DocuSign®

APEX CLM SDK

Unmanaged Package

Maintained by DocuSign Technical Services

Getting Started

This guide will help you get started using the Apex CLM SDK. The guide will cover the following topics:

Overview

The APEX CLM SDK can be used to quickly write code to integrate seamlessly with CLM and Salesforce. The package offers classes and methods to expedite authentication as well as calls to some of the most common CLM APIs.

If you are unfamiliar with the CLM APIs, a good starting point is to review the documentation found on our developer site.

API Overview

CLM.CM	https://developers.docusign.com/docs/clm-api/clm.cm/
CLM.DS	https://developers.docusign.com/docs/clm-api/

Object API

CLM.CM	https://developers.docusign.com/docs/clm-api/clm.cm/clm101/object-api
CLM.DS	https://developers.docusign.com/docs/clm-api/clm101/object-api

Task API

CLM.CM	https://developers.docusign.com/docs/clm-api/clm.cm/clm101/task-api
CLM.DS	https://developers.docusign.com/docs/clm-api/clm101/task-api

Content API

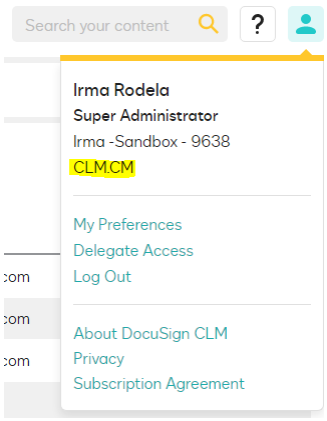
CLM.CM	https://developers.docusign.com/docs/clm-api/clm.cm/clm101/content-api
CLM.DS	https://developers.docusign.com/docs/clm-api/clm101/content-api

Determine CLM Version

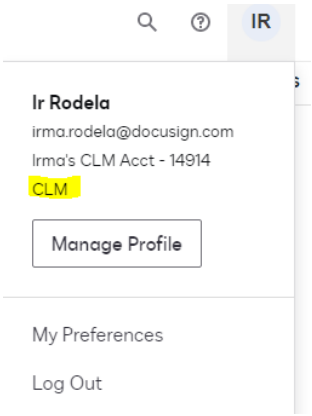
Before moving ahead, take a moment to determine which version of CLM you are using. There are two versions of CLM and each can be referenced by different names.

Both versions can use the SDK but how you authenticate is different. This documentation covers authentication and the data points needed for each of these versions a bit later on.

Legacy SpringCM | CLM.CM



DocuSign CLM | CLM.DS



Install Package

The CLM SDK is an unmanaged package available via the following link:

https://login.salesforce.com/packaging/installPackage.apexp?p0=04t5w000005hiVU

Password: CLMSDKFall2021v1.10

** NOTE: This password changes with each update to the unmanaged package. If you are trying to install this package and the password is not valid, please contact DocuSign's professional services for the latest password.

If you have never installed a package in Salesforce, review this article:
https://help.salesforce.com/articleView?id=distribution_installing_packages.htm&type=5

You will be prompted to install for *admins*, *all users* or *specific profiles*. Choose ***All Users***.

Remote Site Settings

After successful install, please add the following URLs to the *Remote Site Settings* page in Salesforce. From the Setup screen, search for Remote Site Settings.

The [data_center] variable in the URLs below will be your CLM server. To find this, look at the URL of your CLM site or go to *CLM >> Admin >> Integrations >> System Domains* and look at the REST API endpoints. (ex. na11/uatna11, na21, eu11, us11)

CLM.DS UAT

Remote Site Name	Remote Site URL
DSDemoAuth	https://account-d.docusign.com/oauth/token
CLMUATObjectAPI	https://apiuat[data_center].springcm.com
CLMUATDownloadAPI	https://apidownloaduat[data_center].springcm.com
CLMUATUploadAPI	https://apiuploaduat[data_center].springcm.com

CLM.DS Prod

Remote Site Name	Remote Site URL
DSProdAuth	https://account.docusign.com/oauth/token
CLMProdObjectAPI	https://api[data_center].springcm.com
CLMProdDownloadAPI	https://apidownload[data_center].springcm.com
CLMProdUploadAPI	https://apiupload[data_center].springcm.com

CLM.CM UAT

Remote Site Name	Remote Site URL
CLMUATAuth	https://authuat.springcm.com
CLMUATObjectAPI	https://apiuat[data_center].springcm.com
CLMUATDownloadAPI	https://apidownloaduat[data_center].springcm.com
CLMUATUploadAPI	https://apiuploaduat[data_center].springcm.com

CLM.CM Prod

Remote Site Name	Remote Site URL
CLMProdAuth	https://auth.springcm.com
CLMProdObjectAPI	https://api[data_center].springcm.com
CLMProdDownloadAPI	https://apidownload[data_center].springcm.com
CLMProdUploadAPI	https://apiupload[data_center].springcm.com

SDK Folder Structure

The package folder structure is organized into three main folders.

	Description
classes	Root folder and holds service and test classes
Authentication	Authenticator classes for CLM.DS and CLM.CM (Legacy SpringCM) as well as an interface class.
Model	Model classes for every object used in endpoint request and response
Request	Request classes for Document and Folder actions

Classes

Service and Tests Overview

Class Name	Purpose
SpringCMHttpMockResponseGenerator	Used to mock responses from the API for SFDC code coverage
DocusignJWT	Generates JWT for CLM.DS authentication method
SpringCMHTTPRequestor	Class with all HTTP method requests
SpringCMFunctionalTests	Used as part of testing the SDK and part of SFDC code coverage.
SpringCMService	Main class and holds methods for each CLM API endpoint
SpringCMTests	Used as part of SFDC code coverage.

Authentication Overview

Class Name	Purpose
SpringCMIAuthenticator	Interface used in each authenticator class
CMAPIUserAuthenticator	Auth class used for CLM.CM
CMSalesforceUserAuthenticator	Auth class used for CLM.CM Salesforce flow
DSAPIUserAuthenticator	Auth class used for CLM.DS

Models Overview

Class Name	Purpose
CLMAccessLevel.cls	
CLMAccount.cls	
CLMActionBy.cls	
CLMAttributeField.cls	
CLMAttributeFieldValue.cls	
CLMAttributeGroup.cls	
CLMAttributeGroupFields.cls	
CLMAttributeGroups.cls	
CLMContact.cls	

CLMContactItems.cls
CLMCopyTasks.cls
CLMDocLauncher.cls
CLMDocLauncherItems.cls
CLMDocLauncherTasks.cls
CLMDocument.cls
CLMDocumentAttributeGroupDetails.cls
CLMDocumentItems.cls
CLMDocumentMergeTasks.cls
CLMDocumentProcessTrackingActivitesIte
ms.cls
CLMDocumentProcessTrackingActivities.cl
s
CLMDocumentReminder.cls
CLMEosInfo.cls
CLMExternalReviewTasks.cls
CLMFolder.cls
CLMFolderArchiveTasks.cls
CLMFolderItems.clsl
CLMFolderSearchTasks.cls
CLMGroup.cls
CLMGroupItems.cls
CLMHistoryItems.cls
CLMLock.cls
CLMPermissionSet.cls
CLMPermissionSetsItems.cls
CLMSearch.cls
CLMSecurity.cls
CLMShareLinks.cls
CLMSharelinksItems.cls
CLMSignatureTasks.cls
CLMSplitDocumentTasks.cls
CLMTokenRequest.cls
CLMUser.cls
CLMUserActions.cls
CLMUserItems.cls
CLMWorkflow.cls
CLMWorkflowQueue.cls
CLMWorkflowQueueItems.cls
CLMWorkItem.cls
CLMWorkItemItems.cls
SpringCMItem.cls

Request Overview

The objects for Folder and Document are handled in special request classes. These request classes have get and set parameters for expanding specific attributes found on each object.

Class Name	Purpose
CLMDocumentRequest	Retrieves the document object and expands various attributes and handle document download
CLMFolderRequest	Retrieves the folder object and expands various attributes

Authentication Methods

There are two authentication classes, one for each version of CLM available. If you are unsure about the version of CLM you are using, refer to the Determine CLM Version section of this document.

CLM.CM (Legacy SpringCM)

The following steps are prerequisites for authentication.

Step 1: Obtain Client ID and Client Secret

CLM.CM uses a Client ID and Client Secret that is generated for each CLM environment. The Client ID and Client Secret can be requested by contacting DocuSign CLM Support. Once requested, an email will be sent asking a CLM Admin to generate the Client Secret via a secure URL. This can only be done once, so please be sure to store the Client ID and Secret in a secure location.

Step 2: Create an API Only user in CLM

- An API User is created in the DocuSign CLM address book and given the desired role and security groups needed for the API calls. Note that once a user is made an API user, it cannot be undone, so actual users of the DocuSign CLM application should never be created as an API user.

- The API user is mapped to a client id in DocuSign CLM Preferences. To create the mapping, navigate to Preferences->REST API in the DocuSign CLM user interface. In the API User Mappings section your client id can be mapped to the API user created.

  * Note that a single Client Id can be mapped to one and only one API User.

Once you have the Client ID and Client Secret, make a note of the following data points that you will need for the CLM.CM authentication method.

Method Parameters

Method Parameter	Description
clientID	Client ID
clientSecret	Client Secret
authURL	Authentication URL (review the remote site setting section)
dataCenter	CLM data center (review the remote site setting section)
version	CLM API Version (v201411 is the current version)

Method Example

```
//method params
String clientID = 'df5f40fb-xxxx-xxxx-xxxx-ace231e4f175';
String clientSecret = '93089dc1ba4d424994axxxxdxxxx3304aLWF';
String authURL='https://auth.springcm.com/api/v201606/apiuser';
String dataCenter = 'na11';
String version = 'v201411';

//set up method
CMAPIUserAuthenticator auth =
new CMAPIUserAuthenticator(clientID,clientSecret,authURL,dataCenter,version);
```

CLM.DS (DocuSign CLM)

CLM.DS uses the same authentication method as the DocuSign eSign API.

If you are unfamiliar with how to set up an API integrator keys or providing consent for your integration user, please review the following documentation:

https://developers.docusign.com/docs/clm-api/clm101/auth/

After successfully setting up an integration key and providing consent for the integration user. Make note of the following data points needed for the CLM.DS authentication method.

Method Parameters

Method Parameter	Description
dsAccountId	API Account ID
iss	Integration Key
aud	Authentication URL (review the remote site setting section)
sub	DocuSign User ID of your integration user
pkey	Private Key
dataCenter	CLM data center (review the remote site setting section)
version	CLM API Version (v2 is the current version

Method Example

```
String dsAccountId = '66c36d0d-d7bf-4155-8214-c56b2af5c1b1';
String iss = '1efbb3f9-c262-4152-bf41-5bfe0dac05b9';
String aud = 'account-d.docusign.com';
String sub = 'cf5394d9-xxxx-xxxx-xxxx-5dd6f743a4fa';
String pkey = 'MIIEowIBAAKCAQExxxxxxxQqJEVE6nztKyaMyOV';
String dataCenter = 'uatna11';
String version = 'v2';

DSAPIUserAuthenticator auth = new
DSAPIUserAuthenticator(dsAccountId,iss,aud,sub,pkey,dataCenter,version);
```

SpringCMService Methods

Document Methods

Method Name	Purpose
DeleteDocument	Deletes a single document
GetDocumentById	Retrieve a document object by the document GUID
GetDocumentByPath	Retrieve a document object by the document path
UploadDocuments	Download a document
CheckOutDocuments	Checkout a document
CancelCheckOutDocuments	Cancel the checkout of a document
CheckInDocuments	Check in a new version of a document
MoveDocument	Move a document to a new folder
UpdateDocument	Update a document's attributes
Search	Search for documents

DocGen Methods

Method Name	Purpose
GetDocLauncherConfigs	Return a collection of docgen configurations
GetDocLauncherTask	Kick off a docgen form

Folder Methods

Method Name	Purpose
GetFolderById	Get a folder by folder GUID
GetFolderByPath	Get a folder by folder path
CreateFolder	Create a new folder
FindOrCreateEOSFolder	Find or create and EOS folder
DeleteFolder	Delete an existing folder

UpdateFolder	Update an existing folder
MoveFolder	Move an existing folder to a new path
FolderSearch	Search for folders

Workflow Methods

Method Name	**Purpose**
StartWorkflow	Kick off a specific workflow
GetWorkflowByHref	Retrieve a workflow to view status
SignalWorkflow	Send a signal to workflow
DeleteWorkflow	Delete a workflow in progress

Code Examples

Below are a few of more common uses of the CLM API. If you don't see an example that fits your use case, please review the *SpringCMFunctionalTests* class.

Authentication CLM.CM

```
//method params
String clientID = 'df5f40fb-xxxx-xxxx-xxxx-ace231e4f175';
String clientSecret = '93089dc1ba4d424994axxxxdxxxx3304aLWF';
String authURL='https://auth.springcm.com/api/v201606/apiuser';
String dataCenter = 'na11';
String version = 'v201411';

//set up method
CMAPIUserAuthenticator auth =
new CMAPIUserAuthenticator(clientID,clientSecret,authURL,dataCenter,version);
```

Authentication CLM.DS

```
//method params
String dsAccountId = '66c36d0d-xxxx-xxxx-xxxx-c56b2af5c1b1';
String iss = '1efbb3f9-xxxx-xxxx-xxxx-5bfe0dac05b9';
String aud = 'account-d.docusign.com';
String sub = 'cf5394d9-xxxx-xxxx-xxxx-5dd6f743a4fa';
String pkey = 'MIIEowIBAAKCAQExxxxxxxQqJEVE6nztKyaMyOV';
String dataCenter = 'uatna11';
String version = 'v2';

//set up method
DSAPIUserAuthenticator auth = new
DSAPIUserAuthenticator(dsAccountId,iss,aud,sub,pkey,dataCenter,version);
```

Service

```
SpringCMService svc = new SpringCMService(auth);
```

Kick-Off Workflow

```
CLMWorkflow wfParam = new CLMWorkflow();
wfParam.Name = 'Test Workflow';
wfParam.Params = '<data><name>Joe Smith</name></data>';
CLMWorkflow wf = svc.StartWorkflow(wfParam);
```

Kick-Off DocGen

```
CLMDocLauncherItems dli = (CLMDocLauncherItems)svc.GetDocLauncherConfigs();


CLMDocLauncherTasks dlt = new CLMDocLauncherTasks();
dlt.Data = 'hello world!';
dlt.DestinationFolder = dli.Items[0].DestinationFolder;
dlt.DocLauncherConfiguration = dli.Items[0];
dlt.DataType = 'XML';
CLMDocLauncherTasks dlr = svc.GetDocLauncherTask(dlt);
```

Create Folder

```
CLMFolder fldParent = svc.GetFolderByPath('/SDK Test Account/My
Folder').getFolder();

CLMFolder fld = new CLMFolder();
fld.Name = 'My New Folder';
fld.ParentFolder = fldParent;

CLMFolder newFolder = svc.CreateFolder(fld);
```

Create EOS Folder

```
string folderName = 'My EOS Folder';
string path = '/SDK Test Account/Docusign Test/My folder';
string objectType = 'Accounts';
string objectId = '001f200001ZCST5';

CLMFolder newFolder =
svc.FindOrCreateEOSFolder(folderName,path,objectType,objectId);
```

Search for Documents by Attributes

```
CLMAttributeField attr = new CLMAttributeField(); attr.GroupName = 'Legal
Approval'; attr.Field = 'Status';
attr.Value = 'Signed';

List mylist = new List();
mylist.add(attr);

CLMSearch request = new CLMSearch();
request.AttributeFields = mylist;

CLMSearch newSearch = svc.Search(request,true);
```

Upload Document

```
String docName = ‘mydoc.docx’;
Blob file = [file blob];
Map<string,blob> documentsToStore = new Map<string,blob>();
documentsToStore.put(docName, file);
CLMDocument uploadDoc = svc.UploadDocuments(folder,documentsToStore,'');
```

Download Document

```
Blob doc = svc.GetDocumentById('{{docID}}').downloadDocument('native');
```

FAQs

Why is this an unmanaged package?

This package is meant to be a bridge to our product team incorporating the CLM APIs into the APEX Toolkit.

How can I update this package?

The package is open source and can be updated by customers, partners or DocuSign Professional Services.

Will DocuSign support and fix bugs?

Yes, *DocuSign Professional Services is the owner of this package and any bugs or questions will be addressed by Professional Services. It is recommended that if a customer or partner is using this package, a support SOW is signed for ease of support.*

How often will this package change?

This package will change as needed to either enhance or fix bugs. Due to the nature of the package being unmanaged, these changes will not be pushed to users of the package.