Workshop step-by-step instructions

Step 1 - create the eSignature template

Step 1A - log in to the developer account

- Go to https://account-d.docusign.com/ and log in to the Docusign Developer account with the correct credentials that were assigned to you.

Step 1B - use JSON file to create new eSignature template

- Once logged in to the account, select **Templates** from the top level menu.
- Select the **Start** menu from the top-left and open it.
- Under **Envelope Templates** select **Upload Template**.
- In the file picker, select the JSON file provided to you for the eSignature template.
- Click **Open**.
- You should see the new template in the list.

Step 2 - create the Web Form

Step 2A - upload the Web Form configuration to create the Web Form

- Go to https://account-d.docusign.com/ and log in to the Docusign Developer account.
- Once logged in to the account, select **Templates** from the top level menu.
- Select the **Start** menu from the top-left and open it.
- Under **Web Forms** select Upload **Web Form**.
- In the file picker, select the JSON file provided to you for the Web Form.
- A pop-up dialog **Web form template copy created** will show, dismiss it.

Step 2B - activate and publish the new Web Form

- Select the **Activate** button at the top-right of the screen.
- A pop-up dialog will appear with **Public** already selected, click **Activate**.
- **Web Forms successfully activated!** pop-up will show at this point.

Step 3 - create the Maestro Workflow

Step 3A - create a new workflow and decide how it will be triggered

- Go to https://account-d.docusign.com/ and log in to the Docusign Developer account.
- Once logged in to the account, select **Agreements** from the top level menu.
- Select **Maestro Workflows** from the left menu.
- Select **Create Workflow** button from the top-right corner.
- From the pop-up select the **Build Your Own** button.
- Select the **Add workflow start** button at the middle of the screen.
- From the dropdown menu select **From an API Call**.
- Select **Next**.
- Select **Next** (a second time).
- Select the **Apply** button.

Step 3B - add the Web Form step and configure it

- Select the **Add a step** button.
- Select **Collect Data with Web Forms** from the list**.**
- Select the **Configure** button.
- From the **Choose form** dropdown, select the form you created in step 2.
- Select **Next**.
- Select the **Select a participant** dropdown.
- Select the **Add Participants** button.
- In the pop-up select **Add Another Participant**.
- Under **Participant 1** give your participant a name (e.g. “Customer”).
- Select the **Save** button.
- Open the **Select a participant** dropdown and select the participant you created.
- Select the **Apply** button.

Step 3C - add the Identity Verification step and configure it

- Select the **+** button to add a new step.
- Select **Verify Someone’s Identity** from the list**.**
- Select the **Configure** button.
- Select the **Select a participant** dropdown and select the participant you created.
- Select **Next**.
- Select the **Select identification verification** dropdown.
- Select **DocuSign ID Verification** from the list
- Select **Next**.
- Select the **Select variable to map to “Customer Name”** dropdown.
- Select **Collect Data with Web Forms**.

- Select **Customer Name**.
- Select the **Apply** button.

Step 3D - add the eSignature step and configure it

- Select the **+** button to add a new step.
- Select **Get Signatures** from the list.
- Select the **Configure** button.
- Select the **Choose a template** dropdown.
- Select **Account Opening Wire Transfer Approval** from the list.
- Select **Next**.
- Select the **Select a participant** dropdown and select the participant you created.
- Select the **Name** dropdown and select **Collect Data with Web Forms** and then **Customer Name**.
- Select the **Email** dropdown and select **Collect Data with Web Forms** and then **Customer Email**.
- Select **Next**.
- Select **Continue to map data fields (Optional)**.
- Select the Choose template role and select **Applicant.**
- For each of the following dropdowns - Name, Address1, Address2, City, State, and ZIP Code:
  select **Collect Data with Web Forms** and then matching property.
- Select the **Apply** button.

Step 3E - install the Google Extension App

- Go to https://apps-d.docusign.com/app-center
- Select **Log In** at the top right corner.
- If prompted - use the same Developer Account credentials you used before.
- Find the **Google Drive** app tile and select it.
- Select the **Install App** button.
- Select the **Continue** button..
- Once the app is installed select the **Connect Account** button.
- Log in with your Google credentials or select one from the list.
- Select the **Continue** button.
- Select the **Continue** button.
- Dismiss the pop-up dialog if one shows up.

Step 3F - add the Google Archive Step

- Select the **+** button to add a new step.
- Select **Share files in Google Drive** from the list.
- Select the **Configure** button.
- In the **Files to archive** dropdown select **Get Siganture.combinedDocumentBaset64**.

- Select **Next**.
- In the **Select drive** dropdown select **My Drive**.
- Select **Add Folder**.
- In the select folder dropdown pick the folder you want to use.
- Select **Next**.
- In the **What will you call it** section select **Add Variable**.
- Select **Collect Data with Web Forms** and then select **Customer Name**.
- Select the **Apply** button.

Step 3G - publish the new workflow

- Select the **Review & Publish** button from the top right corner.
- If everything is configured correctly you should see **This workflow is ready to publish** shown.
- Select the **Publish** button.
- In the pop-up dialog select **Authorize My Account**.
- In the pop-up dialog select **Allow Access**.
- Select **Publish**
- Select **Go to Workflows**.
- After a few seconds the new workflow status should change to **Published**.

Step 4 - use the VS Code Docusign Copilot Extension to obtain an access token

Step 4A - install the VS Code Docusign Copilot Extension

- Open VS Code.
- Click **Extensions** on the left menu
- In the **Search Extensions in Marketplace** search box type “Docusign.”’
- Find **Docusign Copilot Extension for Developers**.
- Click the blue **Install** button.
- If asked to do so - restart VS Code.

Step 4B - Use Copilot to obtain an access token

- Find the little Copilot button  at the bottom right corner of the screen and click it.
- Select **Github Copilot Chat** from the menu.
- Type `@docusign /getAccessToken` and hit enter.
- Click the blue **Sign In** button.
- If a VS Code popup dialog appears - click the **Open** button.
- A web browser will appear. If asked to login to Docusign, use the developer credentials you planed to use for this workshop.

- After you login you should be automatically redirected back to VS Code and a new popup dialog will appear.
- Click the **Open** button.
- Click the **Continue on the path of generating an access token** button in the Copilot chat window.
- In the question “Does your application need to perform automated tasks with Docusign without user intervention?” answer “**No**.”
- In the question “Can your application securely store sensitive information like client secrets or private keys?” answer “**Yes**.”
- When presented with three options, click the button “**Option 1 - generate an IK automatically using Copilot chat.**”
- Next, you need to provide a name for your IK (Integration Key). Type “Discover Workshop” and hit enter.
- Next, you need to enter a redirectURI for your IK. For this workshop we do not need any particular URI, so let’s type the example provided - http://localhost:3000/callback into the Copilot chat box and hit enter.
- Click the **Open Consent URI in browser** button.
- If a VS Code popup dialog appears - click the **Open** button.
- A web browser will appear, click **Allow Access** to continue.
- The browser will now redirect to http://localhost:3000/callback and will show an error message. Please ignore this and find the URL that the browser has opened. It will include a URL parameter called code with a long token. Copy this token.
- Back in the VS Code Copilot chat window paste the value you copied and hit enter.
- If everything worked correctly, you should see “Here is your Access Token” in the chat box, followed by the token (very long string).
- Copy paste the token and store it somewhere on your laptop, you’ll need it in the next step.

Step 5 - use Maestro API to retrieve the trigger URL

Step 5A - import the Postman collection for this workshop

- Open a browser window and navigate to www.postman.com
- Select the **Sign In** button at the top.
- Use your Postman credentials to log into Postman.
- Select **Workspaces** at the top left and then the **Create Workspace** button.
- Select Next
- In the Name textbox type “Docusign Discover” and select the **Create** button.
- Select the **Import** button at the top.
- Select **files** in the middle.
- Find the Postman collection JSON file provided for this workshop.
- Select **Open**.

- You should now see **Docusign Discover Workshop** and three GET requests on the left side.

Step 5B - call the Get Workflows Definitions endpoint using Postman

- Select the **Authorization** tab in the **Docusign Discover Workshop** Postman collection.
- Paste the token from Step 4 into the **Token** textbox and then select the **Save** button.
- Select the **Get Workflow Definitions** GET request from the list of API requests.
- Replace the `{{accountId}}` placeholder with the accountId of your developer account.
- Select the **Send** button to make the API call.
- You should get back JSON starting with a `count` value that tells you how many workflows were found.
- Find the workflow you created in Step 3 and copy the `id` property that you'll need in the next step.

Step 5C - call the Get Workflows Definition endpoint using Postman

- Select the **Get Workflow Definition** GET request from the list of API requests.
- Replace the `{{accountId}}` placeholder with the accountId of your developer account.
- Replace the `{{workflowId}}` placeholder with the id value of the workflow from the previous step.
- Select the **Send** button to make the API call.
- You should get back JSON information for the workflow definition for the specific workflow you created in step 3.
- The JSON should include a value called `triggerUrl`, this is the endpoint of the API call you'll use to trigger the workflow. Copy this value for use in the next step.

Step 5D - call the endpoint to trigger the workflow using Postman

- Select the **+** icon to add a new blank request.
- Paste the `triggerUl` from the previous step into the **Enter URL or paste text textbox**.
- Change the action dropdown on the left from GET to POST.
- Select **Headers**.
- Select and empty Key and type Authorization.
- In the Value textbox type Bearer and then paste the access token from Step 4.
- Select **Body**.
- Select the **raw** radio button.
- Paste the following JSON into the textbox - `{"instanceName": "Discover test run 1","participants": {},"payload": {},"metadata": {}}`
- Select the **Send** button to make the API call.

- If everything was set up correctly, you should get back a small JSON that includes an instanceId and a `workflowInstanceUrl` property.
- Copy the `workflowInstanceUrl` property for use in the next step.

Step 6 - Use URL from prior step to run through the workflow all the way to the end

Step 6A - fill up the web form

- Open a new web browser to the address you copied at the end of the previous step.
- Select the **Start** button.
- Fill out all the details of the web form. Note: The Customer Name should match the name you'll later use for identity validation.
- Select the **Next** button.
- Select the **Submit** button.

Step 6B - complete the Identity Verification process

- You should now see **Let's verify your identity** on the top of the page.
- Select the **Next** button.
- On the next screen - select either Driver's license or Passport for verification.
- Select the **I Agree** button.
- Option 1 - use your phone to scan images of your ID.
- Option 2 - upload files showing the back and front of your ID.
- Complete the identity verification process (should take 1-2 minutes).
- You should see "Your identity was verified" and be redirected to the next step.

Step 6C - sign the envelope

- Select the **I agree to use electronic records and signature** checkbox.
- Select the **Continue** button.
- You should note that some of the data in the envelope was pre-filled from the form you filled.
- Fill the rest of the data on the form (Feel free to use random/mock information).
- Select the Sign button to sign the envelope.
- Adopt a signature.
- Select the **Finish** button at the top right corner.
- You should now be redirected to the final confirmation page.

Step 7 - Use Navigator API to get list of agreements

- Go back to your Postman collection window
- Select the List Agreements GET request from the list of API requests.
- Replace the `{{accountId}}` placeholder with the accountId of your developer account.
- Select the **Send** button to make the API call.
- You should get back a JSON object with Agreement data from the Navigator product that should show the latest **Wire Transfer Agreement** that you signed in Step 6C.