

Integrating Adobe Sign in Mendix

Hello Readers,

I hope you all are doing good. This is my first ever blog, I am so excited and happy to share my knowledge & experience with all the makers around the world.

First off, let me introduce myself. My name is **Surya Selvakumar**. I am an **Intermediate Certified Mendix Consultant**. Now, let's dive into the blog!

What is Adobe Sign ?

Adobe Sign, also known as Adobe Acrobat Sign, is a cloud-based service that allows users to send, sign, and manage electronic documents. It provides a digital alternative to paper and ink, enabling individuals and businesses to sign documents online from anywhere, on any device, which speeds up processes by eliminating the need for printing, scanning, and faxing. Key features include creating templates, sending documents for signature, tracking progress, setting reminders, and creating workflows for multiple signers .

Why Integrating with Mendix? Unleashing the Potential of Low-Code Development

Mendix empowers businesses to rapidly develop and deploy applications with minimal coding. Its user-friendly interface and robust functionality make it an ideal choice for organizations seeking agile solutions. With Mendix, businesses can create feature-rich applications tailored to their unique needs, all while ensuring scalability and innovation.

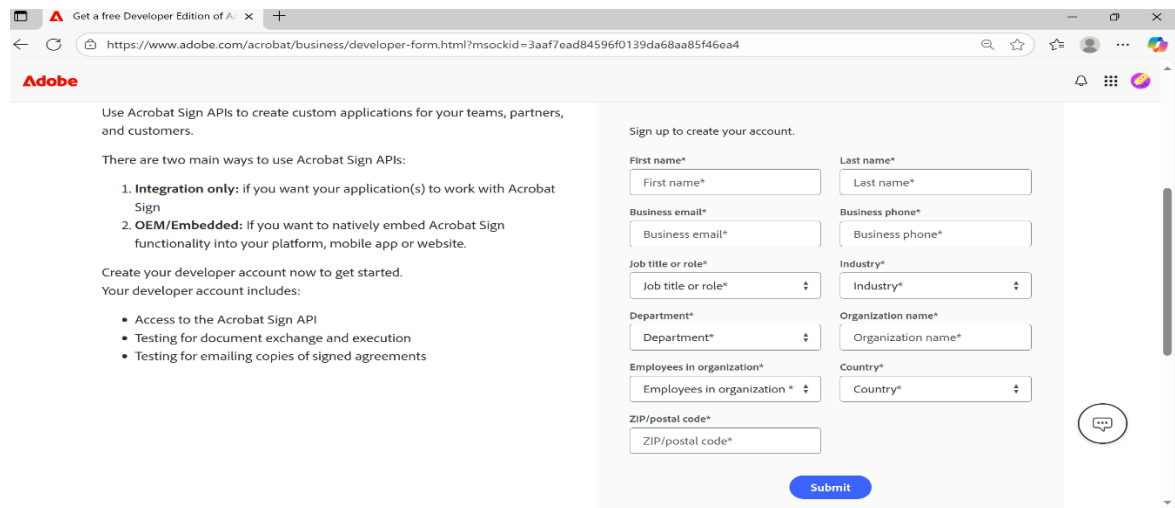
Perquisites:

- **Adobe Sign Account:** Sign up for a Adobe sign account and obtain your API key.

Adobe sign Configurations:

Log in to Adobe Acrobat sign to create account.

- Go to Developer Portal and fill in the required details.
<https://www.adobe.com/acrobat/business/developer-form.html?msocid=3aaf7ead84596f0139da68aa85f46ea4>

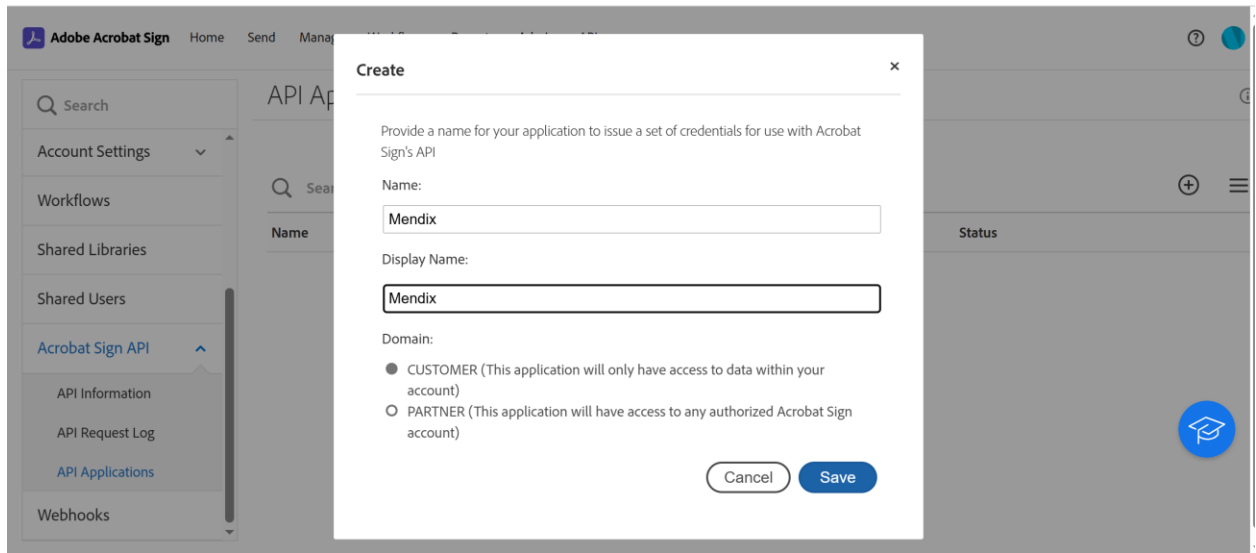


The screenshot shows a web browser window with the Adobe Acrobat Sign Developer Form. The URL in the address bar is <https://www.adobe.com/acrobat/business/developer-form.html?msocid=3aaf7ead84596f0139da68aa85f46ea4>. The page has a light gray background with the Adobe logo in the top left. On the left side, there is a text area explaining the use of Acrobat Sign APIs and providing two main ways to use them: 'Integration only' and 'OEM/Embedded'. It also mentions creating a developer account and lists the benefits: access to the Acrobat Sign API, testing for document exchange and execution, and testing for emailing copies of signed agreements. On the right side, there is a sign-up form titled 'Sign up to create your account.' with various input fields: First name*, Last name*, Business email*, Business phone*, Job title or role*, Industry*, Department*, Organization name*, Employees in organization*, Country*, and ZIP/postal code*. A blue 'Submit' button is at the bottom right of the form.

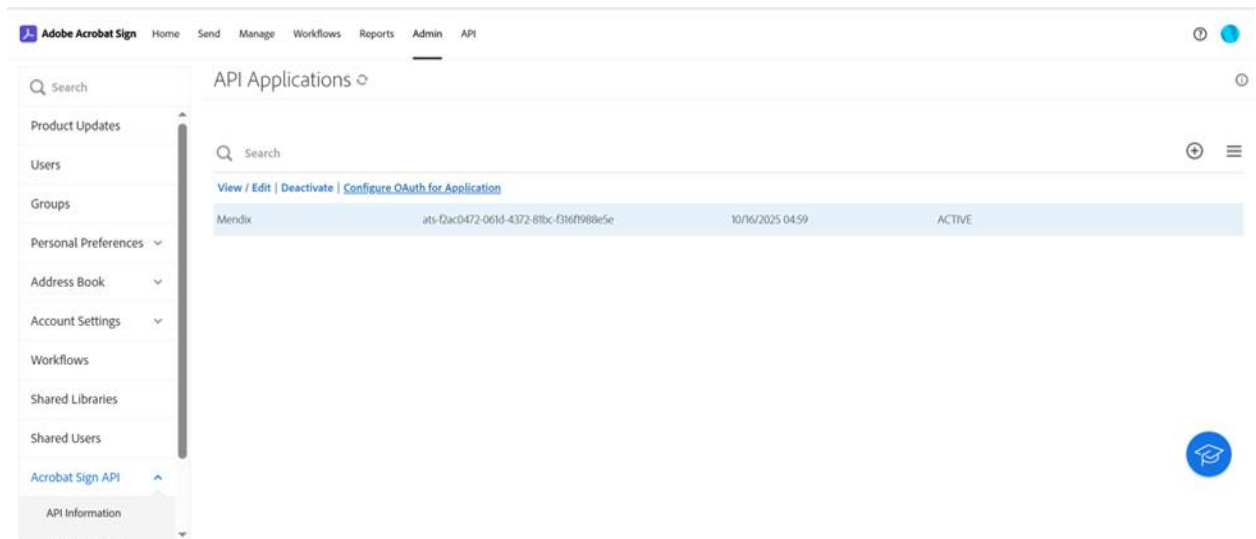
- After entering the details Click “**Submit**”. You will receive confirmation mail entered mail address and activate the account by setup password.
- After created the new password. Log in to your Adobe Sign account and navigate to API section.

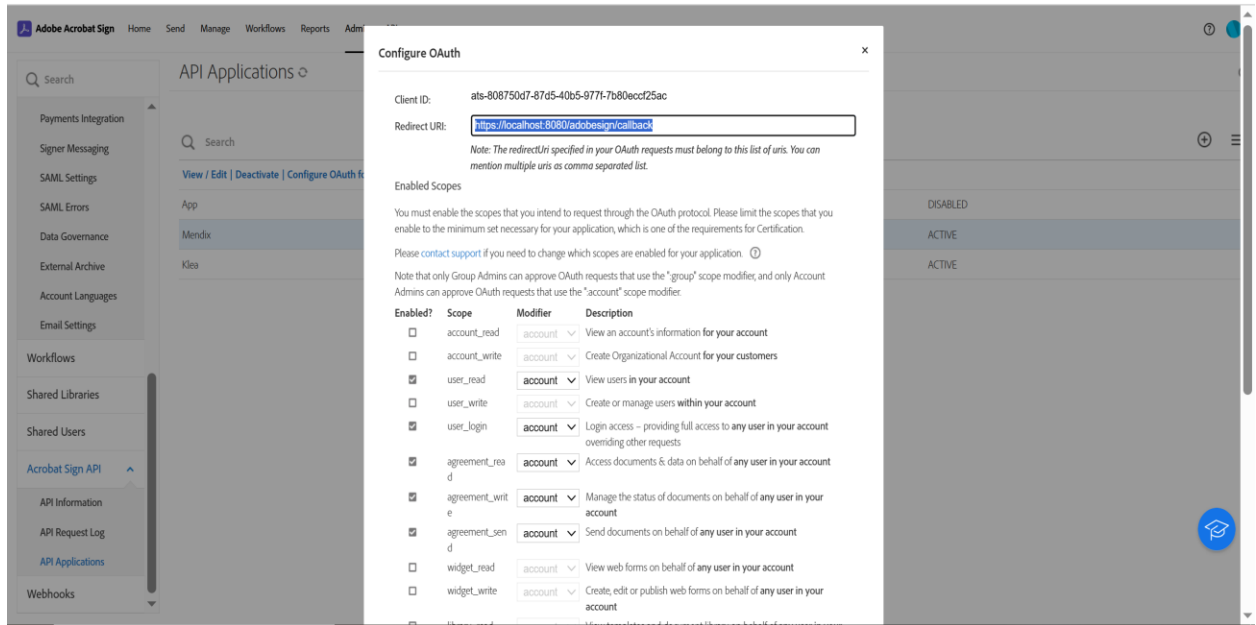
Go to <https://secure.na3.adobesign.com/public/login>

- Create an API Application and enter the following details and click save.



- After creating the API application, click on Configure OAuth for Application and fill in the details like the redirect URL and select the required scopes.





- Store the client ID and client secrets in notes.

Call Back URL :

A callback (also called Redirect URI) is a URL endpoint in your Mendix app that Adobe Sign calls after a user successfully logs in or after a signing event occurs. The callback ensures secure and automated communication between Adobe Sign and Mendix without needing manual steps.

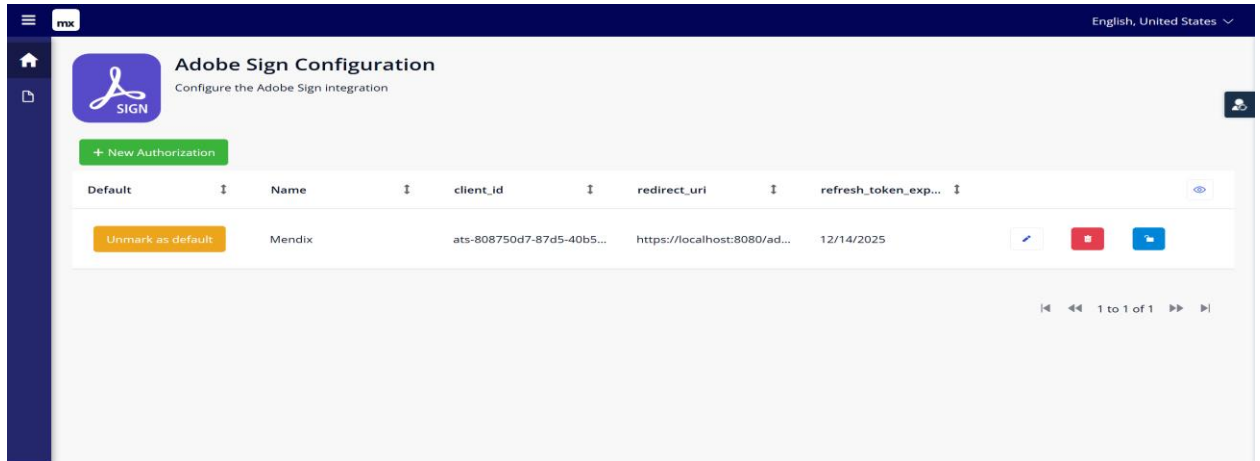
It allows Mendix to:

- Receive authorization codes or tokens (in OAuth login flow)
- Receive real-time notifications about agreements (e.g., document signed, completed, canceled)
- Update your Mendix database automatically
- Example : <https://localhost:8080/adobesign/callback>

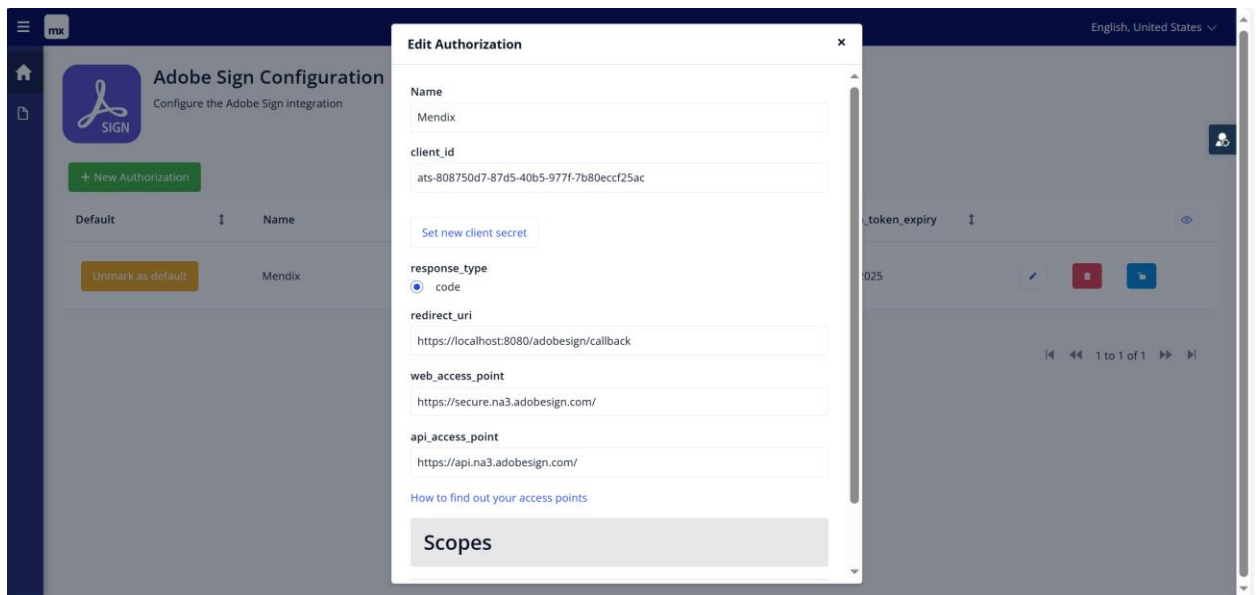
Mendix Configurations:

- Create an account in mendix.
- Create new blank application with any mendix version based on your requirements.

- Download “**Adobe Sign Connector, Adobe Sign commons ,Encryption, Nanoflow commons, community commons**” modules in mendix marketplace.
- Call “**AdobeSign_Configuration**” in navigation already exists in Adobe Sign connector module. Use that page to fill in the required Adobe Sign API details.



- Click the **New Authorization** and fill in the required details .



- Need to take web_access_point and API_access_point. Click the link below.
<https://secure.na3.adobesign.com/public/docs/restapi/v6#/baseUri>

baseUri Base URIs

GET /baseUri Gets the base uri to access other APIs. In case other APIs are accessed from a different end point, it will be considered an invalid request.

Parameters

Cancel

Name	Description
Authorization string (header)	An OAuth Access Token with any of the valid scopes in the format 'Bearer {accessToken}'.

Authorization

Execute Clear

Responses

Curl

Request URL

Server response

Code	Details
200	Response body <pre>{ "apiAccessPoint": "https://api.na3.adobe.com/", "webAccessPoint": "https://secure-na3.adobe.com/"}</pre> Response headers <pre>cache-control: no-cache, no-store, must-revalidate content-length: 104 content-type: application/json; charset=UTF-8 date: Thu, 16 Oct 2025 06:27:30 GMT expires: 0 pragma: no-cache server: Istio-envoy strict-transport-security: max-age=31536000; includeSubDomains;</pre>

- Fill the required scopes and click save button .

Edit Authorization

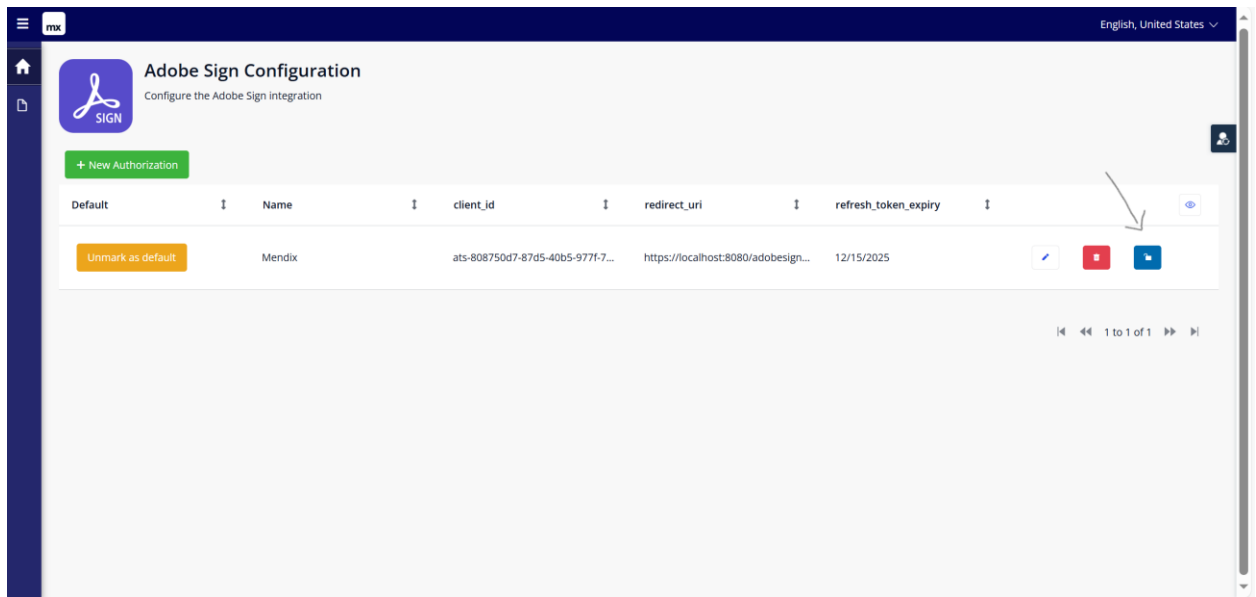
Scopes

+ New Edit Delete

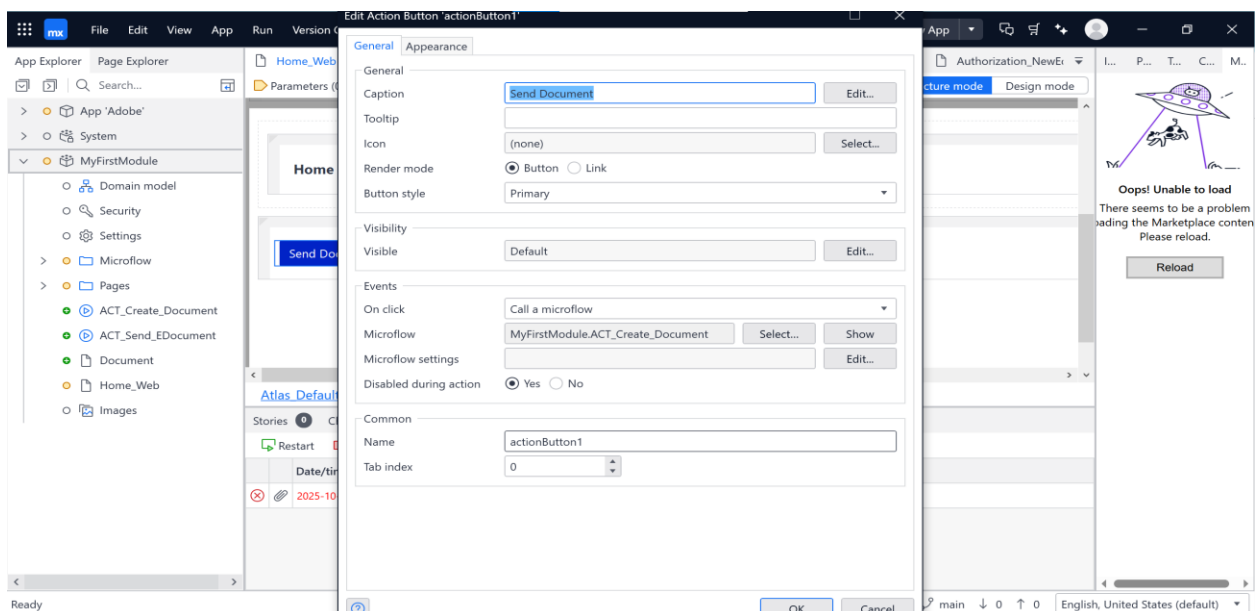
1 to 10 of 10

Scope	Modifier
user_read	account
user_login	account
agreement_read	account
agreement_write	account
agreement_send	account
webhook_read	account
webhook_write	account
webhook_retention	account
application_read	account
application_write	account

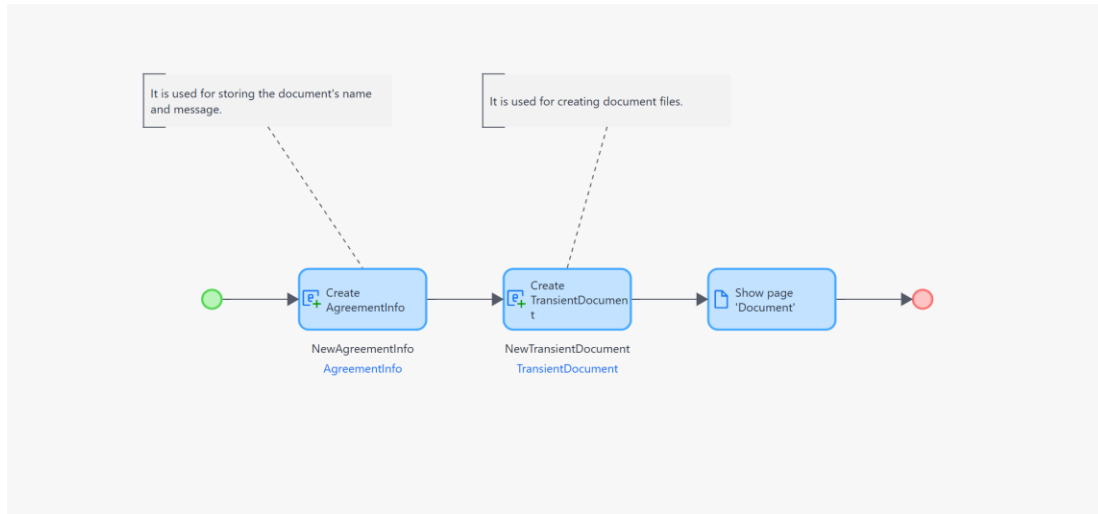
- After saving the required details, click the lock button. It navigates to the Adobe Sign login page. Once the login is completed and all required permissions are granted, the system automatically retrieves the user access token, calls back the Mendix app, and stores the access token in the Mendix database.
- Call back microflow already exist in Adobe sign connector module use that flow to retrieve user access tokens. **“CWS_AccessToken_Get”**



- Create Home page and call in Navigation.
- Create new action button in Home Page to send documents and call new microflow.



- In the newly created microflow add three activities .



- Create Document page – Place 2 Data view containers and call the data source entity. One is TransientDocument and another is an Agreementinfo entity.

Auto-fill

[TransientDocument from page parameter 'TransientDocument' (TransientDocument)]

[AgreementInfo from page parameter 'AgreementInfo' (AgreementInfo)]

Document details

Name
[name]

Message
[message]
(5 lines)

Add recipients

[RecipientInfo, by microflow 'DS_Create_recipient_Document']

Manual - 9

Recipient Type [RecipientType] Add

Name [Name]

Email [Email]

Private message [PrivateMessage]
(5 lines)

Manual - 3

Auto-fill

Selected recipients

Data grid 2
[ParticipantSetInfo, over association 'ParticipantSetInfo_AgreementInfo']

Place widgets like filter widget(s) and action button(s) here

order	role	name	
[order]	[role]	[name]	Place filter widget here

Auto-fill

Selected Ccs

Data grid 2
[Cc, over association 'Cc_AgreementInfo']

Place widgets like filter widget(s) and action button(s) here

email	
[email]	Place filter widget here

Auto-fill

Selected recipients

Data grid 2

[ParticipantSetInfo, over association 'ParticipantSetInfo_AgreementInfo']

Place widgets like filter widget(s) and action button(s) here

order	role	name	
=	v	Ab	Place filter widget here
[order]	[role]	[name]	
[order]	[role]	[name]	
[order]	[role]	[name]	
[order]	[role]	[name]	
[order]	[role]	[name]	

Auto-fill

Selected Ccs

Data grid 2

[Cc, over association 'Cc_AgreementInfo']

Place widgets like filter widget(s) and action button(s) here

email	
Ab	Place filter widget here
[email]	
[email]	
[email]	
[email]	
[email]	

Add file

Browse...

Cancel
Send E-Signature

- Place another data view to store recipient and Carbon Copy information. Call the newly created Microflow as a data source.

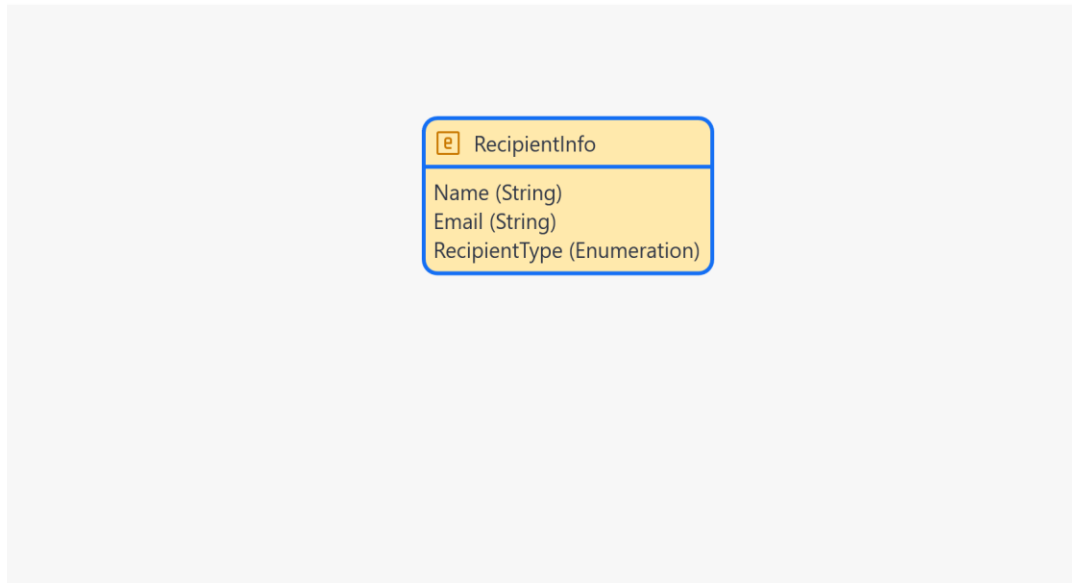
The screenshot shows the Adobe Experience Manager (AEM) interface. The 'Edit Data View' dialog for 'dataView2' is open, displaying the following configuration:

- General** tab is selected.
- Label width (weight)**: 3
- Show footer**: ☒ No (data view is nested)
- Empty entity message**: (empty text box)
- Data source**:
 - Type: ☒ Context ☒ Microflow ☐ Nanoflow ☐ Listen to widget
 - Microflow: MyFirstModule.DS_Create_recipient_Document (with Select... and Show buttons)
 - Microflow settings: (empty text box with Edit... button)
- Editability**:
 - Editable: ☒ Default ☐ Never ☐ Conditionally
 - Read-only style: ☒ Control ☐ Text
- Visibility**:
 - Visible: Default (with Edit... button)
- Common**:
 - Name: dataView2
 - Tab index: 0

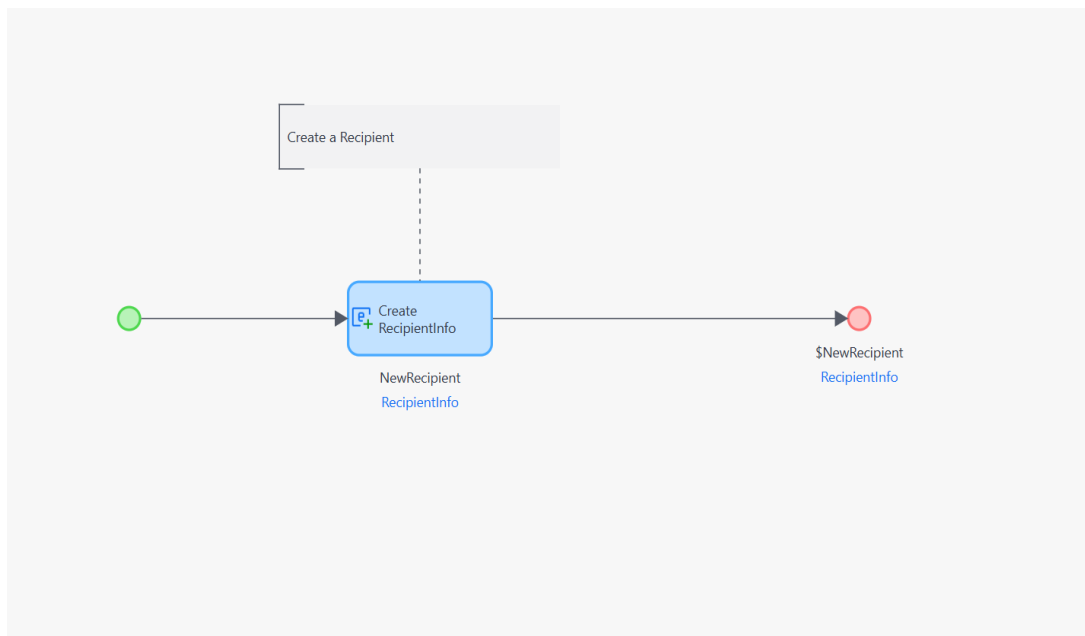
The background shows the AEM console with a log table:

Date/time	Log node
2025-10-16 14:51:44.926	LicenseService
2025-10-16 14:51:44.953	Core
2025-10-16 14:51:45.035	TaskQueue
2025-10-16 14:51:45.035	TaskQueue

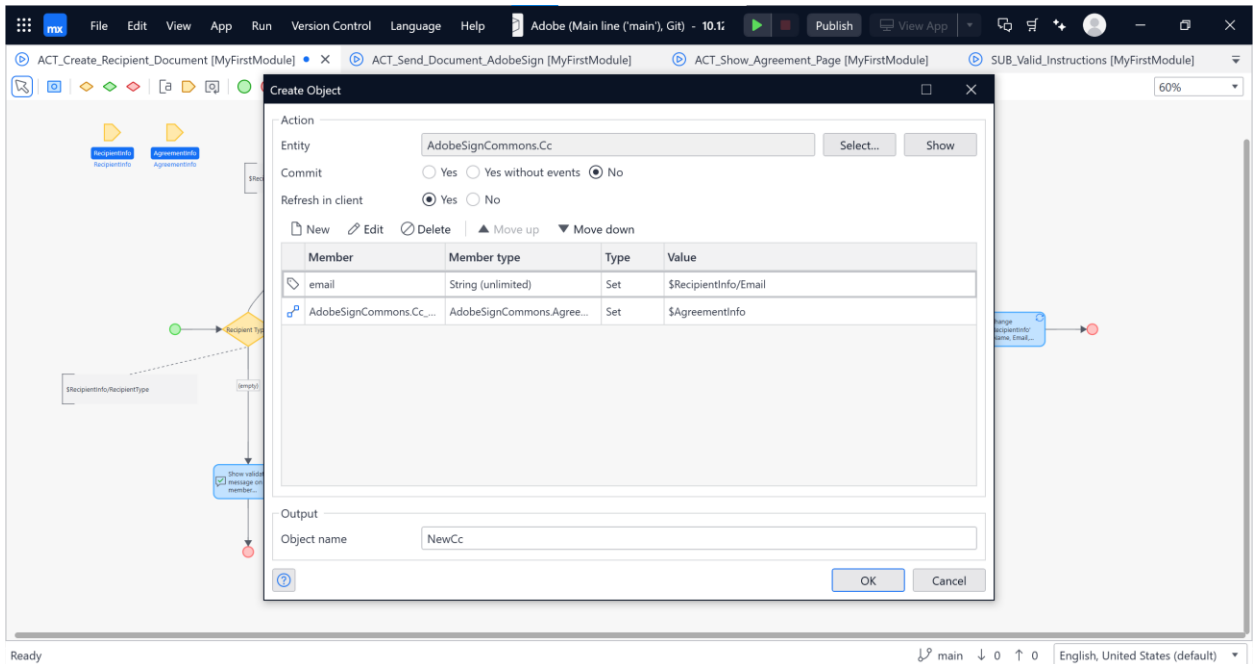
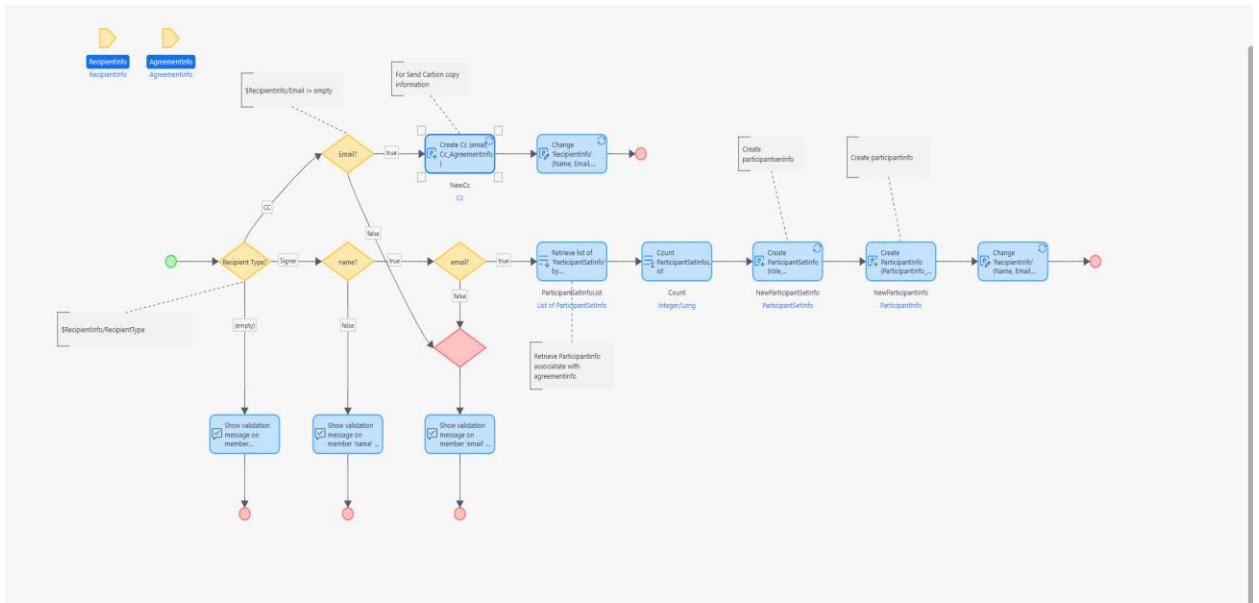
- I have created a new non presistable entity in Adobe Sign Commons module to store the recipient information like signer or Carbon copy.
- Enumeration contains 2 value – Signer, CC



- In the newly created DS microflow add one activities



- Create a new action button '**ADD**' on the document page to add the recipient to sending document and call a new Microflow.



Adobe (Main line ('main'), Git) - 10.1.2

ACT_Create_Recipient_Document [MyFirstModule] X ACT_Send_Document_AdobeSign [MyFirstModule] ACT_Show_Agreement_Page [MyFirstModule] SUB_Valid_Instructions [MyFirstModule]

Create Object

Action

Entity: AdobeSignCommons.ParticipantSetInfo

Commit: ☐ Yes ☐ Yes without events ☒ No

Refresh in client: ☒ Yes ☐ No

Member table:

Member	Member type	Type	Value
role	Enumeration 'ENUM_Particip...	Set	AdobeSignCommons.ENUM_ParticipantSetInfo_role.SIGNER
AdobeSignCommons.Part...	AdobeSignCommons.Agree...	Set	\$AgreementInfo
name	String (unlimited)	Set	\$RecipientInfo/Name
order	Integer	Set	\$Count+1
privateMessage	String (unlimited)	Set	\$RecipientInfo/PrivateMessage

Output

Object name: NewParticipantSetInfo

OK Cancel

Ready

main 0 English, United States (default)

ACT_Create_Recipient_Document [MyFirstModule] X ACT_Send_Document_AdobeSign [MyFirstModule] ACT_Show_Agreement_Page [MyFirstModule] SUB_Valid_Instructions [MyFirstModule]

Create Object

Action

Entity: AdobeSignCommons.ParticipantInfo

Commit: ☐ Yes ☐ Yes without events ☒ No

Refresh in client: ☐ Yes ☒ No

Member table:

Member	Member type	Type	Value
AdobeSignCommons.Part...	AdobeSignCommons.Particip...	Set	\$NewParticipantSetInfo
email	String (200)	Set	\$RecipientInfo/Email
name	String (200)	Set	\$RecipientInfo/Name

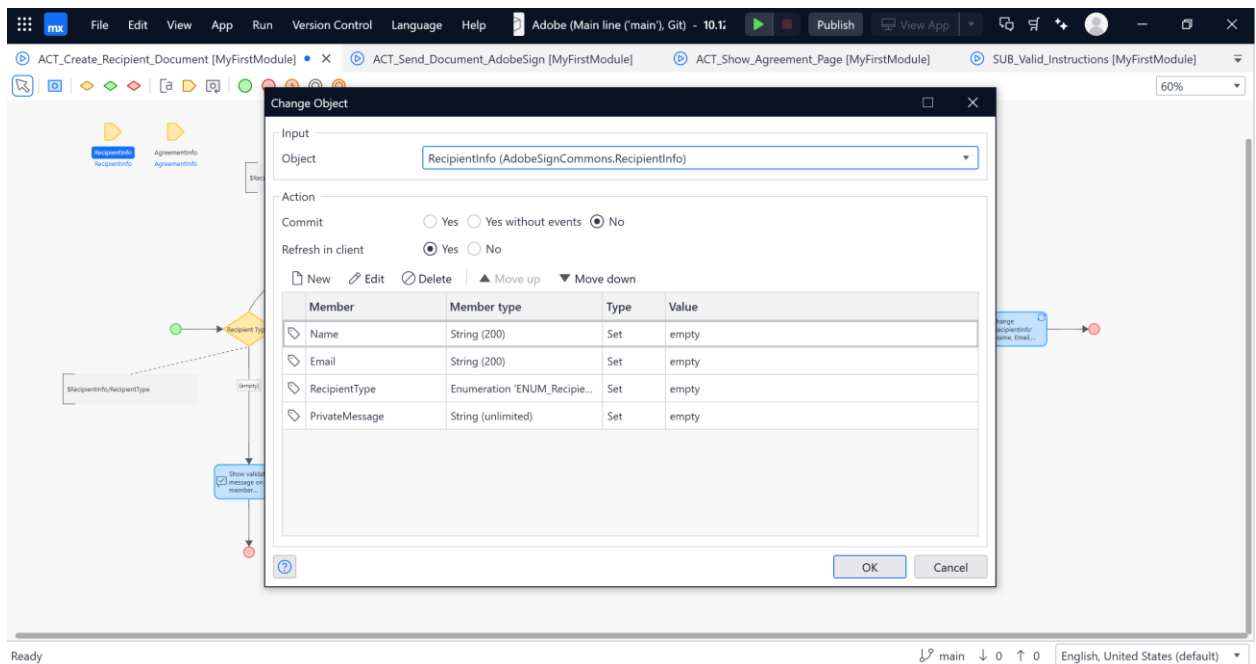
Output

Object name: NewParticipantInfo

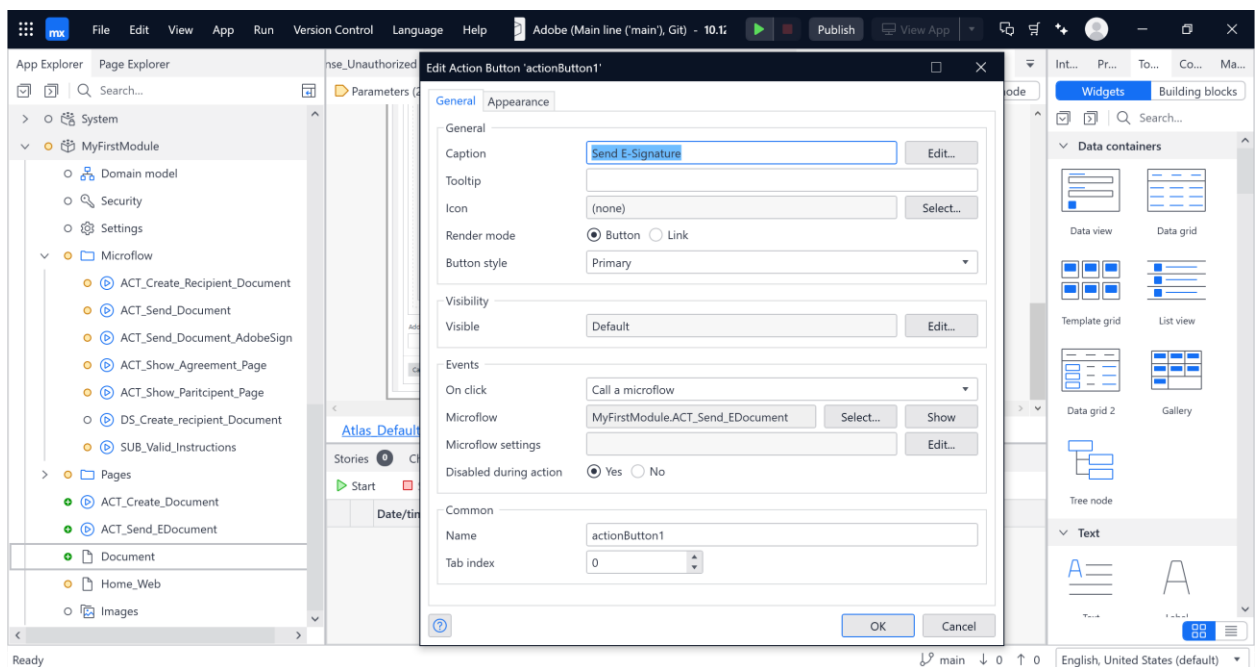
OK Cancel

Ready

main 0 English, United States (default)

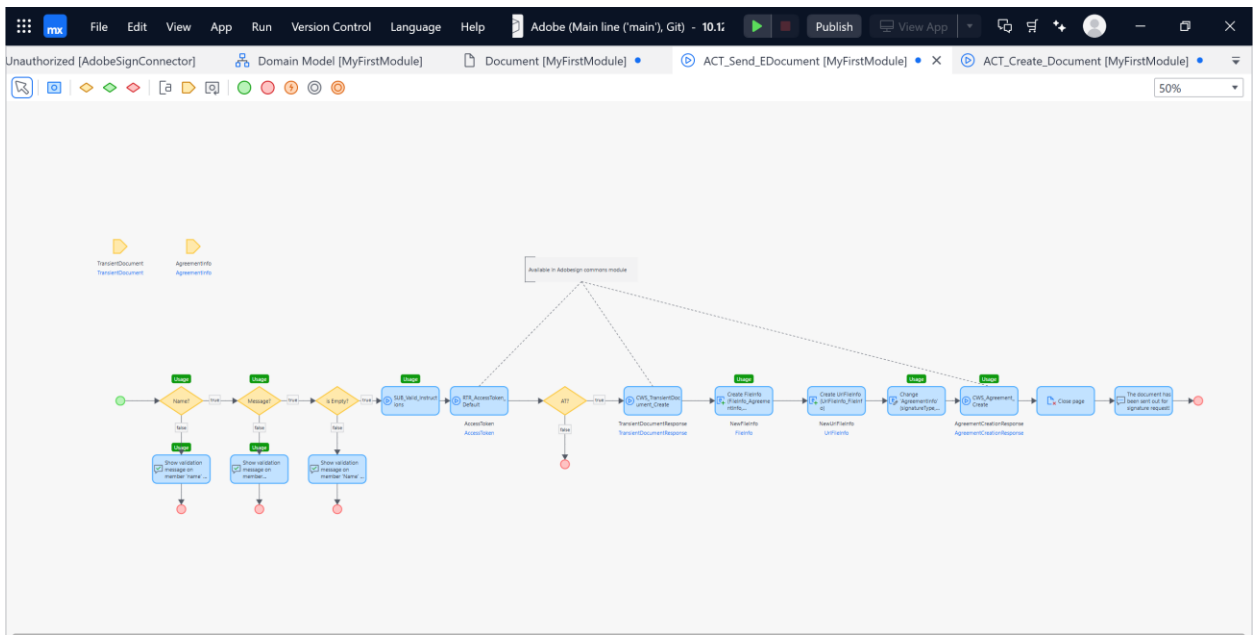


- Use **File Manager** to select the document to send an e-signature request.
- Create a new action button '**Send Document**' on the document page to send documents for e-signature and call a new Microflow.

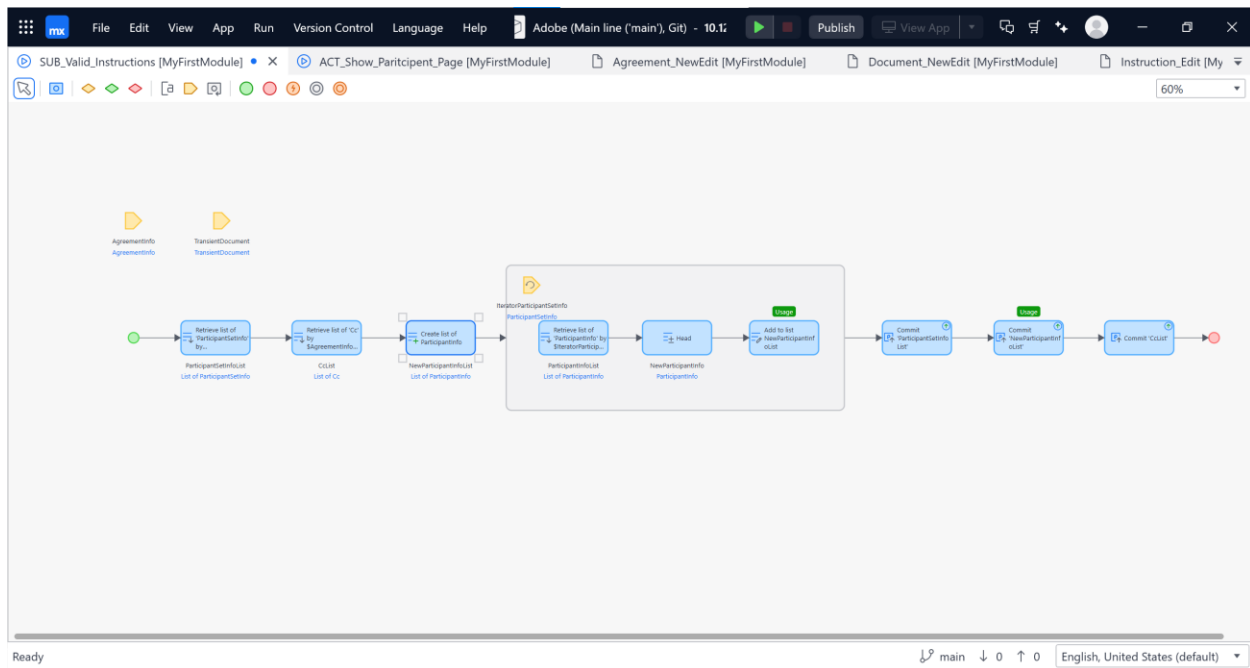


- Sub microflow "**RTR_AccessToken_Default**", "**CWS_TransientDocument_Create**",

“CWS_Agreement_Create” already exist in Adobe sign commons module use that flow to send the document for e-signature.



- Sub microflow **"SUB_Valid_ParticipantInfo"** to save the participant details.



- Complete all the above steps Document has successfully sent recipient email for digital signature.

Now, you can send documents for e-signature seamlessly, enhancing the user experience and streamlining your business processes.

Thank you for your time and exploration of the Adobe-Mendix integration.