if this Domino server is a parent server whose child servers should be scanned as well.
threadLimit	Integer	Specify the number of threads the agent should be able to use during the crawling process. This is only applicable if the scanChildServers setting is set to True.

Response Codes

200: Updated

400: Bad Request

401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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	Reserved.
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, the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

PUT: Update a Server Agent (updateserveragent)

Update a server agent that has been created 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.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
agentName	String	Enter the server agent's name.
agentType	String	Specify "Mailbox Crawl" as the server agent type.
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.
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 obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.

Response Codes

200: Updated
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found

406: Not Acceptable
500: Internal Server Error

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	Reserved.
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, the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

PUT: Update a User Synchronization Mail Server Agent (updateusersyncserveragent)

Update a User Synchronization server agent that has been added to a Microsoft Exchange, Microsoft EWS or Google Mail Server.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
agentName	String	Enter the server agent's name.
agentType	String	Specify "User Synchronization" as the server agent type.
defaultDepartment	Integer	Specify the ID of the default department to assign users to during user synchronization. The UAA/Departments endpoints provide several ways to retrieve department IDs. For more information, refer to <i>UAA/Departments</i> on page 105.
autoCreate	Boolean	Set to True to create user accounts for any new users found during synchronization.
autoTerminate	Boolean	Set to True to automatically terminate users during synchronization if necessary. This may occur if a user previously added to ZL UA no longer exists, or if the user is disabled.
ignoreDepartmentUpdates	Boolean	Set to True to prevent the user synchronization process from updating department information for existing users that have already been added to ZL UA.
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.
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.

Schema Field	Type	Description
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 Google Mail, Microsoft Exchange or Microsoft EWS server the agent will run on. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.

Response Codes

200: Updated
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

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	Reserved.

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, the date that the next iteration is scheduled.
iterationStartDate	String	
iterationUpdate	String	
iterationEndDate	String	
nextIterationDate	String	

FAM/Projects

A project represents a collection of folders and directories on a server that is grouped together for use within the FAM module. The following sections describe the FAM/Projects endpoints available in the REST API. You can use the endpoints to create and manage File Share, SharePoint and OneDrive projects for use within the FAM module:

- *POST: Create OneDrive Project (createonedriveproject)*
- *POST: Create Project - File Share Projects Only (create)*
- *POST: Create SharePoint Project (createsharepointproject)*
- *DELETE: Remove Project (deleteusingid)*
- *GET: Get All Projects (getallprojects)*
- *GET: Get Sub-Folders (subfolders)*
- *GET: Get Project Info by ID (getprojectusingid)*
- *GET: Get Project Info by Name (getprojectusingname)*
- *GET: Get Projects of File Server (getprojectsusingidfilesserver)*
- *GET: Get Projects Using Search (getprojectusingpatternsearch)*
- *PUT: Update OneDrive Project (updateonedriveproject)*
- *PUT: Update Project - File Share Projects Only (update)*
- *PUT: Update SharePoint Project (updatesharepointproject)*

POST: Create OneDrive Project (createonedriveproject)

Create a project on a OneDrive site.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the project ID.
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. Set this to 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 obtain server IDs with the Get > GetAllMailServers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
domainId	Integer	Specify the domain ID.
allowVersioning	Boolean	Not required for FAM operations.
flags	Boolean	<p>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.</p> <ul style="list-style-type: none">• 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

Schema Field	Type	Description
		<p>will automatically have access to the folder in the project in the ZL File Archiving application.</p> <ul style="list-style-type: none"> • 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.
folderRoot	These fields are output fields 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. Not required.
dirSpec	Specify which folders and sub-folders will be included in the project. Each entry in the <code>allEntry</code> array identifies a project folder and includes the following fields:	

Schema Field	Type	Description
symbolicLink	String	The path display name.
rootPath	String	Specify the root path of the folder. For example: <code>C:\\Users\\bross\\Downloads\\fileset</code> You could also specify the path as: <code>C:/Users/bross/Downloads/fileset</code>
exclusionPathSet	String	Specify the relative paths of any sub-folders that should not be included when crawling the project. For example: <code>C:\\Users\\adunna\\Downloads\\fam</code> You could also specify the path as: <code>C:/Users/adunna/Downloads/fam</code>
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: <ul style="list-style-type: none"> • fprojectAdminPrivilege: Perform administrative actions on the project. • fReadPrivilege: Read the contents of the project. • fSearchPrivilege: Search the contents of the project. • fSharePrivilege: Not supported. • fAuditPrivilege: View audit trial data for the project. • fWebDavPrivilege: Not supported.

Response Codes

201: Created
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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 Project - File Share Projects Only (create)

Create a project on a file server.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the project ID.
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. In ZL UA 10.3, this should be set to -1, as only File Server projects are supported.
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 obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
domainId	Integer	Specify the domain ID.
allowVersioning	Boolean	Not required for FAM operations.
flags	Boolean	<p>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.</p> <ul style="list-style-type: none">• 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.

Schema Field	Type	Description
		<ul style="list-style-type: none">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.
folderRoot	These fields are output fields used 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.

Schema Field	Type	Description
dirSpec		Specify which folders and sub-folders will be included in the project. Each entry in the <code>allEntry</code> 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: <code>C:\\Users\\bross\\Downloads\\fileset</code> You could also specify the path as: <code>C:/Users/bross/Downloads/fileset</code>
exclusionPathSet	String	Specify the relative paths of any sub-folders that should not be included when crawling the project. For example: <code>C:\\Users\\adunna\\Downloads\\fam</code> You could also specify the path as: <code>C:/Users/adunna/Downloads/fam</code>
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: <ul style="list-style-type: none"> • fprojectAdminPrivilege: Perform administrative actions on the project. • fReadPrivilege: Read the contents of the project. • fSearchPrivilege: Search the contents of the project. • fSharePrivilege: TBD • fWebDavPrivilege: TBD • fAuditPrivilege: View audit trial data for the project.

Response Codes

200: Successful
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the project ID.
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. For SharePoint, set this to 3.
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 obtain server IDs with the Get > GetAllMailServers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
domainId	Integer	Specify the domain ID.
allowVersioning	Boolean	Not required for FAM operations.
flags	Boolean	<p>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.</p> <ul style="list-style-type: none"> 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

Schema Field	Type	Description
		<p>automatically have access to the folder in the project in the ZL File Archiving application.</p> <ul style="list-style-type: none"> • 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.
folderRoot	These fields are output fields used 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. Not required.
dirSpec	Specify which folders and sub-folders will be included in the project. Each entry in the <code>allEntry</code> array identifies a project folder and includes the following fields:	

Schema Field	Type	Description
symbolicLink	String	The path display name.
rootPath	String	Specify the root path of the folder. For example: <code>C:\\Users\\bross\\Downloads\\fileset</code> You could also specify the path as: <code>C:/Users/bross/Downloads/fileset</code>
exclusionPathSet	String	Specify the relative paths of any sub-folders that should not be included when crawling the project. For example: <code>C:\\Users\\adunna\\Downloads\\fam</code> You could also specify the path as: <code>C:/Users/adunna/Downloads/fam</code>
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: <ul style="list-style-type: none"> • fprojectAdminPrivilege: Perform administrative actions on the project. • fReadPrivilege: Read the contents of the project. • fSearchPrivilege: Search the contents of the project. • fSharePrivilege: TBD • fWebDavPrivilege: TBD • fAuditPrivilege: View audit trial data for the project.
siteTitle	String	Type the site title.
siteURL	String	Type the SharePoint site URL.
guid	String	Type the SharePoint site GUID. Refer to the <i>SharePoint Archiving Administrator's Guide</i> for instructions to follow when retrieving SharePoint site GUID values.

Schema Field	Type	Description
username	String	<p>If using the client-side connector, type the user ID of a site collection administrator who has access to the site URL.</p> <p>When using o365 connector, ensure that you append "o365:" to the user ID/name. For example, if the username is john@demo.onmicrosoft.com, then you specify the value in this field as o365:john@demo.onmicrosoft.com</p>
password	String	<p>If using the Cloud SharePoint Connector, type the password of the user whose ID was specified for the username field.</p>
spServerAPI	Boolean	<p>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.</p> <p>When using the Cloud SharePoint Connector to archive the project files, leave this check box unselected.</p>
skipHidden	Boolean	<p>Set to True to skip archiving lists that are hidden in SharePoint.</p>

Schema Field	Type	Description
inclusive exclusive	String	<p>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.</p> <p>Within the inclusion and exclusion filters there are two different settings:</p> <ul style="list-style-type: none">• 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 ex: Shared Documents and Site Assets are lists under Document Library. Syntax: L:Shared Documents, L:Lists/WORMProject <p>Home/L:Lists/Delegation Agreement; where Delegation Agreement is the name of the list.</p> <p>For some lists, the name is prefixed with 'List/'. You must check the URL before including it in the necessary fields.</p> <p>Exclusion/Inclusion Parameter Syntax</p> <ul style="list-style-type: none">• 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 <p>Please note that Exclusion takes a precedence over inclusions. For example, if you have specified inclusion parameters as <i>Home/regulatory/L:Test</i> and exclusion parameter as: <i>Home/regulatory</i>, "regulatory" will not be crawled.</p>

Response Codes

- 200: Successful
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be deleted.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

Response Schema Fields

None

GET: Get All Projects (getallprojects)

Retrieve a list of all projects on the server.

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 Codes

- 206: Partial Content
- 400: Bad Request
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

Response Schema Fields

The schema fields returned for the project are the same as those required in the request schema when creating it. They represent the configuration of the project. For descriptions of these fields, refer to the following sections:

- **OneDrive Servers:** *POST: Create OneDrive Project* on page 310
- **File Share Servers:** *POST: Create Project - File Share Projects Only (create)* on page 314
- **SharePoint Servers:** *POST: Create SharePoint Project* on page 318

GET: Get Sub-Folders (subfolders)

Retrieve a list of sub-folders for a parent folder.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project to be viewed.
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 this one.
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 Codes

- 206: Partial Content
- 400: Bad Request
- 404: Not Found

Response Schema Fields

Schema Field	Type	Description
Folders: Includes an entry for each sub-folder that is found. The following fields are included in each entry:		
dirId	Integer	The folder ID.
parentId	Integer	The ID of the parent folder.
name	String	Folder name.
displayName	String	The folder display name.
description	String	Description.

Schema Field	Type	Description
relativePath	String	The relative path of the folder.
type	Integer	Reserved.
storageSize	Integer	The folder storage size.
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.

Request Parameters

Parameter	Type	Description
projId	Integer	The ID of the project to be viewed.

Request Body Schema Fields

None.

Response Codes

200: Successful

403: Forbidden

404: Not Found

500: Internal Server Error

Response Schema Fields

The schema fields returned for the project are the same as those required in the request schema when creating it. They represent the configuration of the project. For descriptions of these fields, refer to the following sections:

- **OneDrive Servers:** *POST: Create OneDrive Project* on page 310
- **File Share Servers:** *POST: Create Project - File Share Projects Only (create)* on page 314
- **SharePoint Servers:** *POST: Create SharePoint Project* on page 318

GET: Get Project Info by Name (getprojectusingname)

Retrieve project's configuration. Specify the project by its name.

Request Parameters

Parameter	Type	Description
projName	String	The name of the project to be viewed.

Request Body Schema Fields

None.

Response Codes

200: Successful
403: Forbidden
404: Not Found
500: Internal Server Error

Response Schema Fields

The schema fields returned for the project are the same as those required in the request schema when creating it. They represent the configuration of the project. For descriptions of these fields, refer to the following sections:

- **OneDrive Servers:** *POST: Create OneDrive Project* on page 310
- **File Share Projects:** *POST: Create Project - File Share Projects Only (create)* on page 314
- **SharePoint Servers:** *POST: Create SharePoint Project* on page 318

GET: Get Projects of File Server (getprojectsusingidfileserver)

Retrieve a list of projects that have been created on a given server.

Request Parameters

Parameter	Type	Description
fileServerId	Integer	The ID of the server to be viewed. You can obtain server IDs with the Get > GetAllMailSevers endpoint, as described in <i>GET: Get All Servers</i> on page 164.
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 Codes

206: Partial Content

400: Bad Request

403: Forbidden

404: Not Found

500: Internal Server Error

Response Schema Fields

The response includes an entry for each project found on the server. The schema fields returned for the project are the same as those required in the request schema when creating it. They represent the configuration of the project. For descriptions of these fields, refer to the following sections:

- **OneDrive Servers:** *POST: Create OneDrive Project* on page 310
- **File Share Projects:** *POST: Create Project - File Share Projects Only (create)* on page 314
- **SharePoint Servers:** *POST: Create SharePoint Project* on page 318

GET: Get Projects Using Search (getprojectusingpatternsearch)

Retrieve a project by searching for a specific project name.

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 Codes

206: Partial Content

400: Bad Request

403: Forbidden

404: Not Found

500: Internal Server Error

Response Schema Fields

The schema fields returned for the project are the same as those required in the request schema when creating it. They represent the configuration of the project. For descriptions of these fields, refer to the following sections:

- **OneDrive Servers:** *POST: Create OneDrive Project* on page 310
- **File Share Servers:** *POST: Create Project - File Share Projects Only (create)* on page 314
- **SharePoint Servers:** *POST: Create SharePoint Project* on page 318

PUT: Update OneDrive Project (updateonedriveproject)

Update a OneDrive project's configuration.

Request Parameters

None.

Request Body Schema Fields

The schema fields required to update the OneDrive project's configuration are the same as those that must be specified when creating the project. For more information, refer to *POST: Create OneDrive Project* on page 310.

Response Codes

200: Successful
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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 *POST: Create OneDrive Project* on page 310.

PUT: Update Project - File Share Projects Only (update)

Update a File Share project's configuration.

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. For more information, refer to *POST: Create Project - File Share Projects Only (create)* on page 314.

Response Codes

200: Successful
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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 *POST: Create Project - File Share Projects Only (create)* on page 314.

PUT: Update SharePoint Project (updatesharepointproject)

Update a SharePoint project's configuration.

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. For more information, refer to *POST: Create SharePoint Project* on page 318.

Response Codes

200: Successful

401: Unauthorized

403: Forbidden

404: Not Found

500: Internal Server Error

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 *POST: Create OneDrive Project* on page 310.

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 (getalltagsofaproject)*
- *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)*
- *PUT: Add User Alias (adduseralias)*

DELETE: Delete Tag (deletetag)

Delete a tag.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project that the tag to be deleted belongs to.
tagId	Integer	Specify the ID of the tag to be deleted.

Request Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Bad Request
- 404: Not Found
- 500: Internal Server Error

Response Schema Fields

None.

GET: Get All Tags of a Project (getalltagsofaproject)

Retrieve the tags that have been created within a specific project.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project.
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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Bad Request
- 404: Not Found
- 500: Internal Server Error

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.

Schema Field	Type	Description
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	Indicates the tag is the root tag.
read_only		Indicates the tag is read-only.
enduser_tag		Indicates that the tag can be applied manually.
auto_tag		Indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag		Indicates the tag is a mutually exclusive tag.
tag_32		Indicates the tag is a PII tag.
tag_64		Indicates 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.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project the tag belongs to.
tagId	Integer	Specify the ID of the tag you want to view.

Request Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Bad Request
- 404: Not Found
- 500: Internal Server Error

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.
tagFlags	Array	An array of Boolean values indicating the status of various tag attributes.
root_node	Boolean	Indicates the tag is the root tag.
read_only		Indicates the tag is read-only.

Schema Field	Type	Description
enduser_tag		Indicates that the tag can be applied manually.
auto_tag		Indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag		Indicates the tag is a mutually exclusive tag.
tag_32		Indicates the tag is a PII tag.
tag_64		Indicates 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.

Request Parameters

Parameter	Type	Description
projectId	Integer	Specify the ID of the project the tag belongs to.
tagName	String	Specify the name of the tag you want to view.

Request Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Bad Request
- 404: Not Found
- 500: Internal Server Error

Response Schema Fields

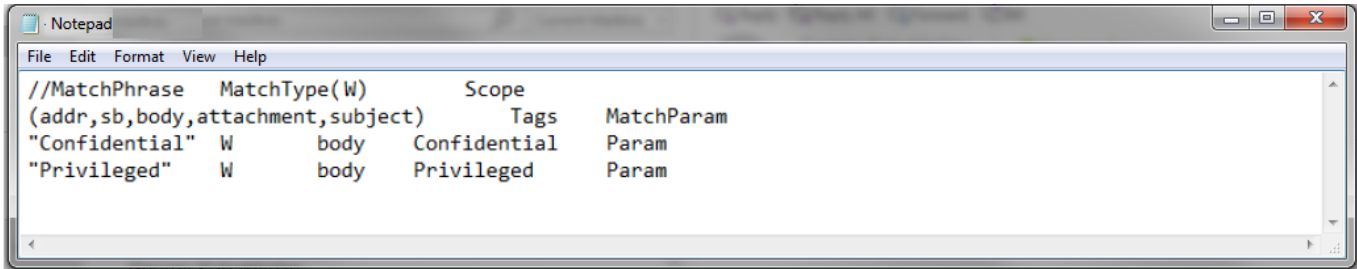
Schema Field	Type	Description
parent	String	The name of the tag's parent tag.
id	Integer	The tag ID.
projectId	Integer	The ID of the project the tag belongs to.
parentId	String	The ID of the tag's parent tag.
name	String	The internal name of the tag.
displayName	String	The display name of the tag.
tagFlags	Array	An array of Boolean values indicating the status of various tag attributes.
root_node	Boolean	Indicates the tag is the root tag.
read_only		Indicates the tag is read-only.

Schema Field	Type	Description
enduser_tag		Indicates that the tag can be applied manually.
auto_tag		Indicates that the tag can be applied automatically, i.e., via a tag specification file.
max_tag		Indicates the tag is a mutually exclusive tag.
tag_64		Indicates the tag is a content tag.
tag_64		Indicates 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 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 216.

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 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 <i>POST: Run Content Tagger</i> on page 221 and <i>POST: Update Index</i> on page 226.
tagContent	String	Copy the contents of the tab-delimited text file.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Bad Request
- 404: Not Found
- 500: Internal Server Error

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:

File Edit Format View Help			
//Rule Name	Rule Type	Param1=xxx,Param2=xxx	Tag Name 1, Tag Name 2
Date Created Rule #1	AgeRange	Field=fileCreated,DaysOlderThan=180	Created 6 to 12 Months Ago
Date Last Modified Rule #1	AgeRange	Field=fileLastModified,DaysOlderThan=182	Not Modified 6 Months

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 "zlbs" department:

```
OwnerOnly=true, DepartmentName=zlbs
```

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 "zlbs" departments on its ACL:

```
DepartmentName=zlbs
```

- **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 applied to 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 216.

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 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 <i>POST: Run Metadata Tagger</i> on page 223 and <i>POST: Update Index</i> on page 226.
tagContent	String	Copy the contents of the tab-delimited text file.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Bad Request
- 404: Not Found
- 500: Internal Server Error

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 **RegExp** 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 **RegExp** 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 216.

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 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 <i>POST: Run PII Tagger</i> on page 224 and <i>POST: Update Index</i> on page 226.
tagContent	String	Copy the contents of the tab-delimited text file.

Response Codes

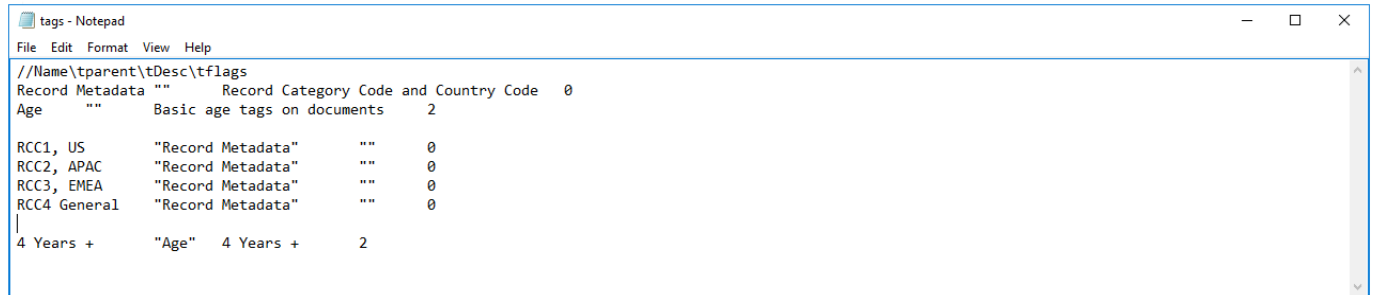
- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Bad Request
- 404: Not Found
- 500: Internal Server Error

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.

If the new tag doesn't have a parent, leave this field empty as “ ”.

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 216.

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 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 <i>POST: Update Index</i> on page 226.
tagContent	String	Copy the contents of the tab-delimited text file.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Bad Request
404: Not Found
500: Internal Server Error

Response Schema Fields

A string indicating whether the upload was successful.

FAM/Task Sets

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.

You can create task sets to run background tasks simultaneously, on an on-demand basis or at regularly scheduled intervals. Each task set configuration includes a group of background tasks and a list of projects and dashboards that the task set is applicable to. When the task set is executed, all of the background tasks will be executed upon the specified projects and upon all projects included in the specified dashboards.

The following sections describe the FAM/Task Sets endpoints available in the REST API. You can use these endpoints to create and manage task sets:

- *POST: Add Task Set Entries (addentries)*
- *DELETE: Delete Task Set Entry (deleteentry)*
- *GET: Get All Task Set Entries (getallentries)*
- *GET: Get All Entries Using Pattern (getallentriesusingpattern)*
- *GET: Get a Task Set Using ID (getusingid)*
- *POST: Update Task Set Entry (updateentry)*

POST: Add Task Set Entries (addentries)

Create a task set. You will specify which projects and dashboards the task set will be applicable to, and which background tasks will be run when the task set is executed.

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
name	String	Specify the name of the task set.
projectIds	Array	Specify the IDs of the project(s) that the task set will apply to. The tasks included in this task set will be executed upon these projects. If you want to apply the task set to multiple projects, specify the projects as a comma-separated list. For example: <pre>"projectIds": [1,2,3]</pre>
dashboardIds	Array	Specify the IDs of the dashboard(s) that the task set will apply to. The tasks included in this task set will be executed upon the projects included in each of these dashboards. If you want to apply the task set to multiple dashboards, specify the IDs as a comma-separated list. For example: <pre>"dashboardIds": [1,2,3]</pre>
fRunCrawl	Boolean	Set to True to include the Run Crawl task in the task set. This task scans the project directories specified for the project and add the contents to the project.

Schema Field	Type	Description
fContentIndex	Boolean	Set to True to include the Update Content Index task in the task set. 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.
fMetaDataTagger	Boolean	Set to True to include the MetaData Tagger task in the task set. This 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.
fContentTagger	Boolean	Set to True to include the Content Tagger task in the task set. This task applies the tags defined in any tag specification files that have been added to the project to the project's files.
fPIITagger	Boolean	Set to True to include the PII Tagger in the task set. 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.
fUpdateIndex	Boolean	Set to True to include the Update Index task in the task set. This updates the tagging information for the files included in the project.
fSearches	Boolean	Set to True to include the Run All Searches task in the task set. In the FAM module, single project searches include an option to save a search so that you can access it later. Unsaved searches are retained temporarily. Execute the Run All Searches task to re-run these searches and update the search results.
fRemediation	Boolean	Set to True to include the Run Remediation task in the task set. 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. When the Run Remediation task is executed on a project, all remediation actions configured for the project will be performed when you run this task.

Response Codes

206: Partial Content

400: Bad Request

401: Unauthorized
403: Forbidden
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

Parameter	Type	Description
id	Integer	The dashboard ID.
taskSetName	String	The task set name.
projectName	String	The name of the project(s) the task set is applicable to.
The remaining fields are Boolean values indicating which tasks were added to the task set. For example, if fRunCrawl is set to True, it indicates that the Run Crawl task was added to the task set.		

DELETE: Delete Task Set Entry (deleteentry)

Delete a task set.

Request Parameters

Parameter	Type	Description
entryId	Integer	Specify the ID of the task set to be deleted.

Request Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Bad Request
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

None.

GET: Get All Task Set Entries (getallentries)

Retrieve the configurations of the task sets that have been added.

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 Codes

206: Partial Content
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

The response schema includes an entry for each task set. The response schema fields returned for each task set are the same as those included in the response returned after the task set was created. For descriptions of these fields, refer to *POST: Add Task Set Entries* on page 353.

GET: Get All Entries of a Task Set (getallentriesofset)

Retrieve the configuration of a specific task set.

Request Parameters

Parameter	Type	Description
taskSetName	String	Specify the name of the task set to be retrieved.
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 Codes

206: Partial Content
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

The response schema includes the entries that have been added to the specified task set. The response schema fields returned for each task set are the same as those included in the response returned after the task set was created. For descriptions of these fields, refer to *POST: Add Task Set Entries* on page 353.

GET: Get All Entries Using Pattern (getallentriesusingpattern)

Retrieve the configurations of the task sets that have been added.

Request Parameters

Parameter	Type	Description
pattern	String	Enter the search pattern. The search will return task sets whose name includes or is similar to the search pattern.
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 Codes

206: Partial Content
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

The response schema includes an entry for each task set. The response schema fields returned for each task set are the same as those included in the response returned after the task set was created. For descriptions of these fields, refer to *POST: Add Task Set Entries* on page 353.

GET: Get a Task Set Using ID (getusingid)

Retrieve the configurations of a task set. Identify the task set by its ID.

Request Parameters

Parameter	Type	Description
entryId	String	Specify the task set ID.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 406: Not Acceptable
- 500: Internal Server Error

Response Schema Fields

The response schema fields returned for the task set are the same as those included in the response returned after the task set was created. For more information, refer to *POST: Add Task Set Entries* on page 353.

POST: Update Task Set Entry (updateentry)

Update the configuration of a task set. Specify the task set to be updated by its ID.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
id	Integer	Specify the ID of the task set to be updated.
taskSetName	String	Specify the name of the task set.
projectName	String	Specify the name of the project.
fRunCrawl	Boolean	Set to True to include the Run Crawl task in the task set. This task scans the project directories specified for the project and add the contents to the project.
fContentIndex	Boolean	Set to True to include the Update Content Index task in the task set. 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.
fMetaDataTagger	Boolean	Set to True to include the MetaData Tagger task in the task set. This 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.
fContentTagger	Boolean	Set to True to include the Content Tagger task in the task set. This task applies the tags defined in any tag specification files that have been added to the project to the project's files.
fPIITagger	Boolean	Set to True to include the PII Tagger in the task set. 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.
fUpdateIndex	Boolean	Set to True to include the Update Index task in the task set. This updates the tagging information for the files included in the project.

Schema Field	Type	Description
fSearches	Boolean	Set to True to include the Run All Searches task in the task set. In the FAM module, single project searches include an option to save a search so that you can access it later. Unsaved searches are retained temporarily. Execute the Run All Searches task to re-run these searches and update the search results.
fRemediation	Boolean	Set to True to include the Run Remediation task in the task set. 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. When the Run Remediation task is executed on a project, all remediation actions configured for the project will be performed when you run this task.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
406: Not Acceptable
500: Internal Server Error

Response Schema Fields

The response schema matches the request schema provided.

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.

The following sections describe the UAA/Users endpoints available in the REST API. You can use these endpoints to create and manage users within the FAM module and 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 (getuserusingalias)*
- *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 (moveusertoneewdepartment)*
- *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.

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.
alias	String	Enter the alias name.
aliasType	Integer	Specify the alias type: <ul style="list-style-type: none">0: Default (Email)1: X500 DN2: Address3: Manual4: Exchange Legacy DN5: Transformed Lotus DN6: NetBios User Name100: IM200: Bloomberg300: Parlano

Response Codes

201: Successful

400: Bad Request

401: Unauthorized

403: Forbidden

404: Not Found

500: Internal Server Error

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.

archiveServerDept	Integer	The department the user is assigned to.
iType	Integer	<div>The alias type:<ul style="list-style-type: none">0: Default (Email)1: X500 DN2: Address3: Manual4: Exchange Legacy DN5: Transformed Lotus DN6: NetBios User Name100: IM200: Bloomberg300: Parlano</div>
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. For more information, refer to *PUT: Update User Mail Server Information* on page 384.

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.
miscField1 miscField2	String	Use these fields to enter any additional information required for the user.

Schema Field	Type	Description
fSyncIncluded	Boolean	Specify whether the user should be included (True) in LDAP User Synchronizations or not (False).

Response Codes

201: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

Response Schema Fields

Schema Field	Type	Description
idZlpUser	Integer	The ZL user ID assigned to the user.
iType	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.
retags	String	Retention tags applied to the user.
deptName	String	The department the user is assigned to.
altReviewDepts	String	The Alternate Review Department(s) 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.

Schema Field	Type	Description
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	
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.
dateHired	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.
dateIsterStart	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.
dateIsterEnd	String	
dateIsterUpdate	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.

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to delete.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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.

Request Parameters

Parameter	Type	Description
departmentName	String	Specify the name of the department whose users you want to view.
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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to <i>POST: Create User</i> on page 366.		

GET: Get All User Aliases (getalluseraliases)

Retrieve a list of aliases that are assigned to a specific user.

Request Parameters

Parameter	Type	Description
zlpUserId	String	Specify the ID of the user whose aliases you want to view.
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 Codes

- 206: Partial Content
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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: <ul style="list-style-type: none">0: Default (Email)1: X500 DN2: Address3: Manual4: Exchange Legacy DN5: Transformed Lotus DN6: NetBios User Name100: IM200: Bloomberg300: 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.

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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.

GET: Get User Using Alias (getuserusingalias)

Retrieve information for a user by specifying an alias assigned to the user.

Request Parameters

Parameter	Type	Description
alias Address	String	Specify an alias assigned to the user in ZL UA.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.

GET: Get User Using External Reference (getuserusingexternalreference)

Retrieve information for a user by specifying the external reference data assigned to the user.

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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.

GET: Get User Using Owner (getuserusingowner)

Retrieve information for a user by specifying the owner assigned to the user.

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 Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.

GET: Get User Using ZLP ID (getuserusingid)

Retrieve information for a user by specifying the ZLP ID assigned to the user.

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ZLP user ID assigned to the user in ZL UA.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.

PUT: Move User to a New Department (moveusertonewdepartment)

Move a user to a new department.

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.
newDeptInfold	Integer	Specify the ID of the department you want to move the user to.
newReviewDeptInfold	Integer	Specify the ID of the review department you want to assign the user to.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.

PUT: Remove User Alias (removealias)

Remove a user alias.

Request Parameters

Parameter	Type	Description
zlpUserId	Integer	Specify the ZL user ID of the user you want to modify.
alias	String	Specify the alias you want to remove.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

Response Schema Fields

A string indicating whether the alias was removed successfully.

PUT: Restore Terminated User (restoreterminateduser)

Restore a terminated user to active status.

Request Parameters

None.

Request Body Schema Fields

Schema Field	Type	Description
zipUserId	Integer	Specify the ZL user ID of the user you want to restore.
restoreAddress	Boolean	Set to True to restore the primary email address previously assigned to the user.
restoreAvailableAliases	Boolean	Set to True to restore the aliases previously assigned to the user.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.

PUT: Terminate User (terminateuser)

Terminate an active user.

Request Parameters

Parameter	Type	Description
zipUserId	Integer	Specify the ZL user ID of the user you want to terminate.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.

PUT: Update User Account Information (updateuseraccountinfo)

Update a user’s account information.

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.
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 Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.

PUT: Update User Email Address (updateuseremailaddress)

Update a user's primary email address.

Request Parameters

Parameter	Type	Description
zipUserId	Integer	Specify the ZL user ID of the user you want to modify.
address	String	Specify the new primary email address to assign to the user.

Request Body Schema Fields

None.

Response Codes

200: Successful
400: Bad Request
401: Unauthorized
403: Forbidden
404: Not Found
500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.

PUT: Update User Mail Server Information (updateusermailserverinfo)

Update the mail server information associated with a user.

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.
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 Codes

200: Successful

400: Bad Request

401: Unauthorized

403: Forbidden

404: Not Found

500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.

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.

Request Parameters

Parameter	Type	Description
zipUserId	String	Specify the ZL user ID of the user you want to update.
owner	String	Specify the name of the new owner.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.

PUT: Update User Sync Status (updateusersyncexclude)

Update a user’s user synchronization status. This determines whether the user will be modified during user synchronizations.

Request Parameters

Parameter	Type	Description
zlpUserId	String	Specify the ZL user ID of the user you want to update.
fSyncExclude	Boolean	Set to True to exclude the user from being updated during user LDAP user synchronizations.

Request Body Schema Fields

None.

Response Codes

- 200: Successful
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

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 when the user was created or last updated. For descriptions of these fields, refer to *POST: Create User* on page 366.