DOCUSIGN PLUGIN -ESIGNATURE API

Introduction:

The DocuSign Sample plug-in demonstrates how you can send documents from an IBM Enterprise Content Management (ECM) repository DocuSign. Users can then use DocuSign to sign and approve the documents digitally from anywhere and on any device. The JWT Auth grant flow is implemented in the sample.

The plug-in provides an IBM Content Navigator menu action to get the document signature status from DocuSign systems. In addition, the plug-in uses IBM Content Navigator task manager to create asynchronous tasks that periodically update signature status automatically updates or automatically check-in signed documents and update signature status.

DocuSign prerequisites:

To use the DocuSign Sample plug-in, you must have a DocuSign developer account. If you don't have a developer account already, you can create it at https://secure.docusign.com/signup/develop. Please follow the instructions to setup up a docusign App. Store the following in a safe location, as it will be required in later steps:

- IntegrationKey
- UserID of user/admin to be impersonated
- Account ID: Can be gotten from the Admin Icon expanded view (#xxxxxxx)
- Generate RSA Key pairs, Store them in a file, namely; privateKey.txt, publicKey.txt

From the sandbox, you can find the integrator key or application key that is used to connect to your DocuSign account from any application. For information about your integrator key, log in to https://admindemo.docusign.com/api-integrator-key

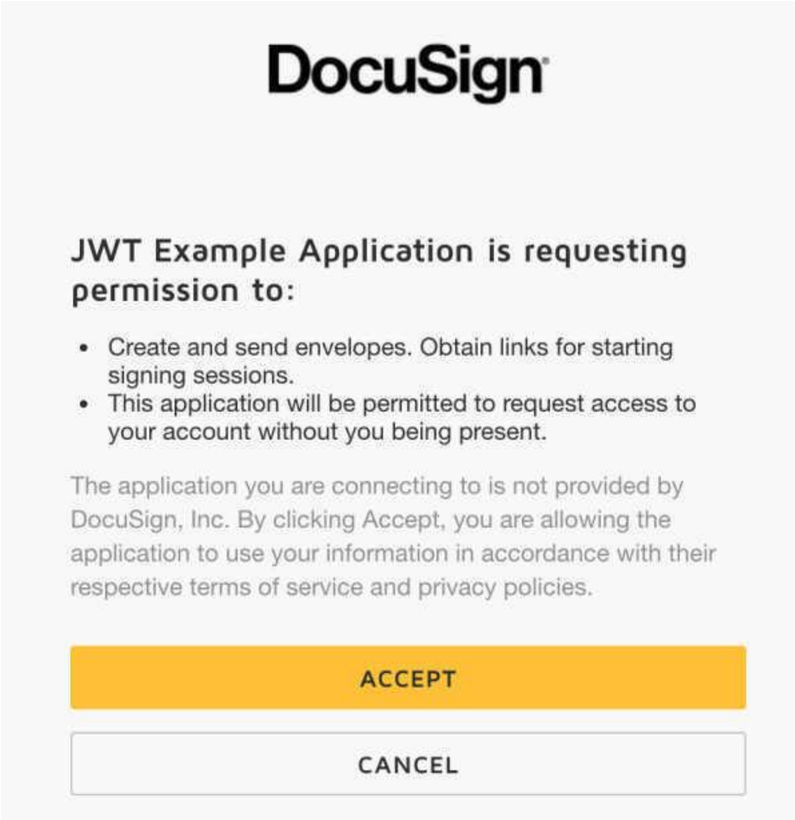
For JWT Authorization Grant prerequisites:

Granting consent individually

Before your client ID (**integration key**) can be used to obtain an access token for a user, the user must grant the client ID consent to do so. Consent is granted to a client ID to obtain one or more permission **scopes** for a particular user.

There are 3 main types; Administrative, Individual and ISV application grant. In this demo, we will be using Individual grant consent. To obtain consent, ask the users who will be impersonated by the JWT grant application to open a particular URL in their browser. You will supply them with the URL to be used during this one-time task.

When the user opens the URL, they will be asked to authenticate with DocuSign. After authenticating, they'll then be asked to grant consent to your application via a form:



The individual consent URL
This Individual consent URL code is a one time grant that requires the user in question to give consent. Use the following URL format to grant consent.
Its format is:
SERVER/oauth/auth?response_type=code &scope=signature%20impersonation&client_id=CLIENT_ID &redirect_uri=REDIRECT_URI

Where
Server: https://account-d.docusign.com
Client ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
Redirect URL: localhost or any URL. One can customize the URL to make UX seamless.

An example of such URL would be:
https://account-d.docusign.com/oauth/auth?response_type=code &scope=signature%20impersonation&client_id=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx&redirect_uri=http://localhost:9080/navigator/

For more information on the other method of giving consent follow the link:
https://www.docusign.com/blog/developers/oauth-jwt-granting-consent

IBM Content Navigator prerequisites:

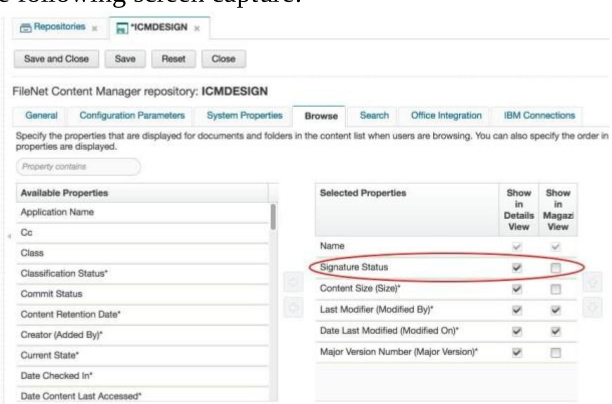
1. Make sure Maven is installed.
2. Copy all the dependencies into a lib directory.
   a. navigatorAPI.jar
   b. com.ibm.ws.prereq.commonj-twm.jar
   c. ICNTask.jar
   d. log4j.jar
   e. taskManagerAPI.jar taskServiceClient.jar

FileNet P8 prerequisites:
Before you deploy and use the plug-in, make the following changes to the document metadata in FileNet P8 using Administrative Console for Content Engine (ACCE):

a. Copy the following JAR file into the lib directory Jace.jar
b. Create an integer choice list with following settings:
    i. Symbolic Name: DSSignatureStatusChoiceList
    ii. Type : Integer iii. Options :
        a. None: 0
        b. Draft: 1
        c. Sent: 2
        d. Completed: 3
        e. Checkedin: 4
        f. Voided: 5
c. Create a property definition called DSSignatureStatus with the following settings:
    Display Name : Signature Status
    Symbolic Name : DSSignatureStatus
    Type : Integer
    Associate the DSSignatureStatus property definition with the DSSignatureStatusChoiceList choice list and set the default value to 0.
d. Create a property definition called DSEnvelopeID with the following settings:
    Display Name : Envelope ID
    Symbolic Name : DSEnvelopeID
    Type : String
e. Add the DSSignatureStatus property definition to the Document class.
f. Add the DSEnvelopeID property definition to the Document class as a hidden property.
g. Copy DocuSignPlugin.jar to /opt/IBM/ECMClient/configure/explodedformat~~/taskManager/taskManagerWeb/WEB-INF/dro~~pins
    The above mentioned path will be different for Windows.
h. Restart WebSphere.

Building the JAR & Setting up the DocuSign Sample plug-in IBM Content Navigator

1. Download the Docusign repository
2. In the directory containing the pom.xml, run "mvn clean package"
3. Upon successful build, you can find the docuSignPlugin.jar
4. In the IBM Content Navigator, open the administration tool.
5. Configure your repository to show the Signature Status property in the Details View as shown in the following screen capture:



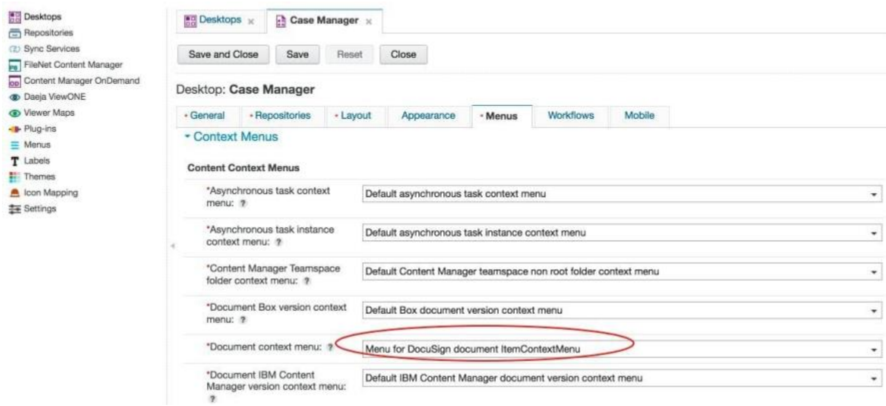
2. Log in to your DocuSign developer account (https://secure.docusign.com/signup/develop).

3. Create a DocuSign template as described in Creating a DocuSign template.

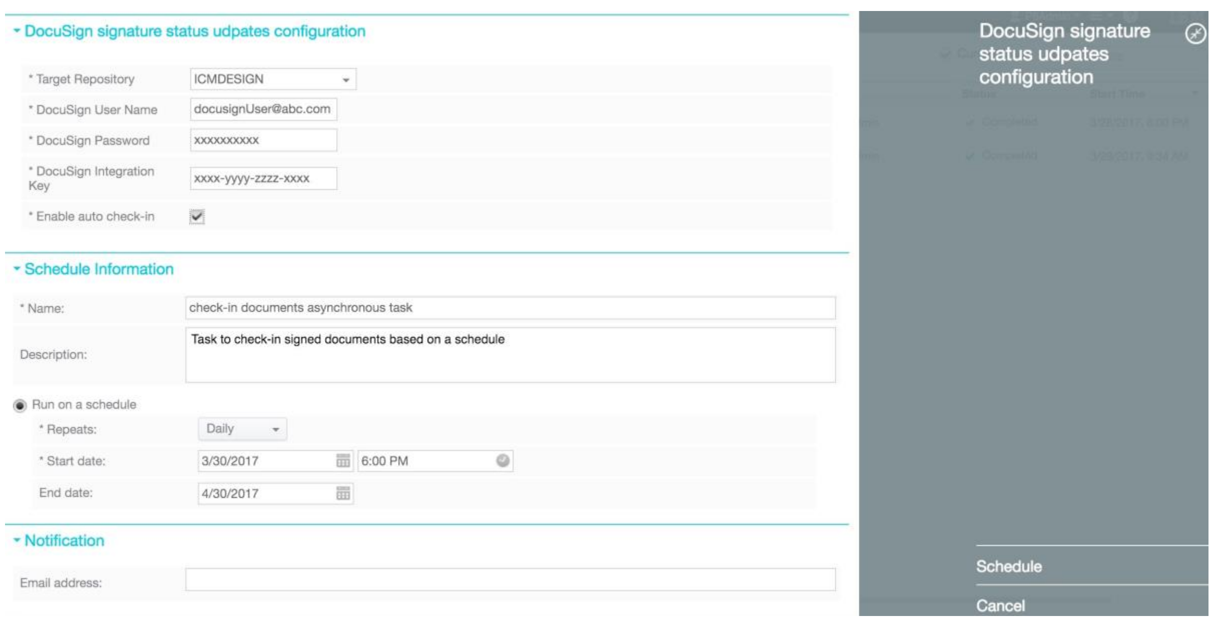
4. In the IBM Content Navigator administration tool, register and configure DocuSign Sample plug-in. You must provide the DocuSign Integration Key, UserID, AccoundID and location of the private key file as part of the configuration.

5. Configure the IBM Content Navigator menu actions:

   a. In the IBM Content Navigator administration tool, open the desktop that you want to configure.
   b. On the Menus tab, change the value for the Document Context Menu field to "Menu for DocuSign document itemContextMenu" as in the following screen capture:

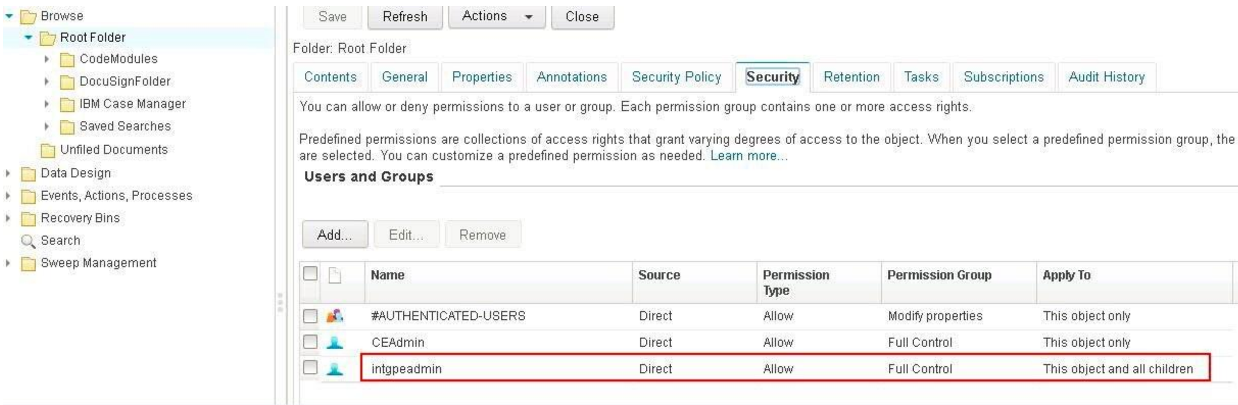


6. Click 'Open Asynchronous Tasks View' on Navigator features pane and create a new task 'DocuSign signature status updates configuration' from 'Schedule Task' drop down list. You can schedule this task by providing the following details. If you want auto check-in of signed documents, select 'Enable auto check-in' checkbox. If you want only the signature status to be updated, then do not 'Enable auto check-in' checkbox.



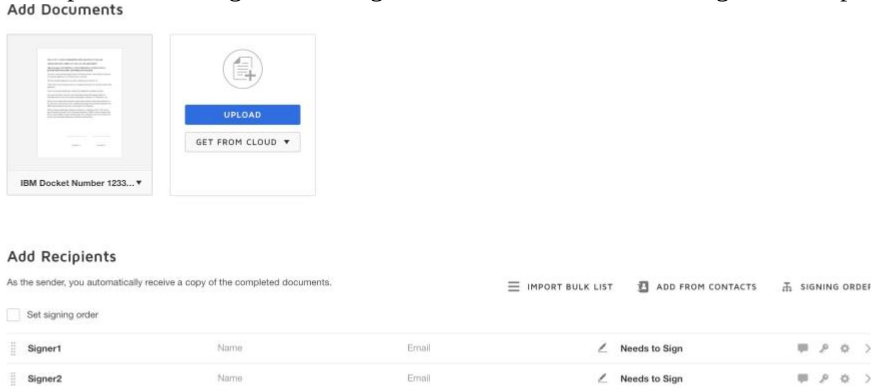
7. Provide appropriate security access to the hidden "DocuSignFolder" folder that the plugin creates for staging the documents. Provide Full Control (or File-in/Un-file) security access permissions on root folder level using ACCE for all the users who would be sending the signature requests to DocuSign.

Below snapshot shows the security configuration to add "intgpeadmin" user.

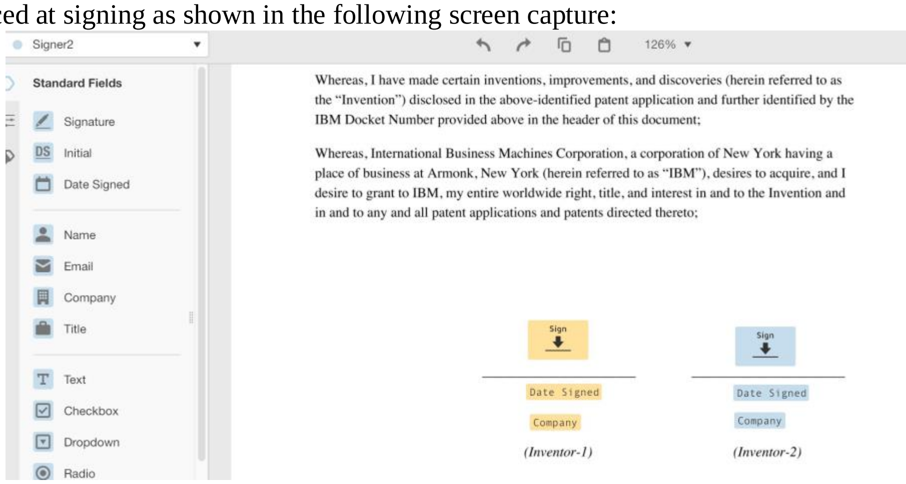


Creating a DocuSign template:

1. On the TEMPLATES tab, select the New option to create a new template.
2. Provide a template name and upload the sample document IBM Docket Number 123344 Assignment Declaration.pdf from the Box share folder for the sample plug-in.
3. Provide a placeholder Signer1 and Signer2 as shown in the following screen capture and click Next:



4. Drag and drop the required Sign tabs from the palette onto the document where the signatures are to be placed at signing as shown in the following screen capture:



5. Click Save and Close the template