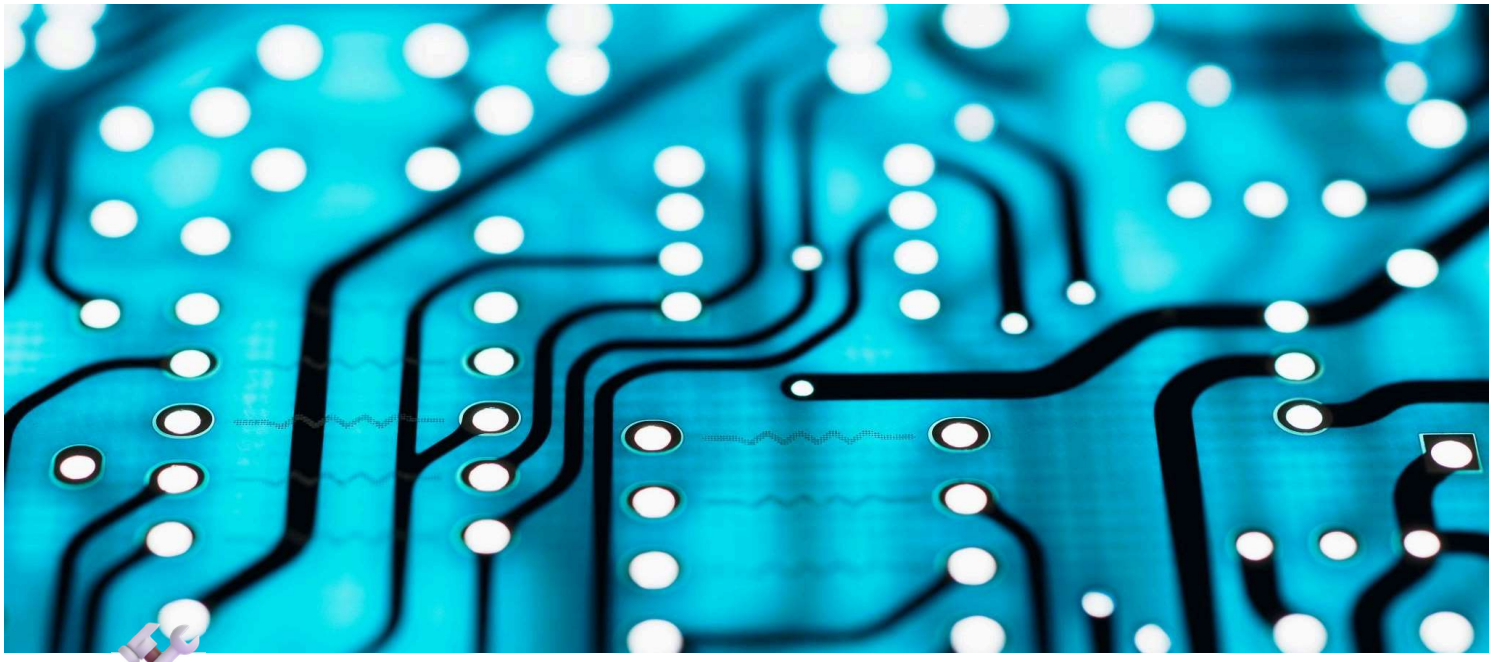


Hands On Lab #1: Getting Started

Overview

In this lab, we will use the SharePoint Embedded VS Code extension to setup a free trial SharePoint Embedded Container Type. Using that, we'll explore the SharePoint Embedded APIs in Postman and get a sample app running on our local machine.

Pre-Reqs

1. Laptop with the following applications installed:
    a. VS Code
    b. Postman client and a free Postman online account
    c. Git
    d. Node.js v20
    e. Azure Functions Core Tools: `npm i -g azure-functions-core-tools@4 --unsafe-perm true`
2. Global admin account on an M365 tenant with at least one SharePoint license

---

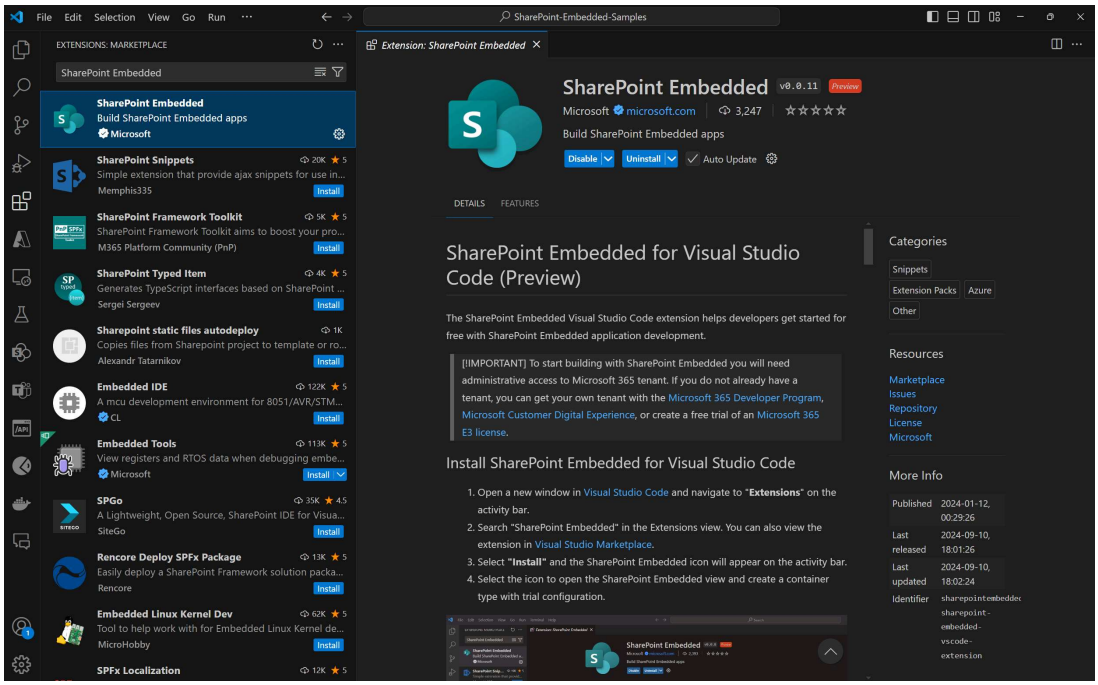
Steps

📌 Steps in VS Code are blue

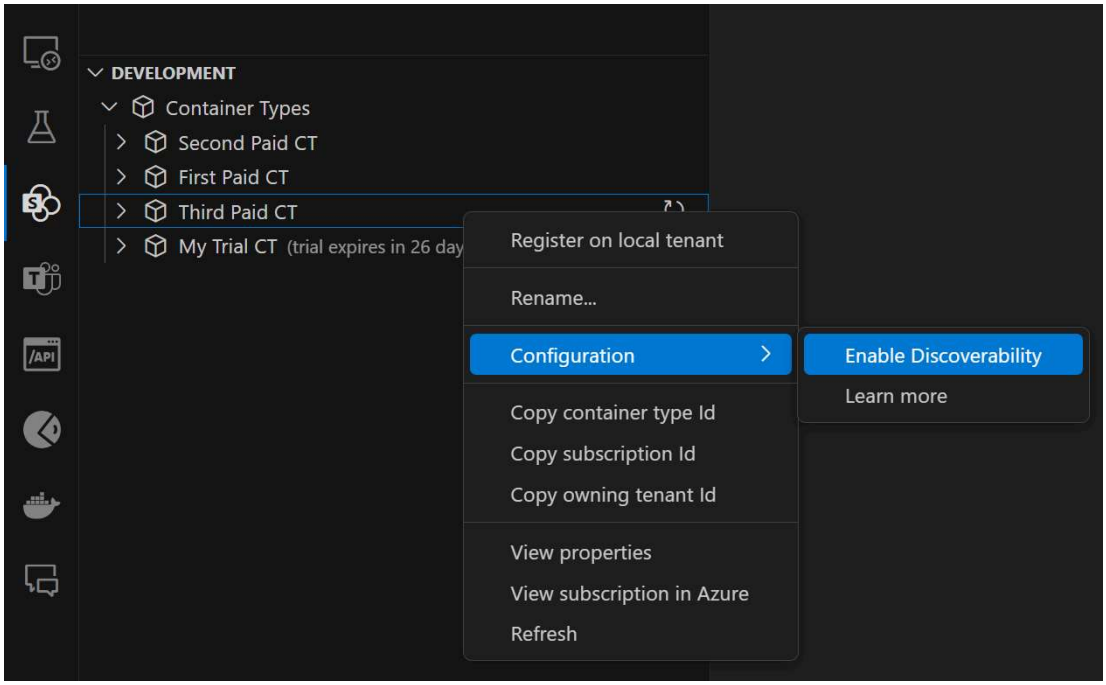
Steps in Postman client are orange

Steps in a Web browser are purple

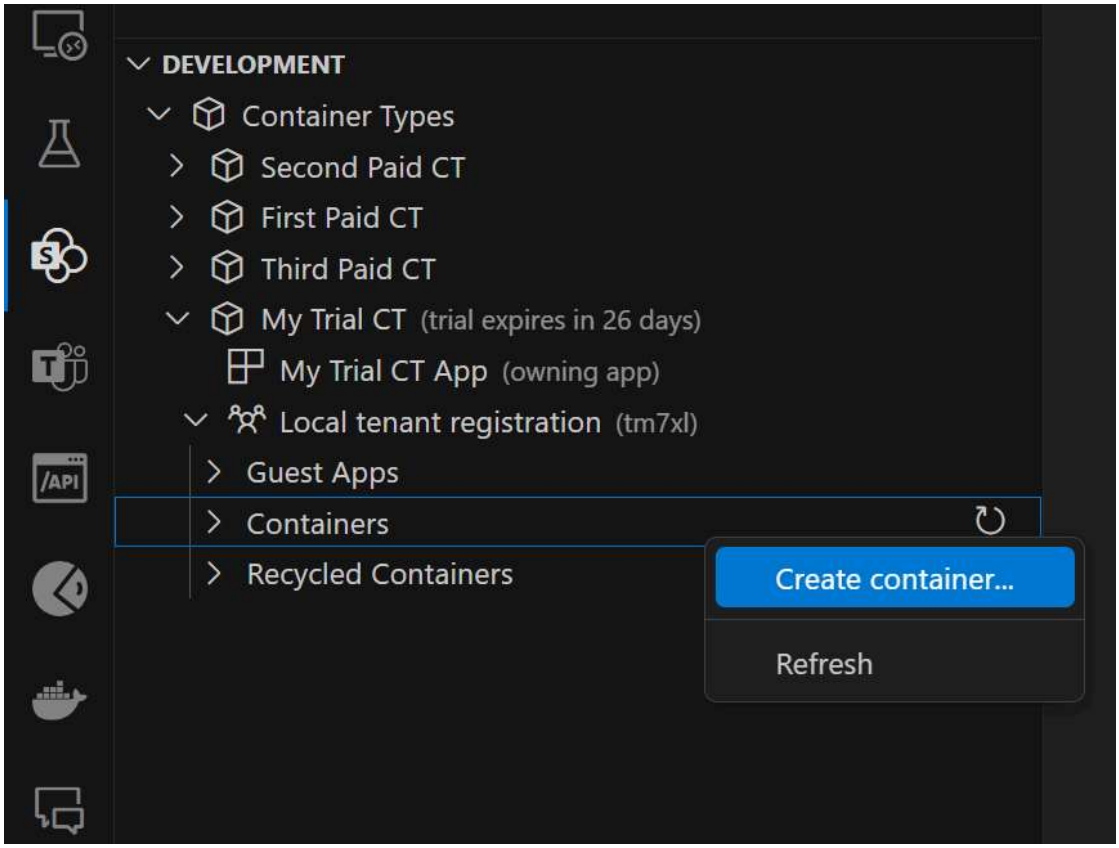
1. Open VS Code and install the SharePoint Embedded extension



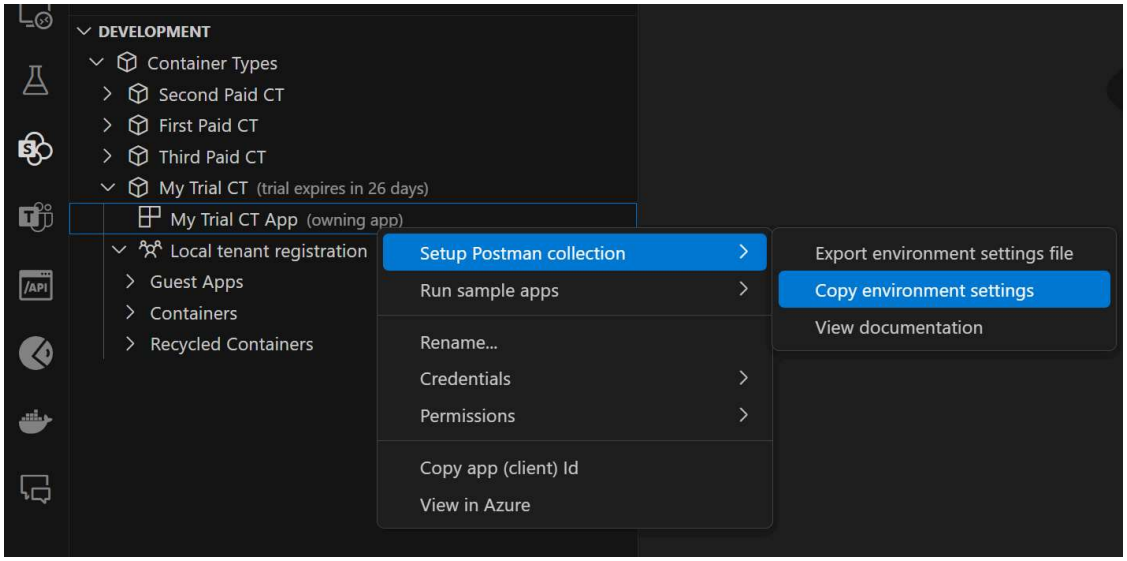
2. Sign in to the extension using your Global admin user account
3. Create a new Trial Container Type.
    a. Container Type display name: Call it whatever you like
    b. Entra App: Choose the option to create a new one, name it whatever you like
    c. Follow the prompt to register the container type on your local tenant, granting admin consent on your new app
4. Update the Container Type configuration to enable Discoverability.



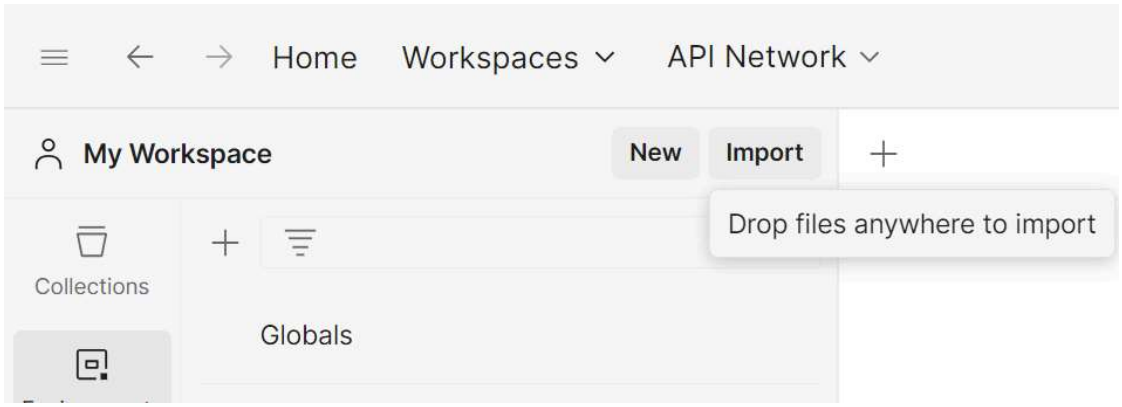
5. Create a Container.

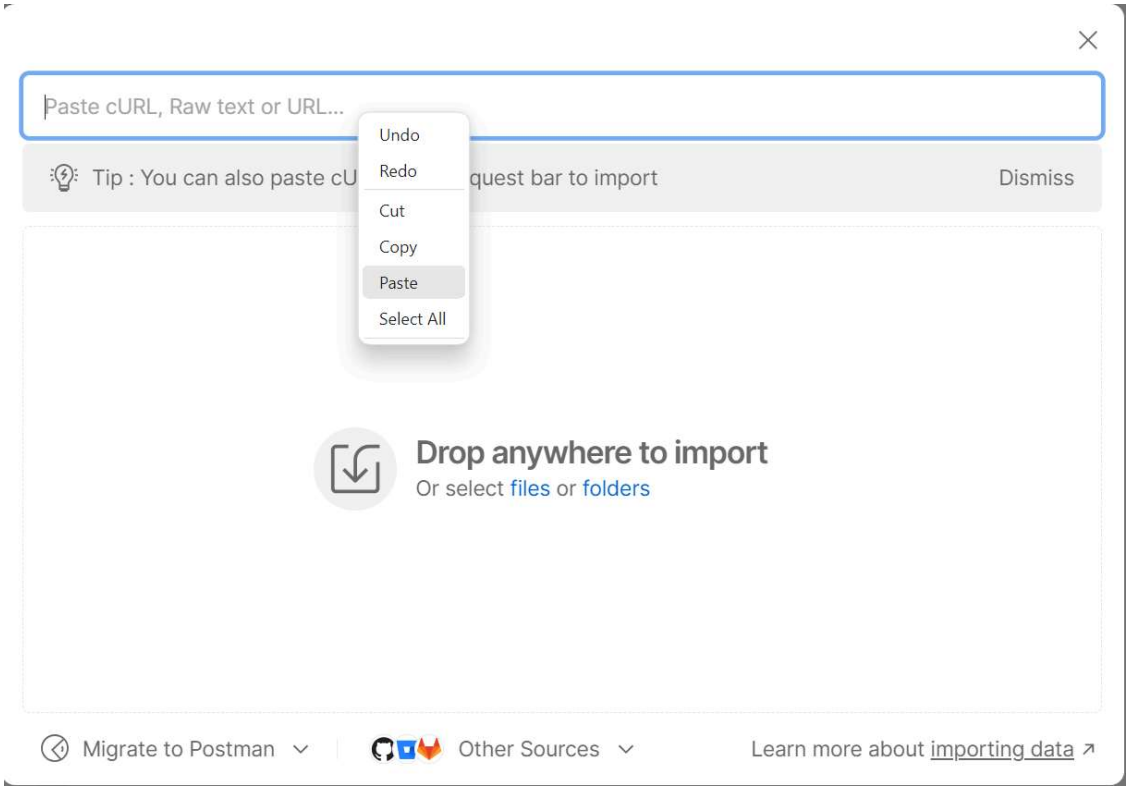


6. Export or copy a Postman environment from your owning Entra app. Accept the prompt to generate a new client secret to include in your environment, you'll need it.

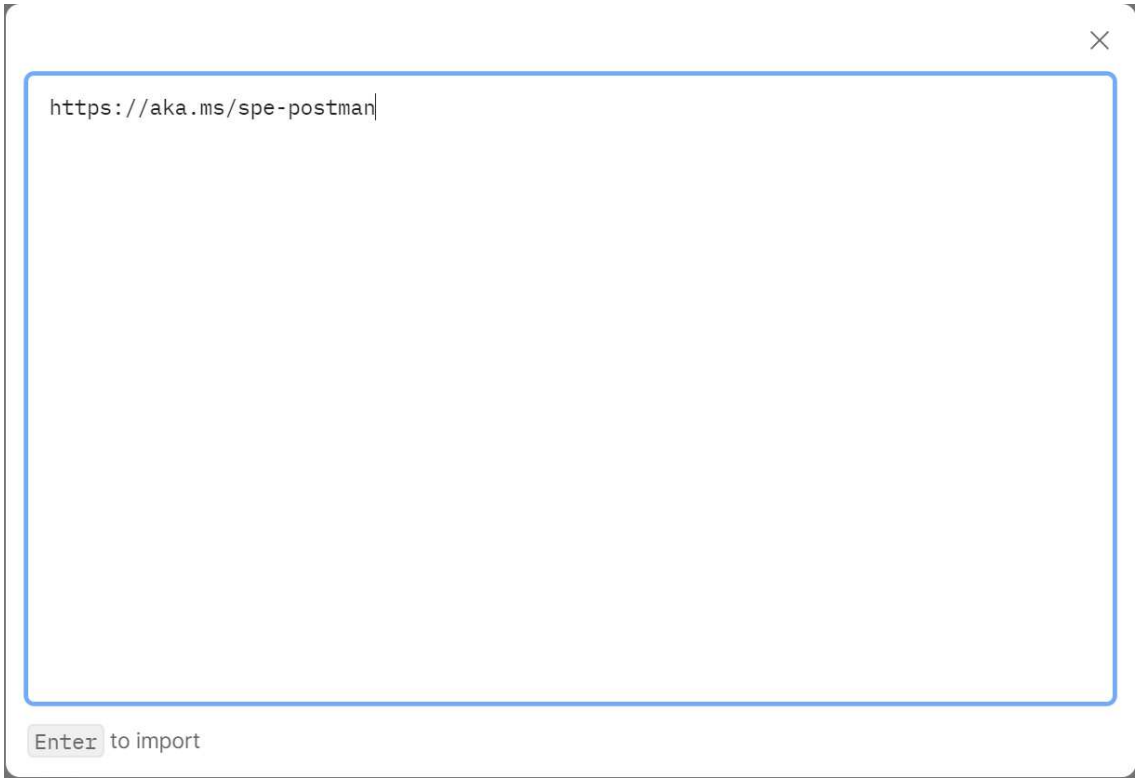


7. Open the Postman desktop client and import your environment

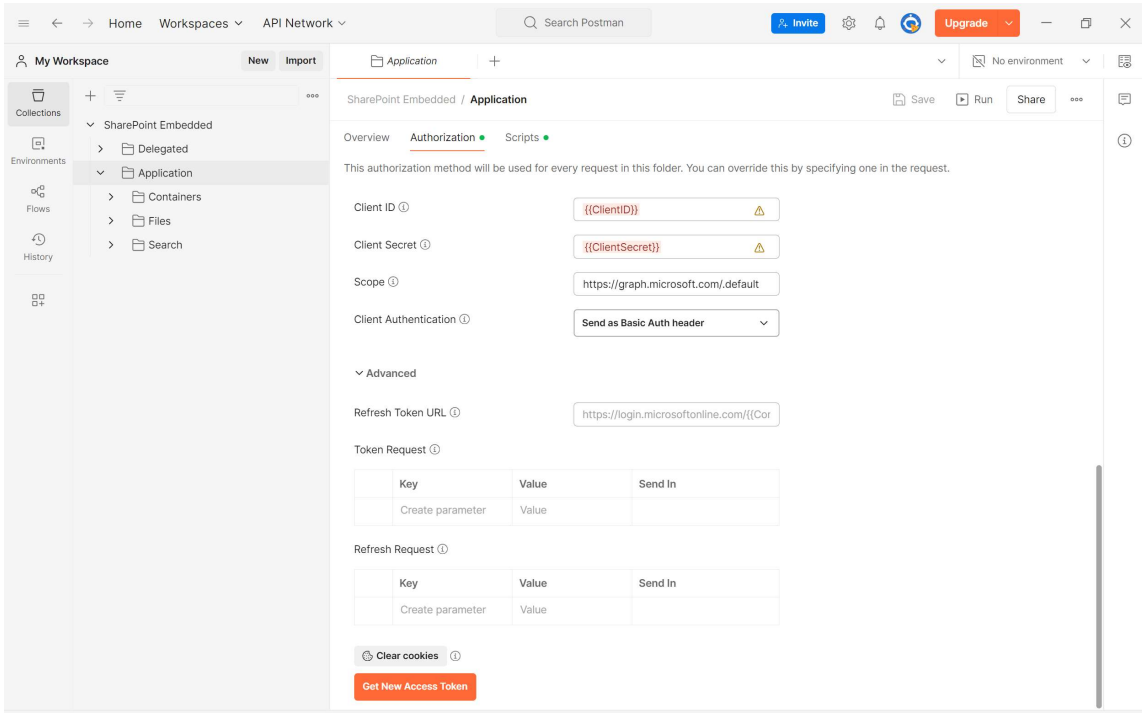




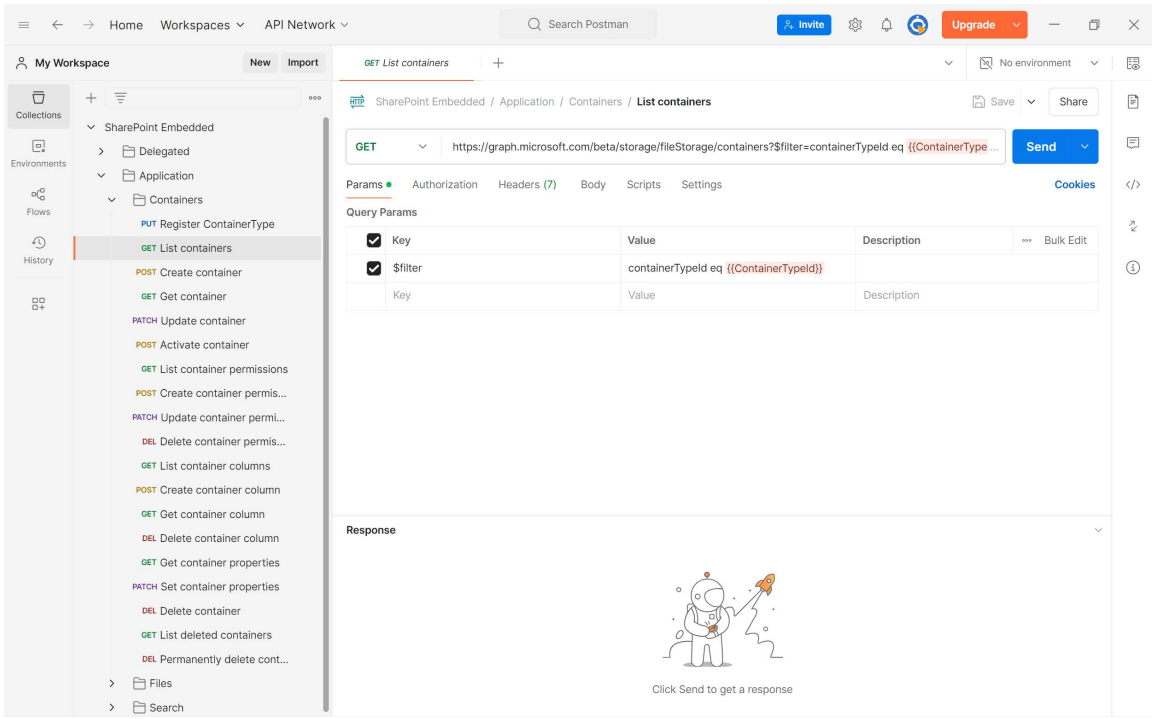
8. Import the SharePoint Embedded Postman collection from https://aka.ms/spe-postman



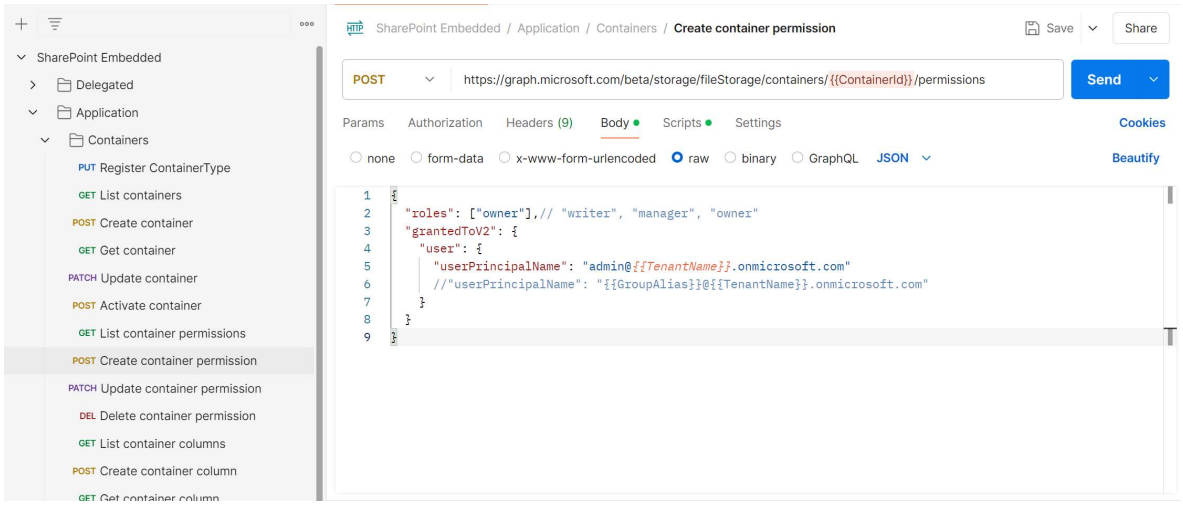
9. On the "Application" folder of the collection, get an access token from the 'Authentication' tab



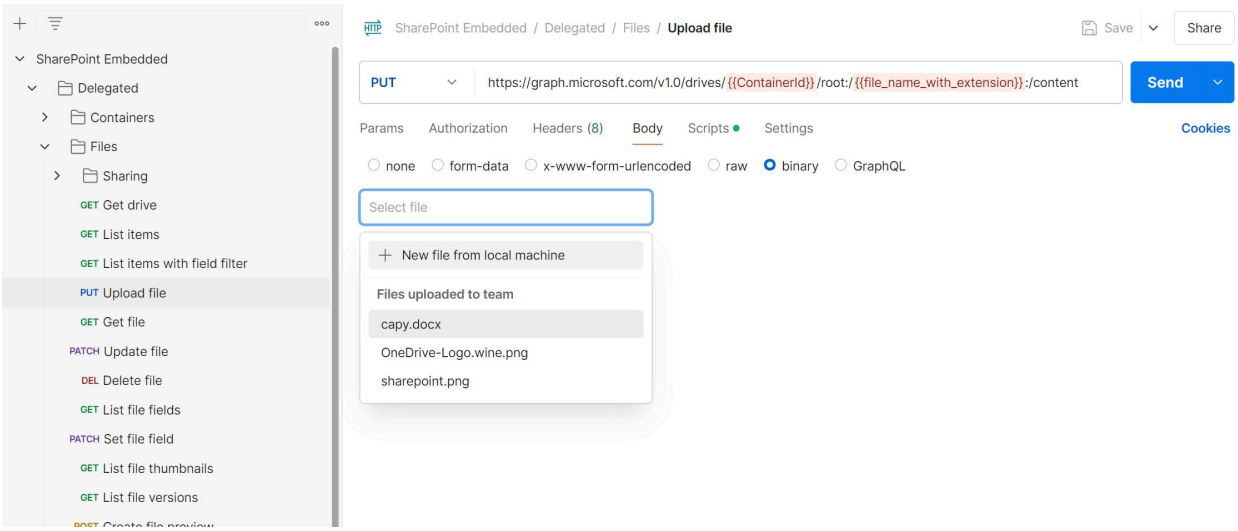
10. In the 'Containers' sub-folder, open the 'List containers' request and send it.



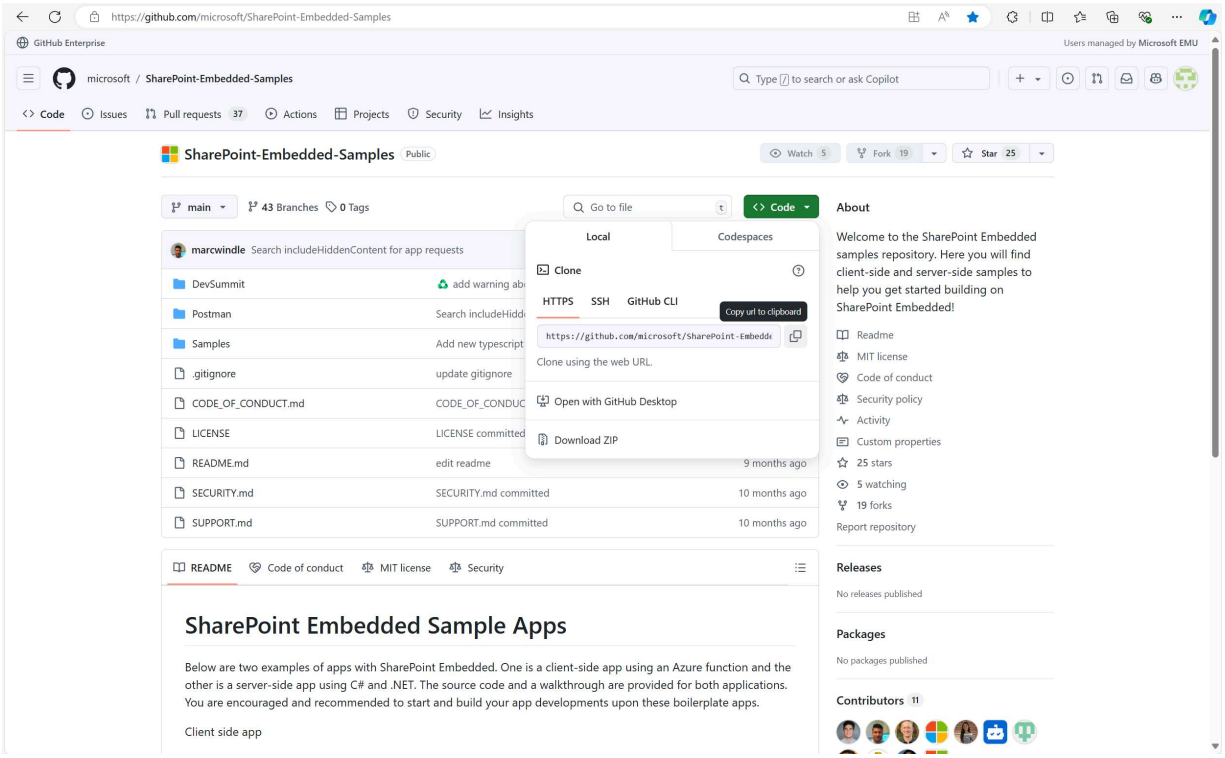
11. GET the container and look at its permissions
12. Add your admin user account as an owner on that container. If you are doing the Copilot chat lab later, do another request as needed to set your Copilot-licensed user account as an owner on that container too.



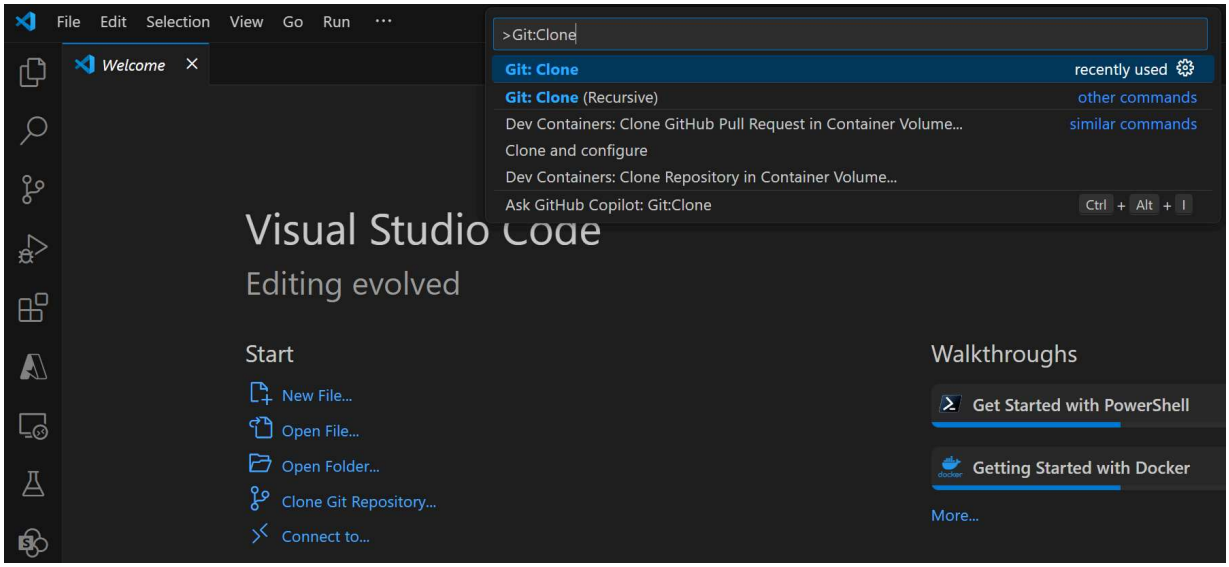
13. On the "Delegated" folder of the collection, get an access token from the 'Authentication' tab
14. Get the container as a Drive from the Files folder
15. Upload an Office document using the 'Upload file' request. You'll need to select a file from your local machine on the Body tab and update the *{{file_name_with_extension}}* URL variable with the name of your file.



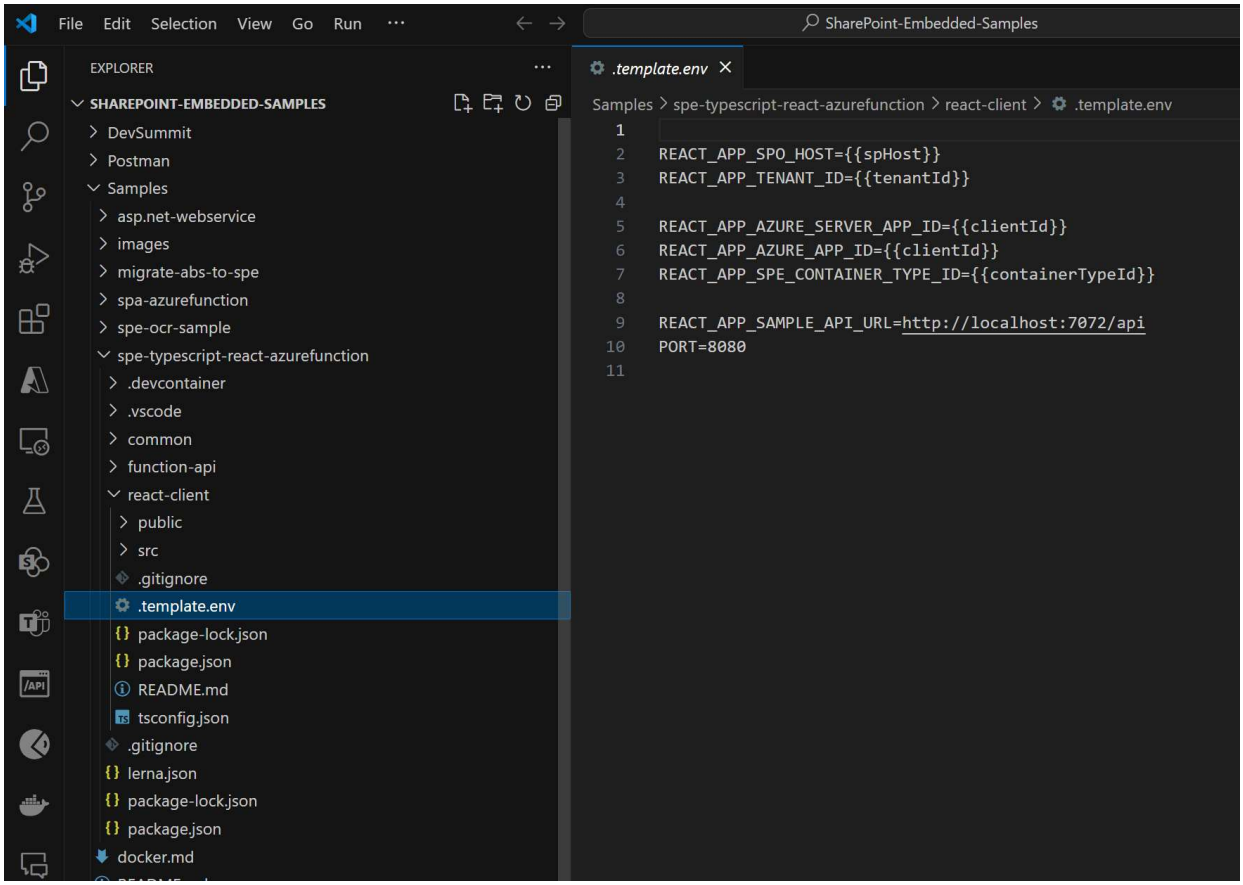
16. Visit the webUrl property link on the newly uploaded driveItem in a web browser where you can sign in with your global admin account
17. In a browser, visit https://aka.ms/spe-samples to view our sample app repository on GitHub and copy the 'Clone' URL to your clipboard.



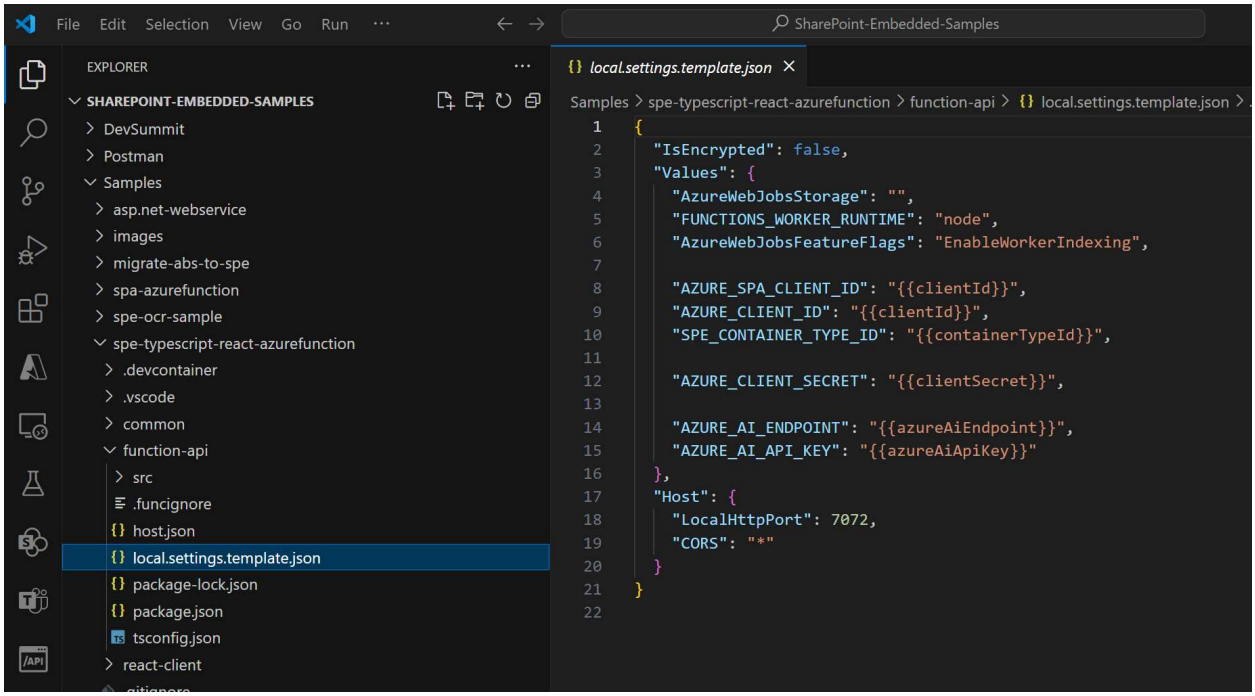
18. Back in VS Code, use the "Clone Git Repository" link from the Welcome view or run >Git:Clone in the top command input and provide the repository URL from your clipboard.



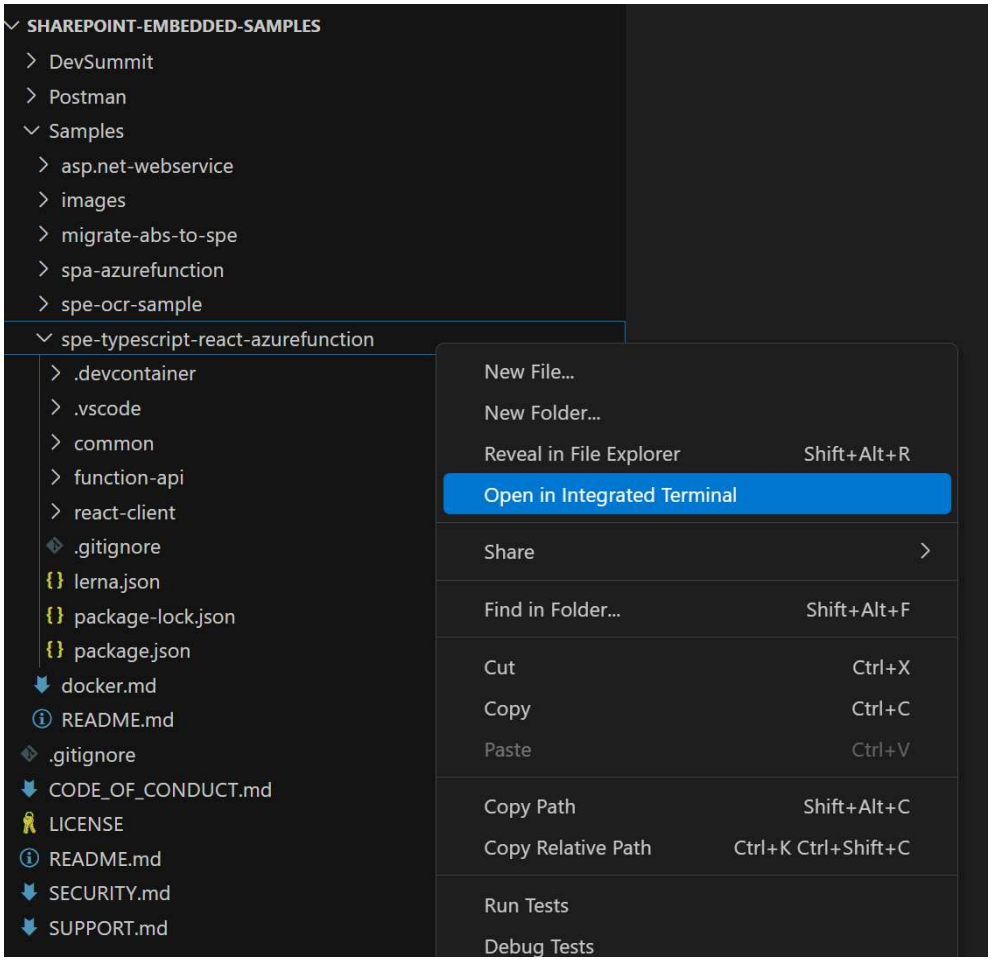
19. In the Samples/spe-typescript-react-azurefunction/react-client folder, copy the .template.env file to .env.local and set each of the environment variables in there according to your app. You can copy them from your Postman environment or use the SharePoint Embedded extension.

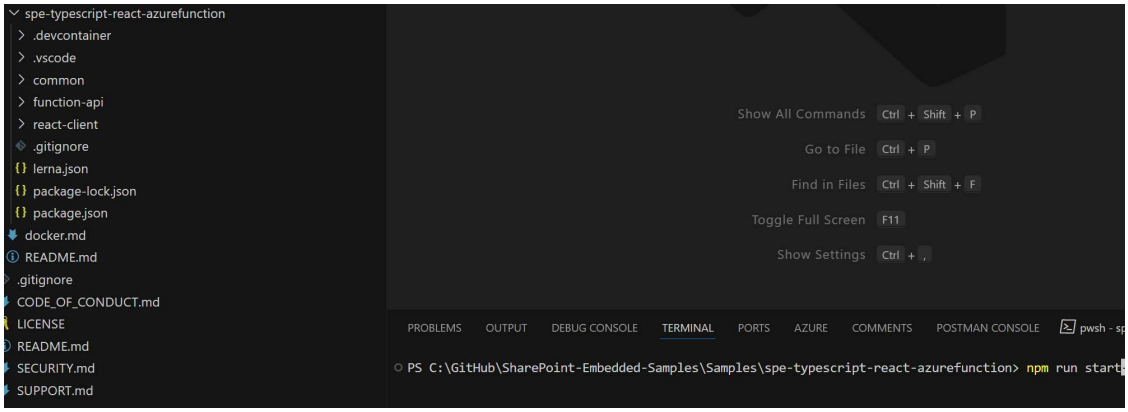


20. In the Samples/spe-typescript-react-azurefunction/function-api folder, copy the local.settings.template.json to local.settings.json and set each of the environment variables in there too. You can ignore the Azure AI related variables, we won't be using those in these labs.



21. Open an Integrated Terminal on the spe-typescript-react-azurefunction folder and run the 'npm run start' command. This will install all of the required libraries for the React front-end and Azure Functions backend and start both of them. Go grab a coffee, this step will take 5-10min, depending on your connection and machine.





22. In a browser, open http://localhost:8080 and sign into the sample using your global admin account (if you are doing the Copilot lab later, sign in with your Copilot-licensed user instead).
23. Click on your existing container and add some more files to it. Here are some things to try:
    a. Add Office documents, like Word, Excel, and PowerPoint files
    b. Select an Office document and try opening it in Web, Desktop, and Preview
    c. In Office for Web or Desktop, try adding comments, @-mention, sharing, or other rich collaboration experiences.
    d. Add and open a PDF and notice the full-fidelity document rendering ability of the preview client
    e. Add and open videos images, audio files, and/or 3D models and try out the media capabilities of the built-in preview client
24. If you're doing the Copilot lab: On the 'Containers' page, create a new container named 'fruits' and another one named 'vegetables'. Upload the content from the Fall24Events/hands-on-labs/content folder into the corresponding containers.