OpenText™ Thrust Studio Aviator

Basic Test Scenario

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

This basic test scenario has been created for software version OpenText™ Thrust Studio Aviator 24.2.2 CE.

It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Note that if you are using this test scenario with a later version of the OpenText™ Thrust Studio Aviator, the screenshots and usage might not always correspond.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111

Toll Free Canada/USA: 1-800-499-6544 | International: +800-4996-5440

Fax: +1-519-888-0677

Support: https://support.opentext.com

For more information, visit http://www.opentext.com

Copyright © 2024 Open Text. All Rights Reserved.

Trademarks owned by Open Text.

One or more patents may cover this product. For more information, please visit https://www.opentext.com/patents

Disclaimer

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the accuracy of this publication.

Last updated: 06/21/2024

Contents

Introduction **4**

1 Setting up your environment **5**

- 1.1 Prerequisites 5
- 1.2 Download and install VS Code 6
- 1.3 Add the OpenText™ Thrust Studio extension pack to VS Code 9
- 1.4 Install the OpenText™ Thrust Studio Aviator extension from VSIX 11

2 Set up the organization profiles **13**

- 2.1 Add the organization profile for your developer organization 13
- 2.2 Add a tenant to the developer organization profile 18
- 2.3 OPTIONAL: Add the organization profile for the OpenText™ Thrust Studio Aviator 20

3 Getting started with Thrust Studio Aviator **22**

- 3.1 Exploring the chat panel and selecting the Aviator organization 22
- 3.2 A developer conversation with Thrust Studio Aviator 27
- 3.3 Building an application with the assistance of Thrust Studio Aviator 46

About OpenText **50**

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

Introduction

This basic test scenario document is intended as an introduction to using the OpenText™ Thrust Studio Aviator digital assistant.

It will guide you through:

- Setting up the developer environment
- Configuring the organization (connection) profiles:
    - Organization profile to connect to your development organization
    - Organization profile to connect with the OpenText™ Thrust Studio Aviator
- Understanding how to ask questions and which responses to expect from the Generative AI enabled OpenText™ Thrust Studio Aviator
- Building an OpenText™ Thrust Services consuming application with the help of the OpenText™ Thrust Studio Aviator

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

1 Setting up your environment

The following software environment is used:

Software	Version
Operating System	Windows 11 Enterprise 22H2, 64-bit
VS Code	1.88.1
OpenText™ Cloud Developer Tools for VS Code	24.2.2 CE

Note

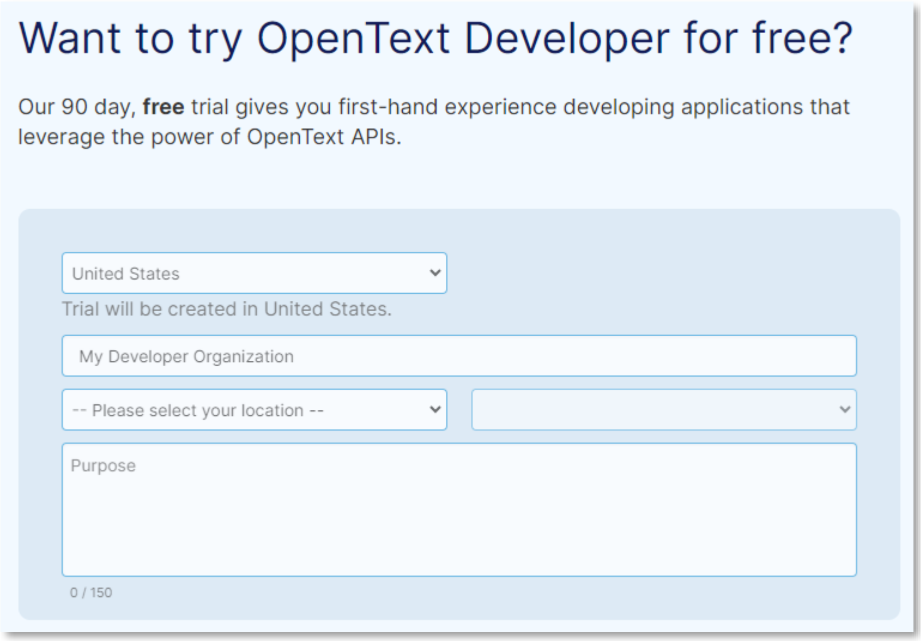
The procedure steps, images, and the other references used in this testing guide are based on the above environment. If you are using any other environment or software versions, refer to the software documentation for that version.

1.1 Prerequisites

Make sure you can sign in to the developer.opentext.com website and have an active trial or paid developer plan in the **United States** region.

To sign up for a trial, navigate to developer.opentext.com/plans to sign up for one.

IMPORTANT: Please, be certain to select the **United States** option when subscribing to the trial plan (see below screenshot).



Want to try OpenText Developer for free?

Our 90 day, **free** trial gives you first-hand experience developing applications that leverage the power of OpenText APIs.

United States

Trial will be created in United States.

For a detailed description on how to create a developer account and sign up for a developer trial, refer to: developer.opentext.com/services/trial.

1.2 Download and install VS Code

1. Go to https://code.visualstudio.com/download and download the Microsoft VS Code distribution that matches your system.

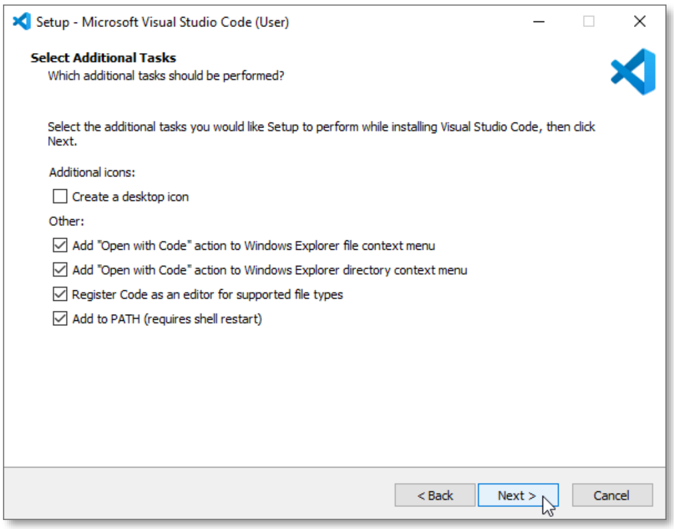
Note

To install VS Code on a 64-bit Windows 11 system, choose to download the x64 User Installer for Windows and install the latest version of VS Code.

Important

OpenText™ Thrust Studio is tested with Windows and Mac Operating Systems. You can use Linux Operating System but if you run into problems, OpenText might not be able to provide a solution.

2. Save and run the installer.
3. Select **I accept the agreement** and click **Next** to continue.
4. Select the installation destination location and click **Next**. You can use the suggested default location.
5. Keep the default setting for the selecting of the start menu folder and click **Next**.
6. Select all the additional tasks under **Other** and optionally select **Create a desktop icon**. Click **Next** to continue.



Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

☐ Create a desktop icon

Other:

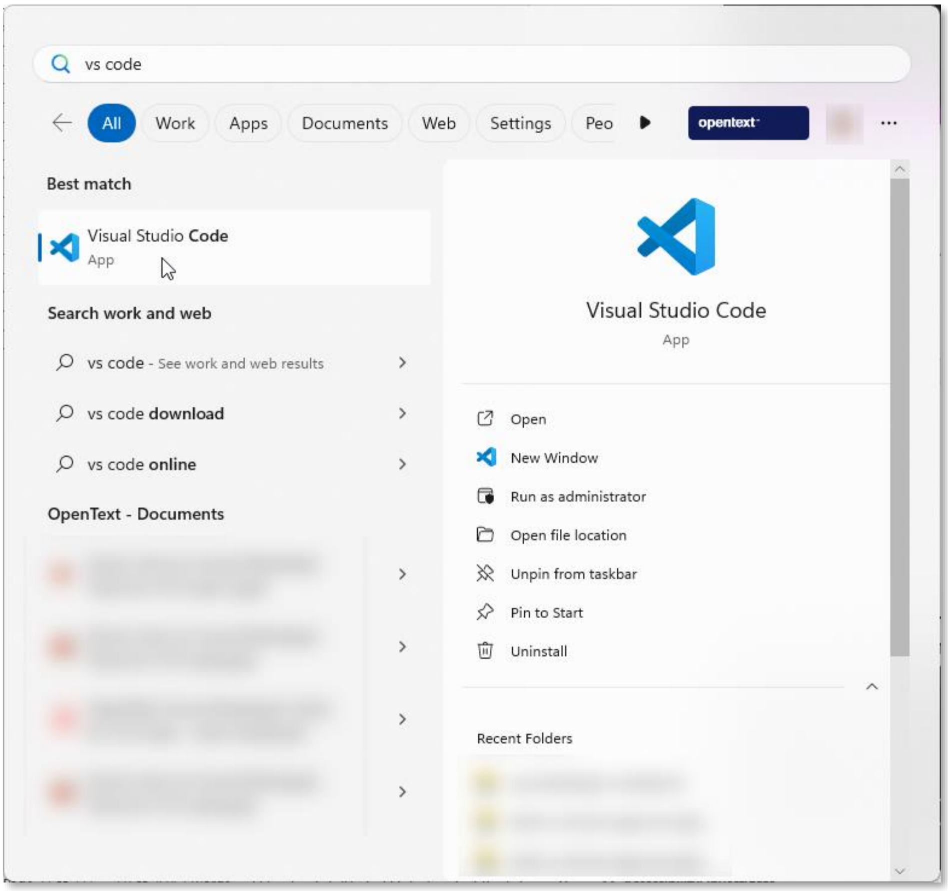
☑ Add "Open with Code" action to Windows Explorer file context menu
☑ Add "Open with Code" action to Windows Explorer directory context menu
☑ Register Code as an editor for supported file types
☑ Add to PATH (requires shell restart)

7. Verify your choices and click **Install** to start the VS Code installation.
8. After the installation is complete, you can click **Finish** to close the VS Code Setup Wizard.
9. After VS Code is installed, in the Windows **Start** menu, type `vs code` in the search box and select the **Visual Studio Code** application.

Tip

You can Pin to taskbar for easy access from the task bar.

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.



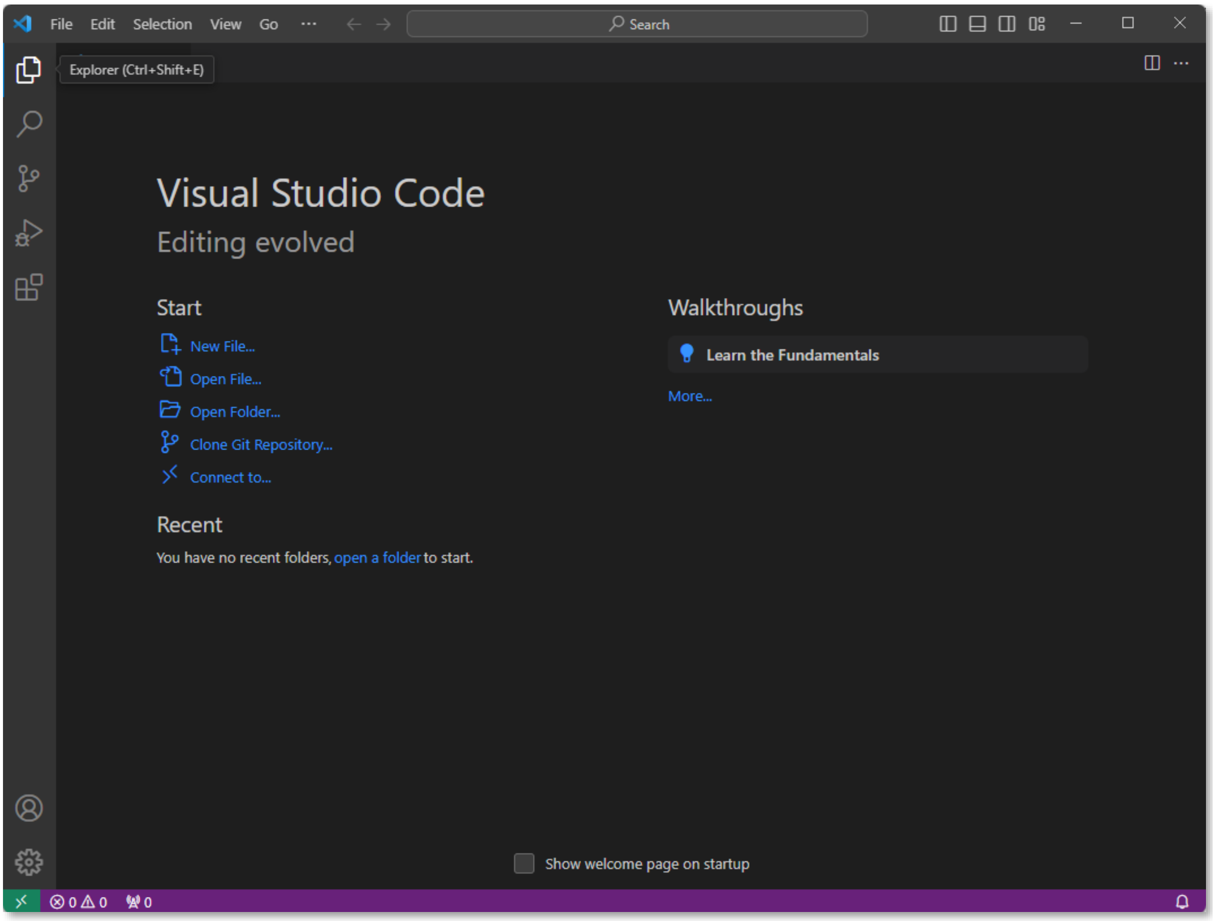
10. The **Get Started with VS Code** wizard is displayed when VS Code is launched for the first time.
11. Select a theme and click **Mark Done** to confirm your choice.

Note

For this testing guide the **Dark Modern** theme is selected. Take this into account when comparing your VS Code appearance with the VS Code screen shots throughout this document.

12. The standard VS Code welcome page is now displayed.

13. Make sure **Show welcome page on** startup is not selected so that the page does not appear when VS Code is launched every time.



You have no recent folders, open a folder to start.

☐ Show welcome page on startup

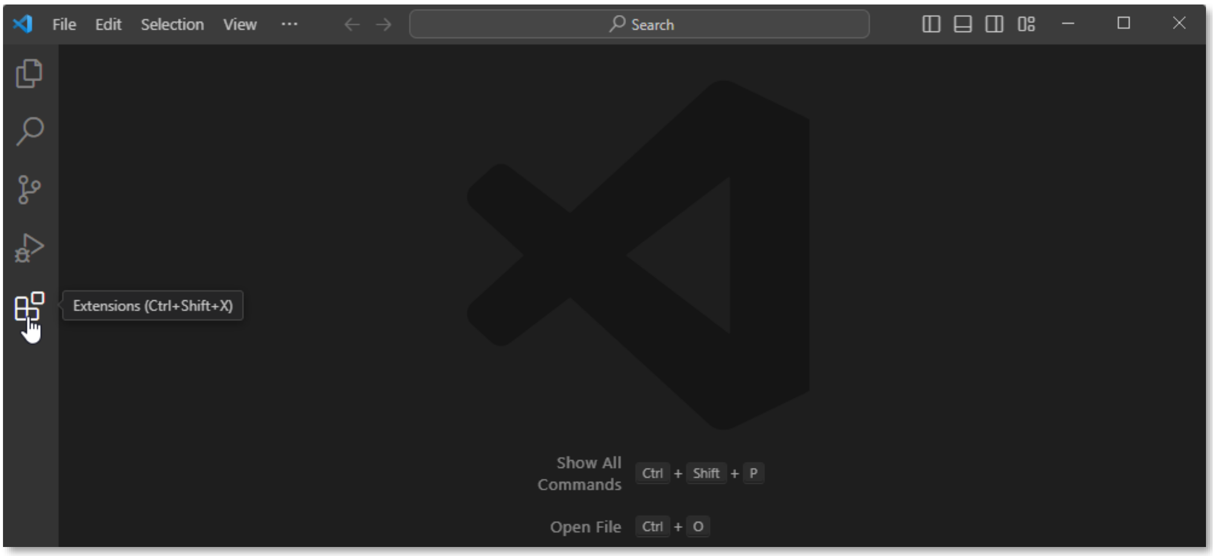
14. Close the welcome page.

Next step:

Add the OpenText™ Thrust Studio extension pack to your VS Code IDE.

1.3 Add the OpenText™ Thrust Studio extension pack to VS Code

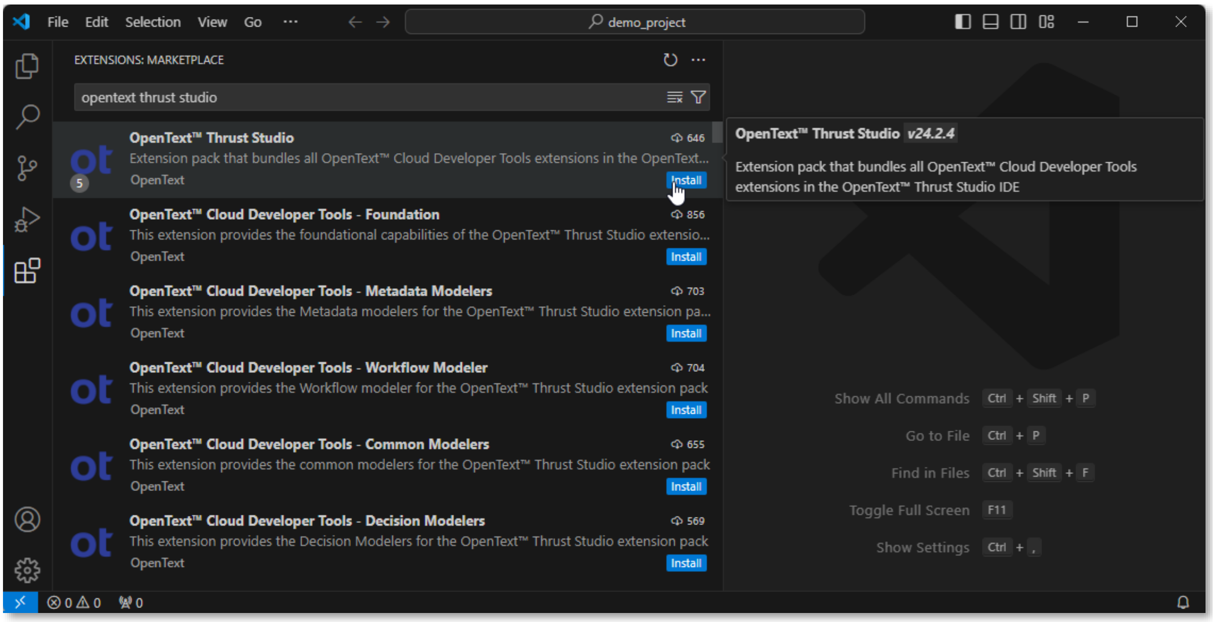
1. On the VS Code Activity Bar, click **Extensions**.



2. On the **Search** Extensions in Marketplace search bar, type `opentext thrust studio` and choose to install **OpenText™ Thrust Studio**.

Note

If prompted, click Reload to ensure that the installed VS Code extension pack is enabled.



opentext thrust studio

OpenText™ Thrust Studio 646
Extension pack that bundles all OpenText™ Cloud Developer Tools extensions in the OpenText...
OpenText

OpenText™ Cloud Developer Tools - Foundation 856
This extension provides the foundational capabilities of the OpenText™ Thrust Studio extensio...
OpenText

OpenText™ Cloud Developer Tools - Metadata Modelers 703
This extension provides the Metadata modelers for the OpenText™ Thrust Studio extension pa...
OpenText

OpenText™ Cloud Developer Tools - Workflow Modeler 704
This extension provides the Workflow modeler for the OpenText™ Thrust Studio extension pack
OpenText

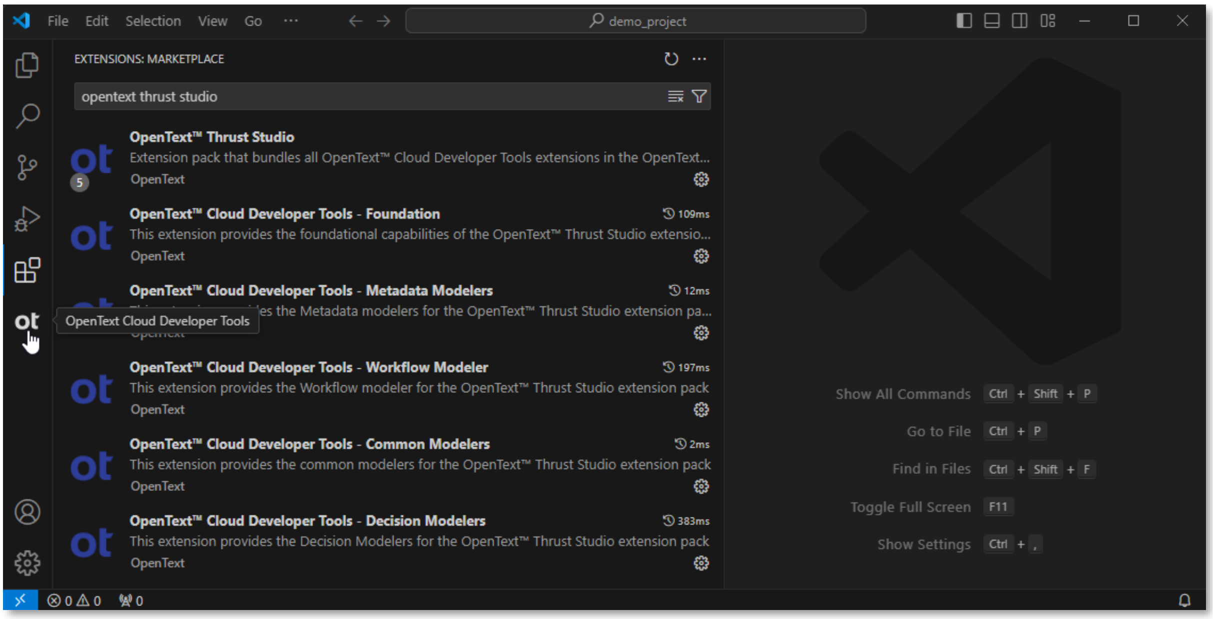
OpenText™ Cloud Developer Tools - Common Modelers 655
This extension provides the common modelers for the OpenText™ Thrust Studio extension pack
OpenText

OpenText™ Cloud Developer Tools - Decision Modelers 569
This extension provides the Decision Modelers for the OpenText™ Thrust Studio extension pack
OpenText

OpenText™ Thrust Studio *v24.2.4*

Extension pack that bundles all OpenText™ Cloud Developer Tools extensions in the OpenText™ Thrust Studio IDE

3. On the **Activity Bar**, click the **OpenText Cloud Developer Tools** button ot.



opentext thrust studio

OpenText™ Thrust Studio
Extension pack that bundles all OpenText™ Cloud Developer Tools extensions in the OpenText...
OpenText

OpenText™ Cloud Developer Tools - Foundation 109ms
This extension provides the foundational capabilities of the OpenText™ Thrust Studio extensio...
OpenText

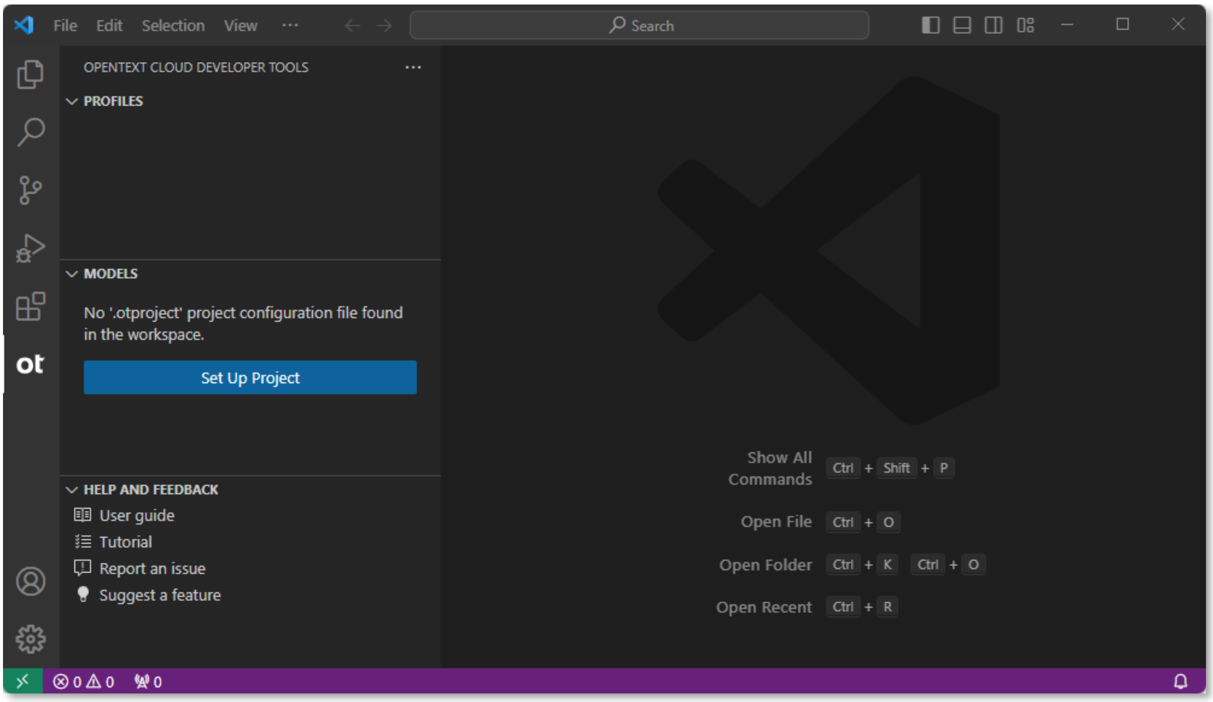
OpenText™ Cloud Developer Tools - Metadata Modelers 12ms
...es the Metadata modelers for the OpenText™ Thrust Studio extension pa...

OpenText™ Cloud Developer Tools - Workflow Modeler 197ms
This extension provides the Workflow modeler for the OpenText™ Thrust Studio extension pack
OpenText

OpenText™ Cloud Developer Tools - Common Modelers 2ms
This extension provides the common modelers for the OpenText™ Thrust Studio extension pack
OpenText

OpenText™ Cloud Developer Tools - Decision Modelers 383ms
This extension provides the Decision Modelers for the OpenText™ Thrust Studio extension pack
OpenText

4. This opens the **OpenText Cloud Developer Tools** view.



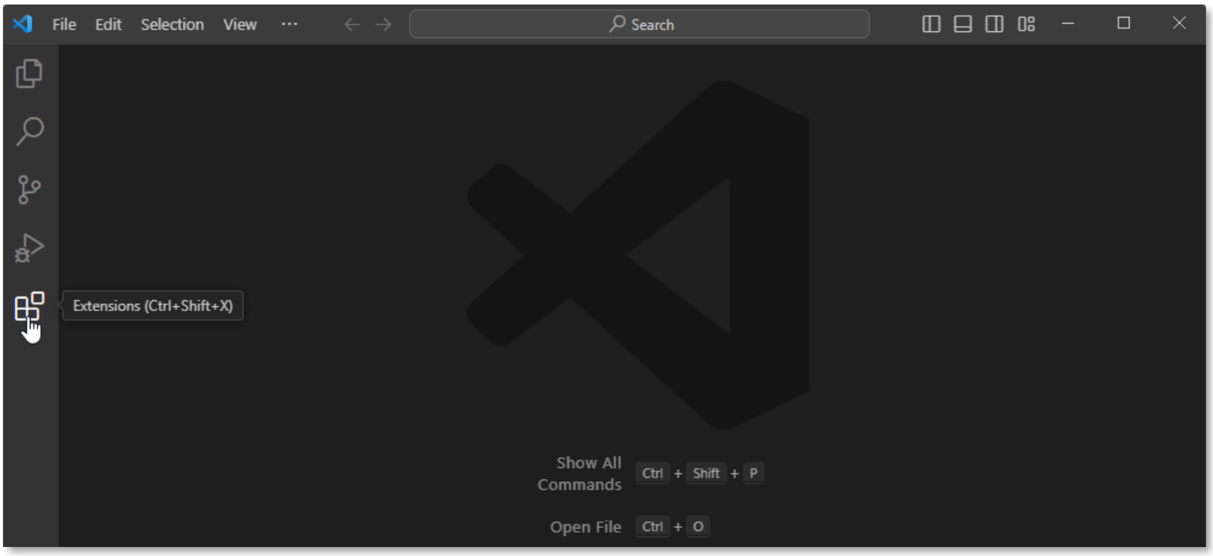
No '.otproject' project configuration file found in the workspace.

Next step:

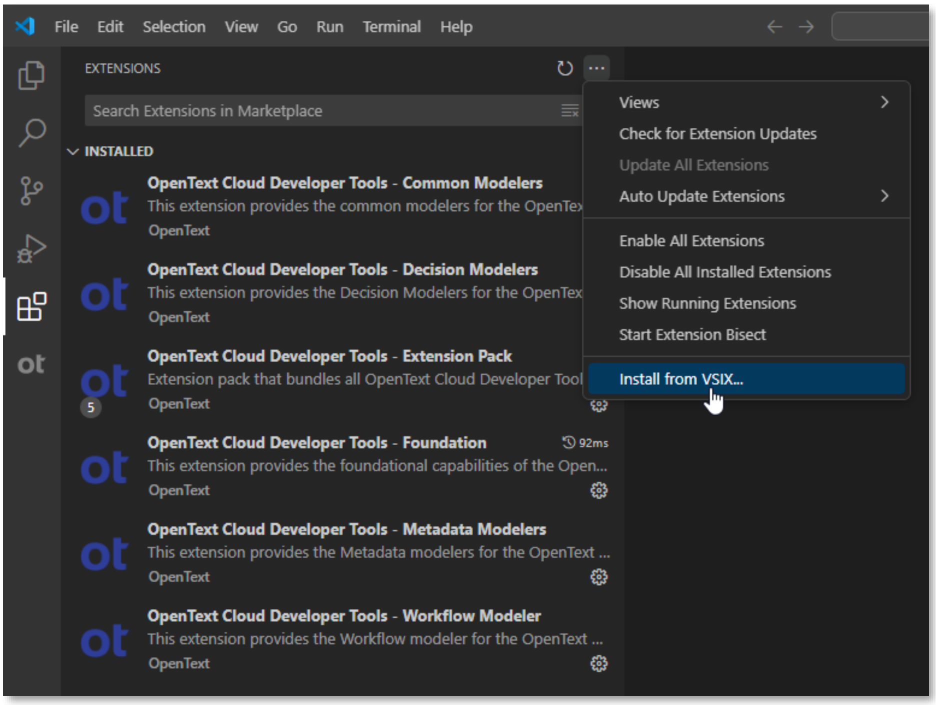
Install the OpenText™ Thrust Studio Aviator extension from VSIX.

1.4 Install the OpenText™ Thrust Studio Aviator extension from VSIX

1. Download the **ocp-vscode-aviator.vsix** file from this link.
2. In VS Code, from the Activity Bar, click **Extensions**.



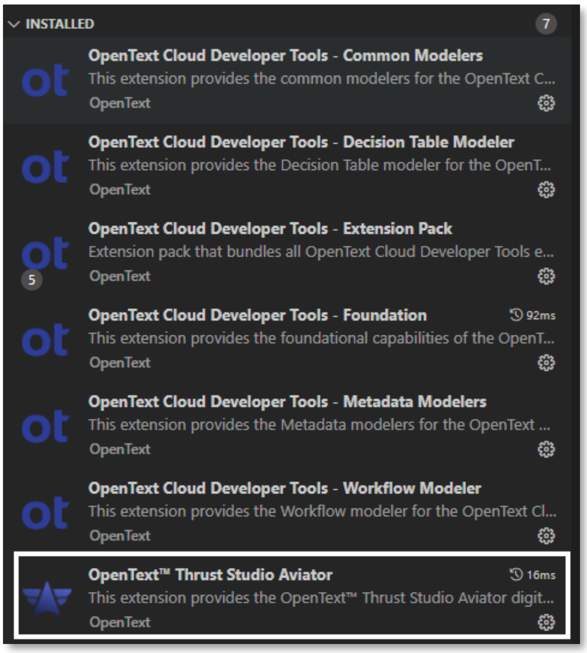
3. Click the **View and More Actions…** button ⋯ and select **Install from VSIX…**


File
Edit
Selection
View
Go
Run
Terminal
Help
EXTENSIONS
Search Extensions in Marketplace
INSTALLED
Views
Check for Extension Updates
Update All Extensions
Auto Update Extensions
Enable All Extensions
Disable All Installed Extensions
Show Running Extensions
Start Extension Bisect
Install from VSIX...
OpenText Cloud Developer Tools - Common Modelers
This extension provides the common modelers for the OpenTex
OpenText
OpenText Cloud Developer Tools - Decision Modelers
This extension provides the Decision Modelers for the OpenTex
OpenText
OpenText Cloud Developer Tools - Extension Pack
Extension pack that bundles all OpenText Cloud Developer Tool
OpenText
OpenText Cloud Developer Tools - Foundation
92ms
This extension provides the foundational capabilities of the Open...
OpenText
OpenText Cloud Developer Tools - Metadata Modelers
This extension provides the Metadata modelers for the OpenText ...
OpenText
OpenText Cloud Developer Tools - Workflow Modeler
This extension provides the Workflow modeler for the OpenText ...
OpenText

4. Select the previously downloaded **ocp-vscode-aviator.vsix** and click **Install**.

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

5. The **OpenText™ Thrust Studio Aviator** extension is now installed.



INSTALLED 7

OpenText Cloud Developer Tools - Common Modelers
This extension provides the common modelers for the OpenText C...
OpenText

OpenText Cloud Developer Tools - Decision Table Modeler
This extension provides the Decision Table modeler for the OpenT...
OpenText

OpenText Cloud Developer Tools - Extension Pack
Extension pack that bundles all OpenText Cloud Developer Tools e...
OpenText

OpenText Cloud Developer Tools - Foundation 92ms
This extension provides the foundational capabilities of the OpenT...
OpenText

OpenText Cloud Developer Tools - Metadata Modelers
This extension provides the Metadata modelers for the OpenText ...
OpenText

OpenText Cloud Developer Tools - Workflow Modeler
This extension provides the Workflow modeler for the OpenText Cl...
OpenText

OpenText™ Thrust Studio Aviator 16ms
This extension provides the OpenText™ Thrust Studio Aviator digit...
OpenText

Next chapter:

Set up the organization profiles.

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

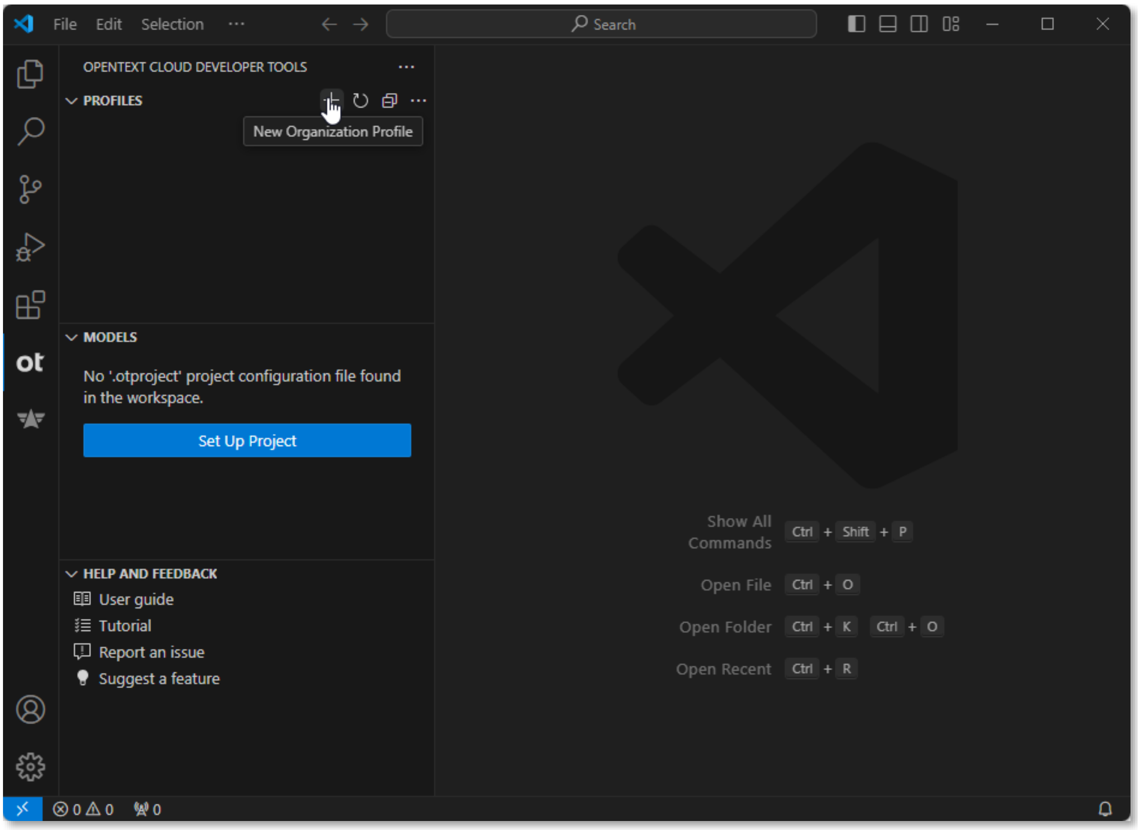
2 Set up the organization profiles

2.1 Add the organization profile for your developer organization

IMPORTANT: If you want to use a different developer organization than the one in the United States (US) region as per the Prerequisites (e.g., you have a developer organization on the NA Build & Test region or in Europe), you can choose to use that organization in this step, and then do the OPTIONAL: Add the organization profile for the OpenText™ Thrust Studio Aviator configuration step to add the US-based organization for the Thrust Studio Aviator digital assistant.

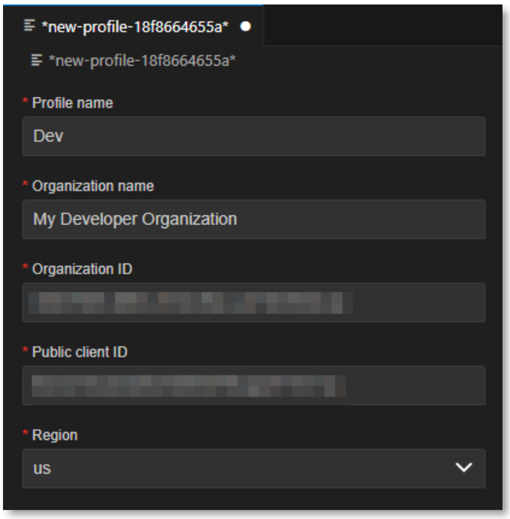
In case you only use a single (US) developer organization, there is no need to do the optional configuration step.

1. Open VS Code and on the Activity Bar, select the **OpenText Cloud Developer Tools** view.
2. In the **PROFILES** section, click the **New Organization Profile** button ⊞ to create a new organization profile.



No '.otproject' project configuration file found in the workspace.

3. In the organization profile screen, fill the property fields for the developer organization:



new-profile-18f8664655a

new-profile-18f8664655a

* Profile name
Dev

* Organization name
My Developer Organization

* Organization ID
[redacted]

* Public client ID
[redacted]

* Region
us

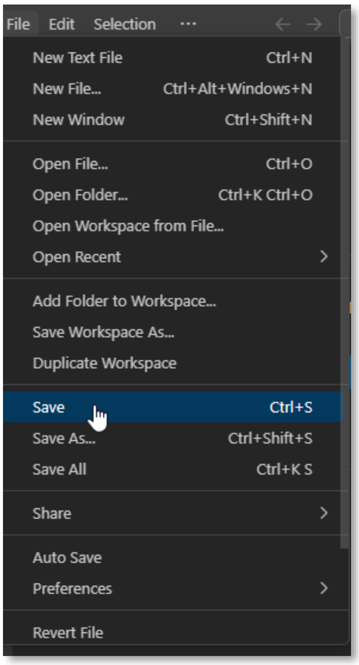
The following table describes the organization profile property fields:

Property	Description
Profile name	The profile name is used as a display name in the PROFILES section.
Organization name	The name of the organization. It is recommended to use the value from the organization overview page in Admin Center. For more information on accessing the organization overview page in Admin Center, see Admin Center User Interface.
Organization ID	The unique identifier for the organization. For more information about how to view your organization ID, see View your organization ID using Admin Center.
Public client ID	The unique identifier of the public OAuth service client for the organization. For more information about how to view your public client ID, see View and manage OAuth service clients.
Region	The region where the organization is made available. By default, the value is na-1-dev but except if you are using a different developer organization you should select **us**. If you want to confirm the region of your developer organization, you can do this by opening Admin Center for the organization and using the URL (region is part of the domain name).

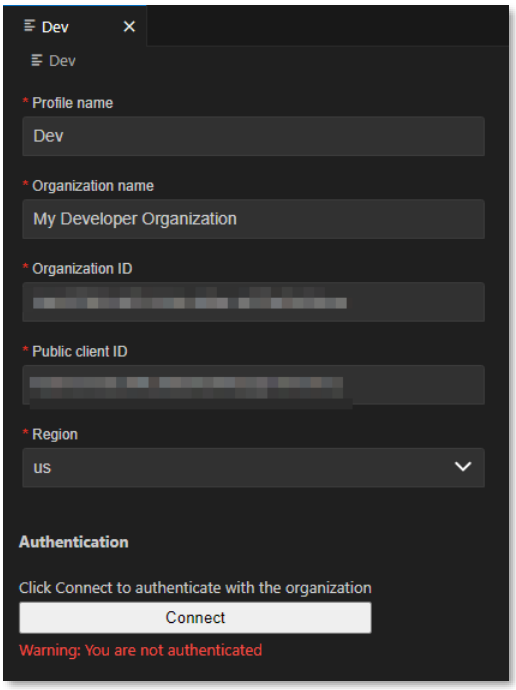
4. Select **File > Save** to save the organization profile.

Note

OpenText Cloud Developer Tools configuration artifacts such as setting up organization connection, project set up, and different model configurations use the standard VS Code file saving functionality. To save your changes to any configuration artifact, press **Ctrl+S** (for Windows systems) or select **File > Save**.



5. After saving, the organization profile is listed as the default profile in the **PROFILES** section and the tab above the organization profile is renamed to the profile name. On the organization profile screen an **Authentication** section with the **Connect** button appears.



* Profile name: Dev

* Organization name: My Developer Organization

* Organization ID: [redacted]

* Public client ID: [redacted]

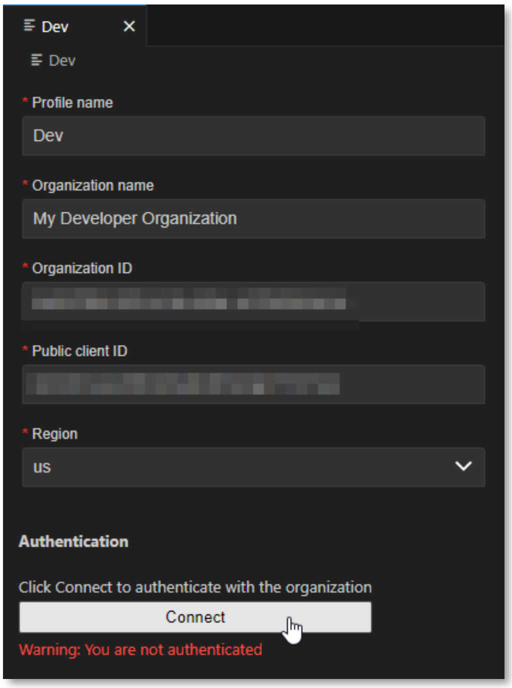
* Region: us

Authentication

Click Connect to authenticate with the organization

Warning: You are not authenticated

6. Click **Connect** to test the newly configured connection to your developer organization.



* Profile name: Dev

* Organization name: My Developer Organization

* Organization ID: [redacted]

* Public client ID: [redacted]

* Region: us

Authentication

Click Connect to authenticate with the organization

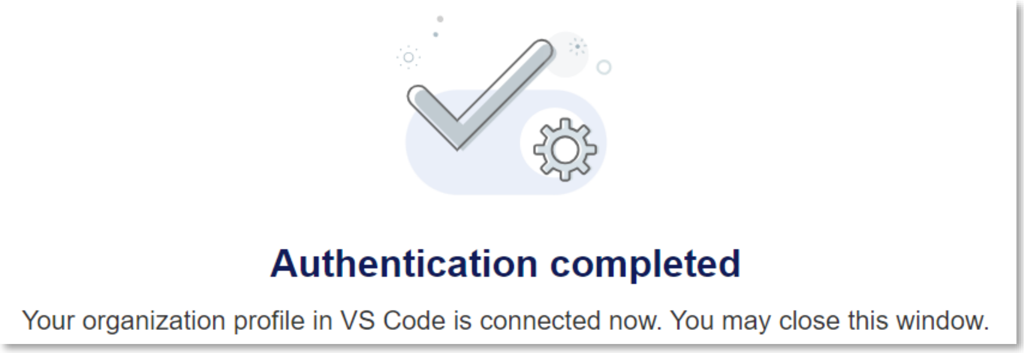
Warning: You are not authenticated

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

7. On the sign in page, authenticate with your developer.opentext.com username and password.

Note

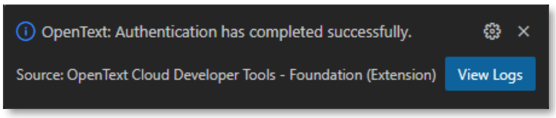
If you are already signed in, the browser will immediately show the **Authentication completed** page.



Authentication completed

Your organization profile in VS Code is connected now. You may close this window.

8. After successful authentication, a confirmation message is displayed in VS Code.



OpenText: Authentication has completed successfully.

Source: OpenText Cloud Developer Tools - Foundation (Extension)

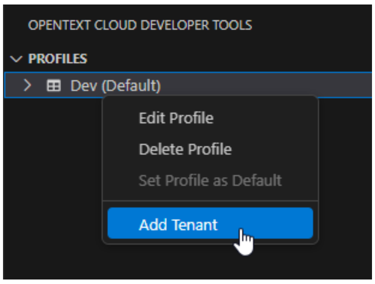
9. Close the organization profile screen.

Next step:

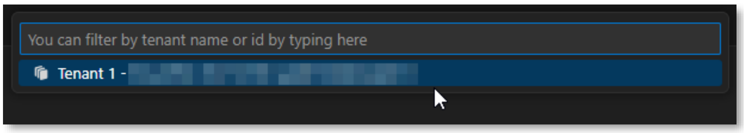
Add a tenant to the developer organization profile.

2.2 Add a tenant to the developer organization profile

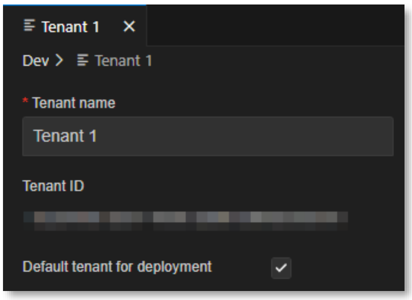
1. In the **PROFILES** section, right-click the profile and select **Add Tenant**.



2. From the drop-down in the top center of VS Code, select the developer tenant you want to use (typically "Tenant 1" if you are on a trial plan).



Tenant 1 - [redacted]



* Tenant name: Tenant 1

Tenant ID: [redacted]

Default tenant for deployment ☑

The following table describes the tenant property fields:

Property	Description
Tenant name	The name of the tenant. It is recommended to use the value from the tenant page in Admin Center. For more information on accessing the tenant page in Admin Center, see Retrieve your tenant ID or reset tenant password using Admin Center.
Tenant ID	The unique identifier for the tenant. For more information about how to view your tenant ID, see Retrieve your tenant ID or reset tenant password using Admin Center.
Default tenant for deployment	Whether to use this tenant by default when deploying to the corresponding organization.

Note

The first tenant added to the organization profile is made the default tenant for deployment.

Next step:

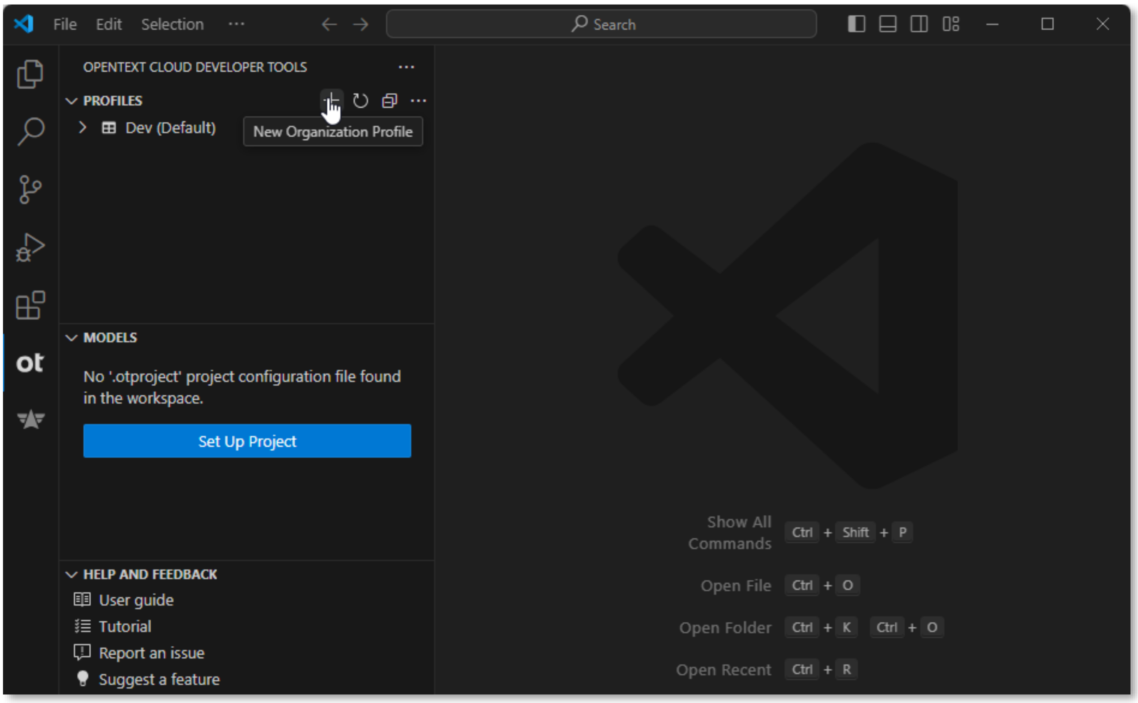
OPTIONAL: Add the organization profile for the OpenText™ Thrust Studio Aviator.

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

2.3 OPTIONAL: Add the organization profile for the OpenText™ Thrust Studio Aviator

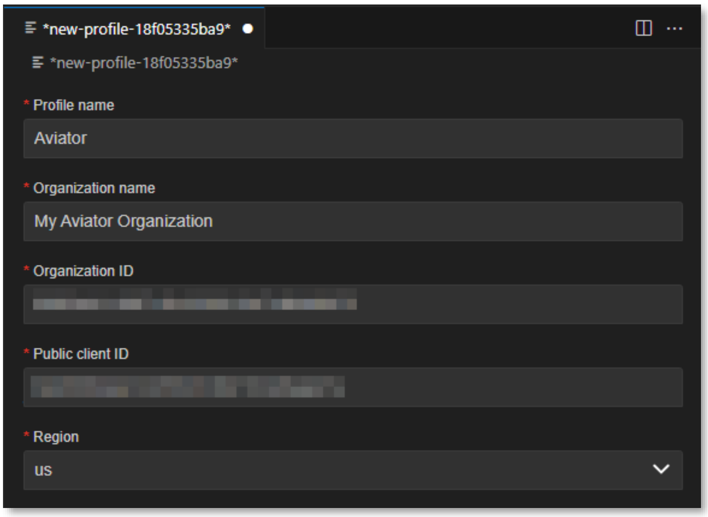
This optional step is only required if you want to work with a different developer organization than the **United States** based organization you subscribed to (see Prerequisites), and you have set this non-US organization as the developer organization when performing the Add the organization profile for your developer organization step.

1. From the **OpenText Cloud Developer Tools** view, add another organization profile.



No '.otproject' project configuration file found in the workspace.

2. In the organization profile screen, fill the property fields for the US-based organization (in the same way as described in Add the organization profile for your developer organization) giving it a name indicating its use for the OpenText™ Thrust Studio Aviator (e.g., naming the profile "Aviator"):



* Profile name: Aviator

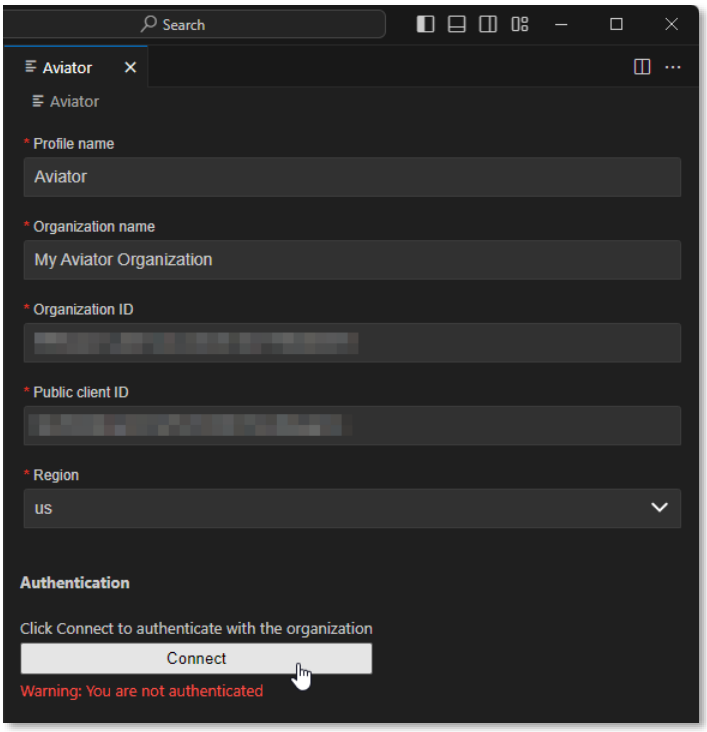
* Organization name: My Aviator Organization

* Organization ID: [redacted]

* Public client ID: [redacted]

* Region: us

3. Select **File > Save** to save the organization profile and click **Connect** to test the newly configured connection to your Aviator organization.



Aviator

* Profile name
Aviator

* Organization name
My Aviator Organization

* Organization ID
[redacted]

* Public client ID
[redacted]

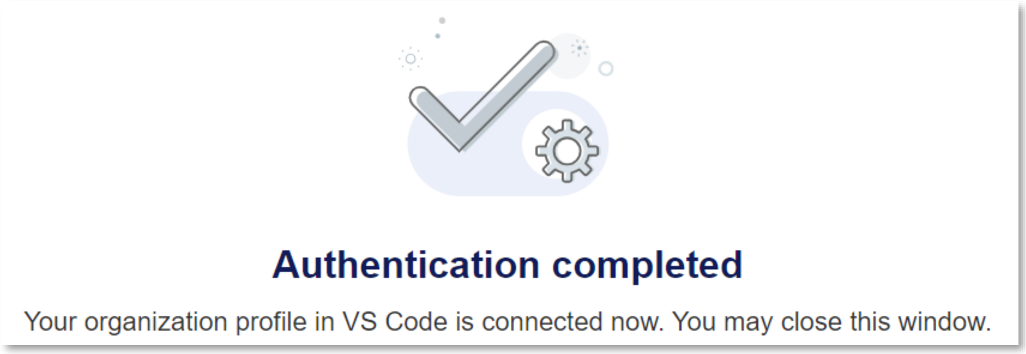
* Region
us

Authentication

Click Connect to authenticate with the organization

Warning: You are not authenticated

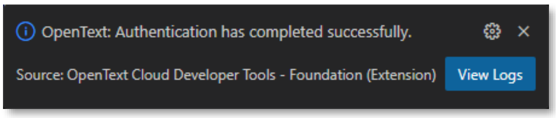
4. On the sign in page, authenticate with your developer.opentext.com username and password.



Authentication completed

Your organization profile in VS Code is connected now. You may close this window.

5. After successful authentication, a confirmation message is displayed in VS Code.



OpenText: Authentication has completed successfully.

Source: OpenText Cloud Developer Tools - Foundation (Extension)

6. Close the organization profile screen.

Next chapter:

Getting started with Thrust Studio Aviator.

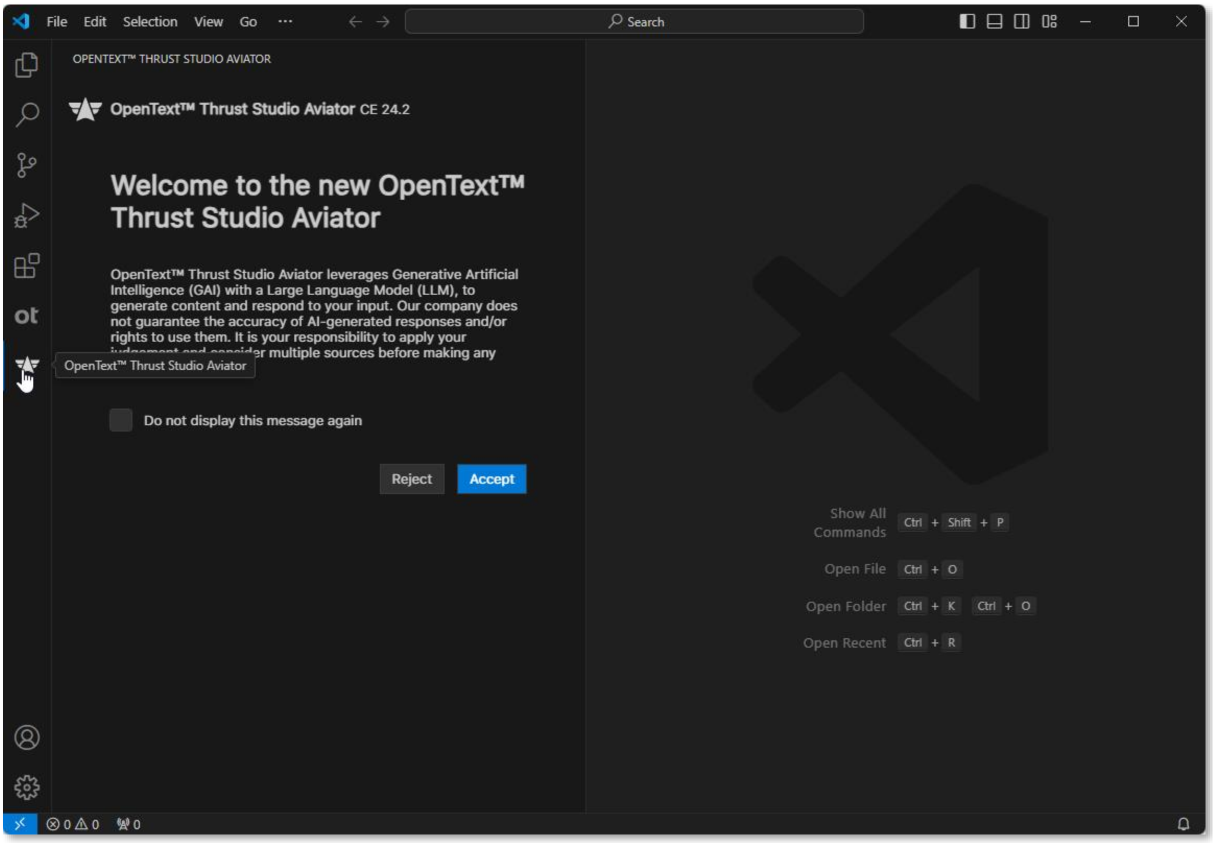
Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

3 Getting started with Thrust Studio Aviator

This chapter will introduce you to using the OpenText™ Thrust Studio Aviator, by exploring the chat panel and asking a set of questions you would typically ask in context of working with the OpenText™ Thrust Services.

3.1 Exploring the chat panel and selecting the Aviator organization

1. Open VS Code and on the Activity Bar, select the **OpenText™ Thrust Studio Aviator** view.



File
Edit
Selection
View
Go
Search
OPENTEXT™ THRUST STUDIO AVIATOR
OpenText™ Thrust Studio Aviator CE 24.2
ot
OpenText™ Thrust Studio Aviator
Show All Commands Ctrl + Shift + P
Open File Ctrl + O
Open Folder Ctrl + K Ctrl + O
Open Recent Ctrl + R
⊗ 0 ⚠ 0
0

Welcome to the new OpenText™ Thrust Studio Aviator

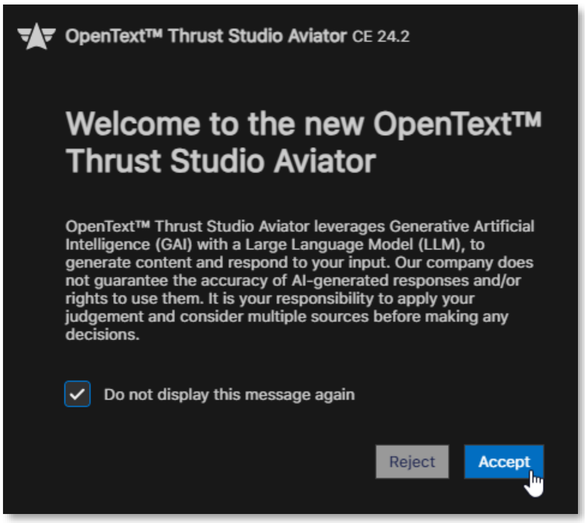
OpenText™ Thrust Studio Aviator leverages Generative Artificial Intelligence (GAI) with a Large Language Model (LLM), to generate content and respond to your input. Our company does not guarantee the accuracy of AI-generated responses and/or rights to use them. It is your responsibility to apply your judgement and consider multiple sources before making any

☐ Do not display this message again

Reject Accept

2. Read the disclaimer and click **Accept** if you accept it.

   You can additionally choose to select **Do not display this message again** so that the disclaimer no longer displays when you open VS Code and use the OpenText™ Thrust Studio Aviator.

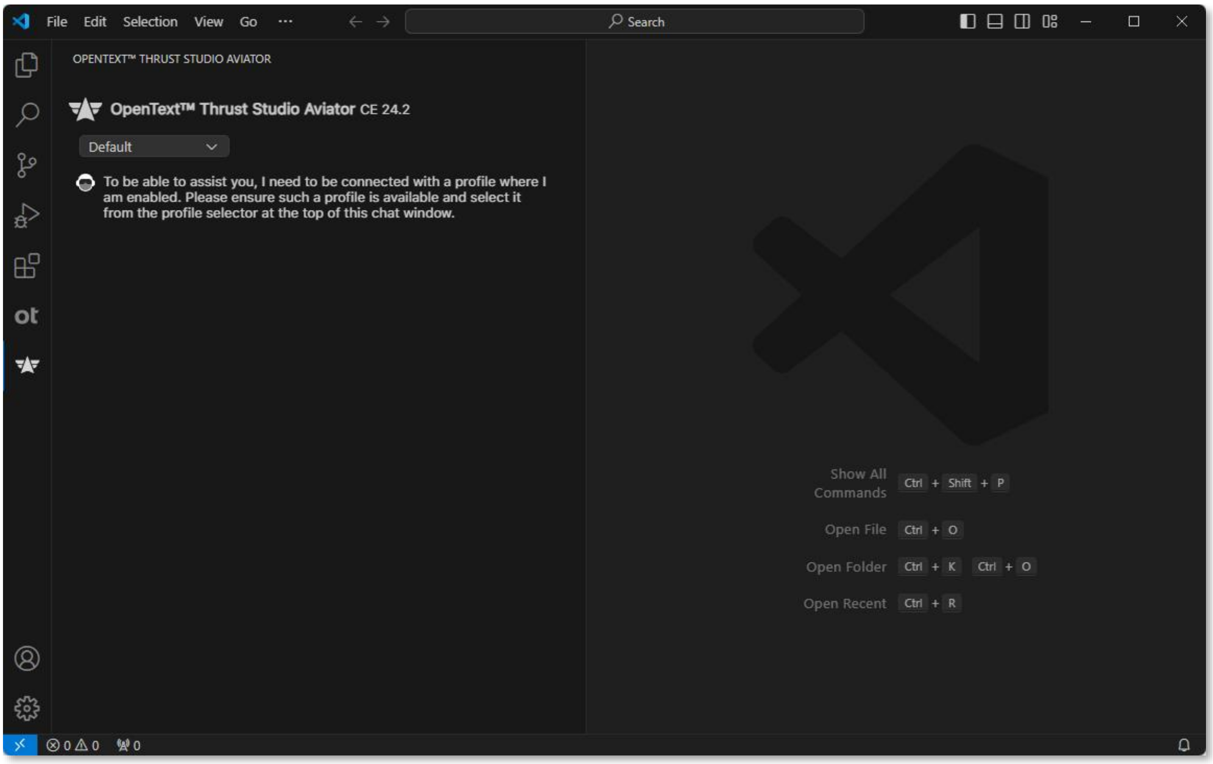


OpenText™ Thrust Studio Aviator CE 24.2

Welcome to the new OpenText™ Thrust Studio Aviator

OpenText™ Thrust Studio Aviator leverages Generative Artificial Intelligence (GAI) with a Large Language Model (LLM), to generate content and respond to your input. Our company does not guarantee the accuracy of AI-generated responses and/or rights to use them. It is your responsibility to apply your judgement and consider multiple sources before making any decisions.

☑ Do not display this message again



OPENTEXT™ THRUST STUDIO AVIATOR

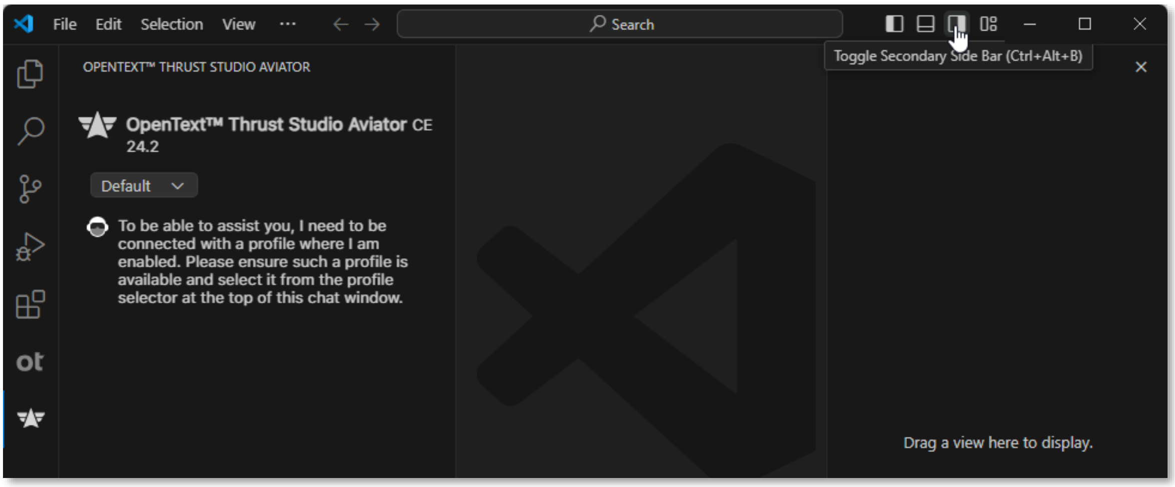
OpenText™ Thrust Studio Aviator CE 24.2

Default

To be able to assist you, I need to be connected with a profile where I am enabled. Please ensure such a profile is available and select it from the profile selector at the top of this chat window.

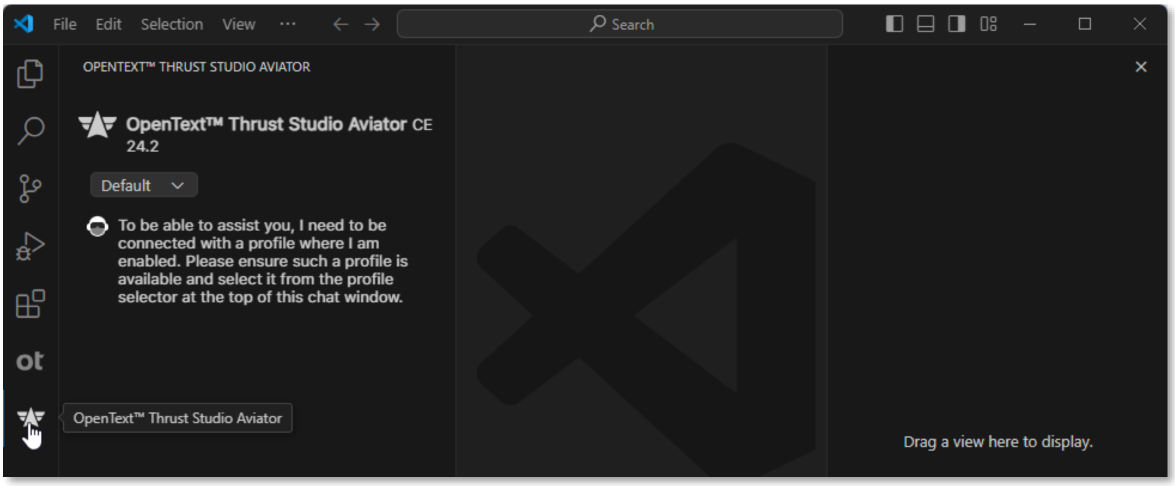
Note that the above screen shot is an example with a separate developer organization that is not US-based. In case your default (developer) organization is the US-based one (i.e., the one where OpenText™ Thrust Studio Aviator is enabled), you will see a different message stating that the Aviator is ready to take your questions.

3. First, for convenience, select to expand the right pane and drag and drop the Thrust Studio Aviator view to that pane (so that you can use the left pane for other activities).


File
Edit
Selection
View
Search
Toggle Secondary Side Bar (Ctrl+Alt+B)
OPENTEXT™ THRUST STUDIO AVIATOR
OpenText™ Thrust Studio Aviator CE
24.2
Default
ot

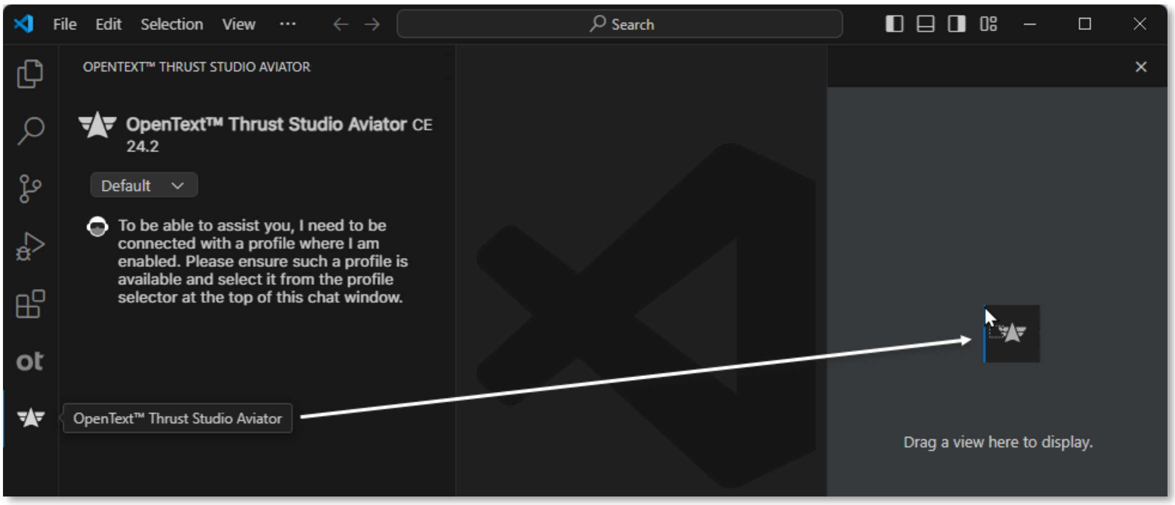
To be able to assist you, I need to be connected with a profile where I am enabled. Please ensure such a profile is available and select it from the profile selector at the top of this chat window.

Drag a view here to display.


File
Edit
Selection
View
Search
OPENTEXT™ THRUST STUDIO AVIATOR
OpenText™ Thrust Studio Aviator CE
24.2
Default
ot
OpenText™ Thrust Studio Aviator

To be able to assist you, I need to be connected with a profile where I am enabled. Please ensure such a profile is available and select it from the profile selector at the top of this chat window.

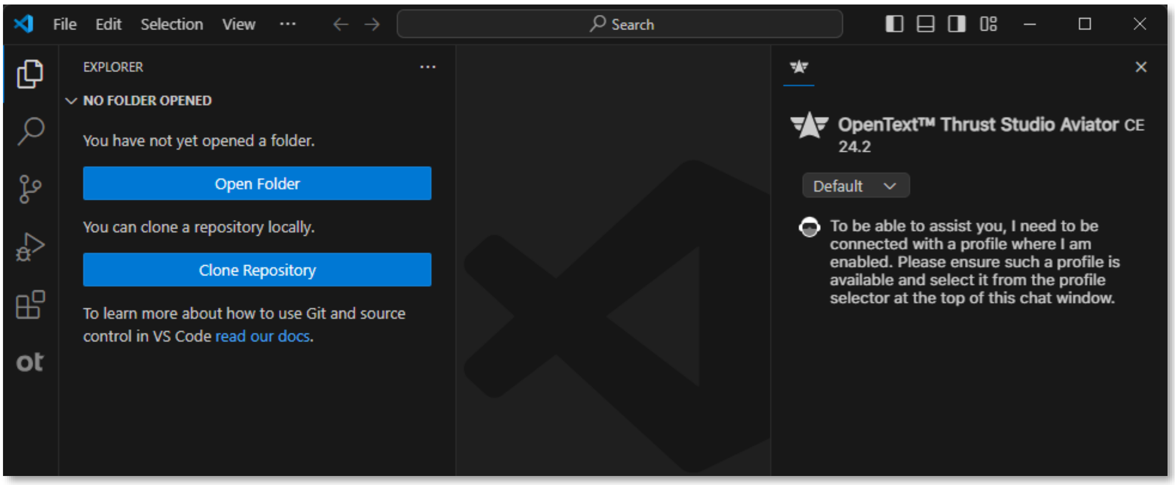
Drag a view here to display.


File
Edit
Selection
View
Search
OPENTEXT™ THRUST STUDIO AVIATOR
OpenText™ Thrust Studio Aviator CE
24.2
Default
ot
OpenText™ Thrust Studio Aviator

To be able to assist you, I need to be connected with a profile where I am enabled. Please ensure such a profile is available and select it from the profile selector at the top of this chat window.

Drag a view here to display.

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.



You have not yet opened a folder.

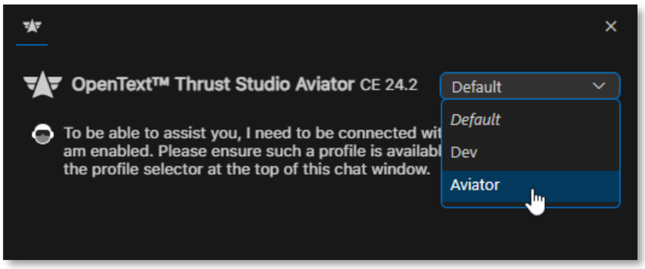
You can clone a repository locally.

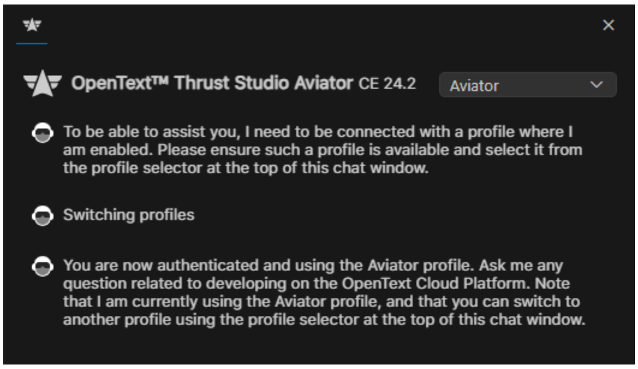
To learn more about how to use Git and source control in VS Code read our docs.

OpenText™ Thrust Studio Aviator CE 24.2

To be able to assist you, I need to be connected with a profile where I am enabled. Please ensure such a profile is available and select it from the profile selector at the top of this chat window.

4. In case the default organization is not the US-based one, from the Thrust Studio Aviator organization profile dropdown, select the **Aviator** organization.



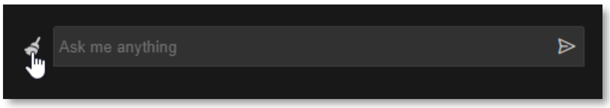


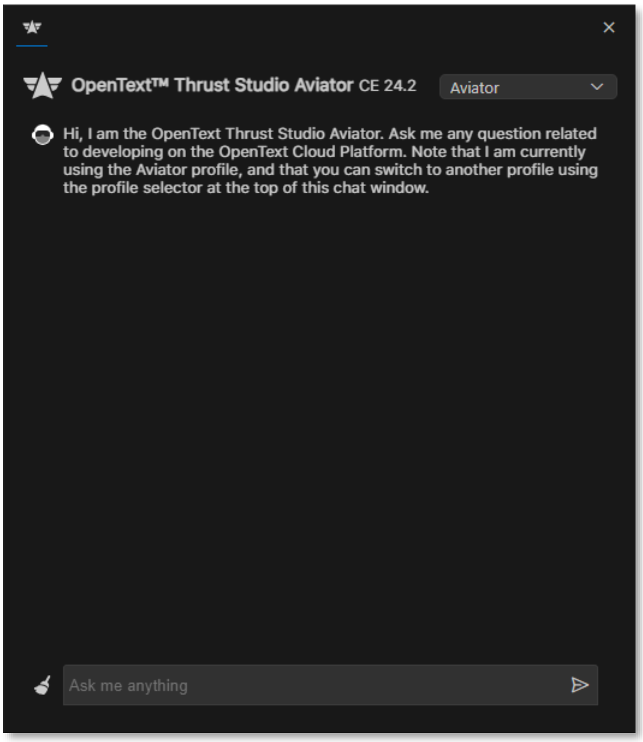
To be able to assist you, I need to be connected with a profile where I am enabled. Please ensure such a profile is available and select it from the profile selector at the top of this chat window.

Switching profiles

You are now authenticated and using the Aviator profile. Ask me any question related to developing on the OpenText Cloud Platform. Note that I am currently using the Aviator profile, and that you can switch to another profile using the profile selector at the top of this chat window.

5. As a last step, you can clear/reset the chat panel with the brush (🧹) button next to the chat input box to have a nice clean chat window.





Hi, I am the OpenText Thrust Studio Aviator. Ask me any question related to developing on the OpenText Cloud Platform. Note that I am currently using the Aviator profile, and that you can switch to another profile using the profile selector at the top of this chat window.

6. You are now ready to start asking questions to the OpenText™ Thrust Studio Aviator.

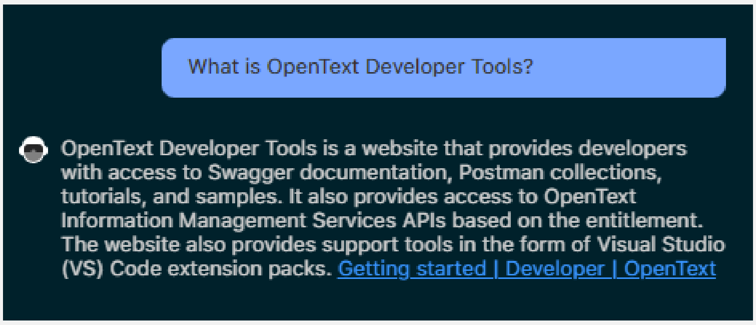
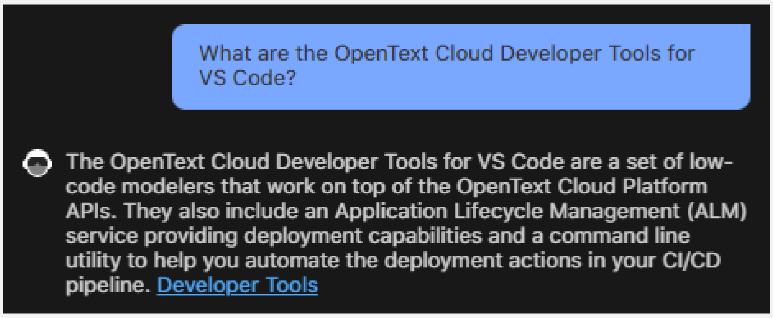
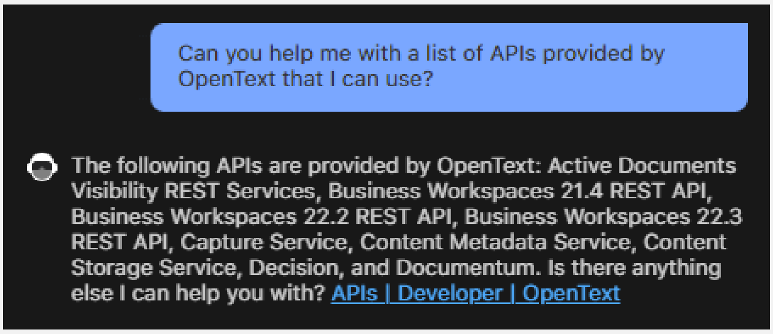
Next step:

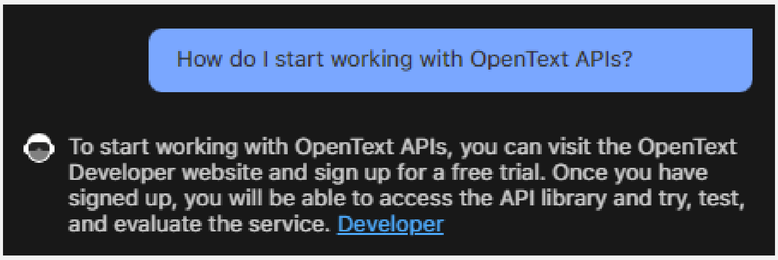
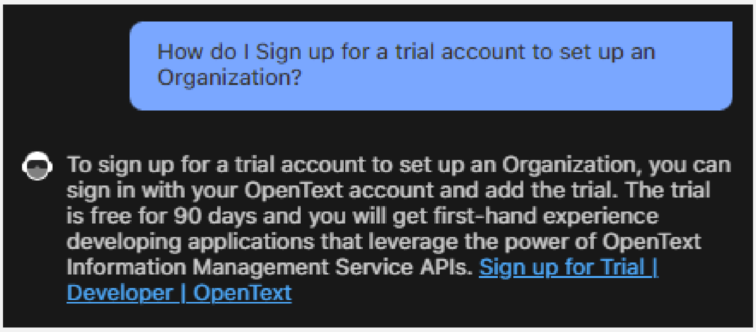
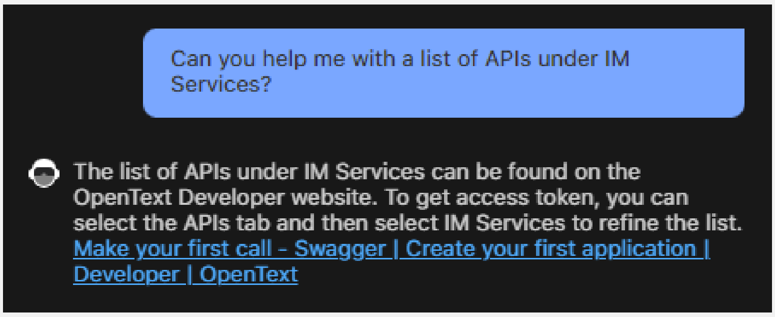
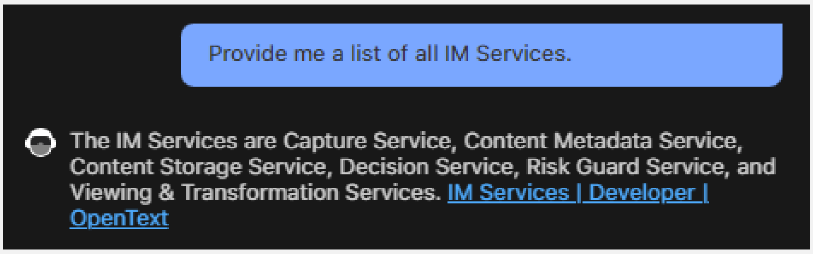
A developer conversation with Thrust Studio Aviator.

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

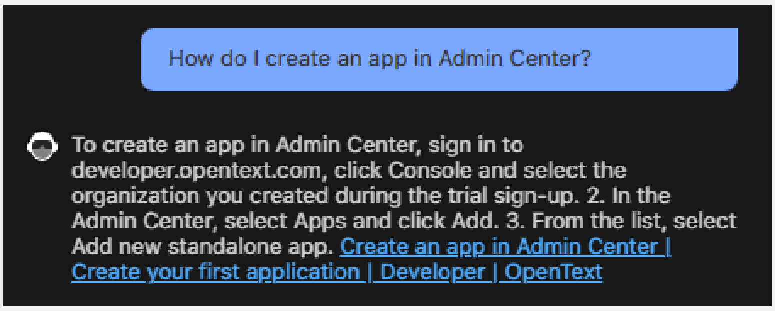
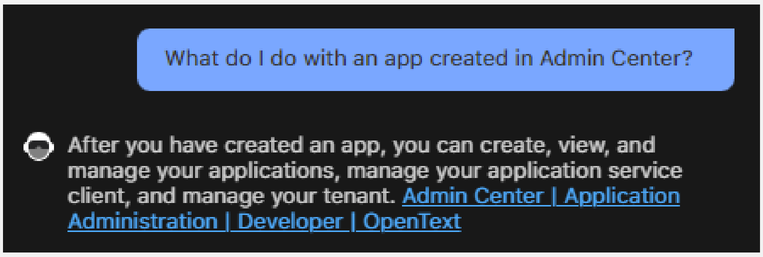
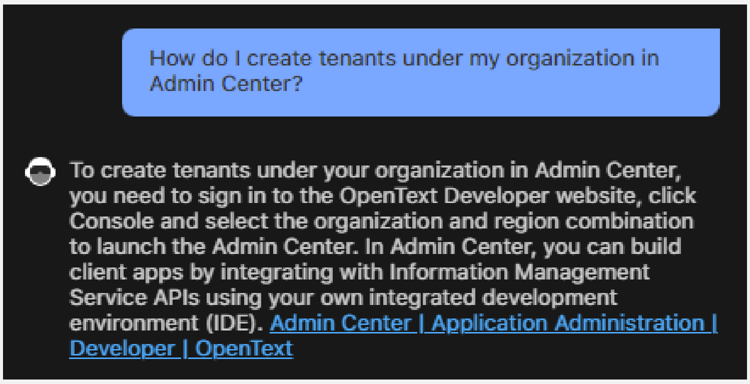
3.2 A developer conversation with Thrust Studio Aviator

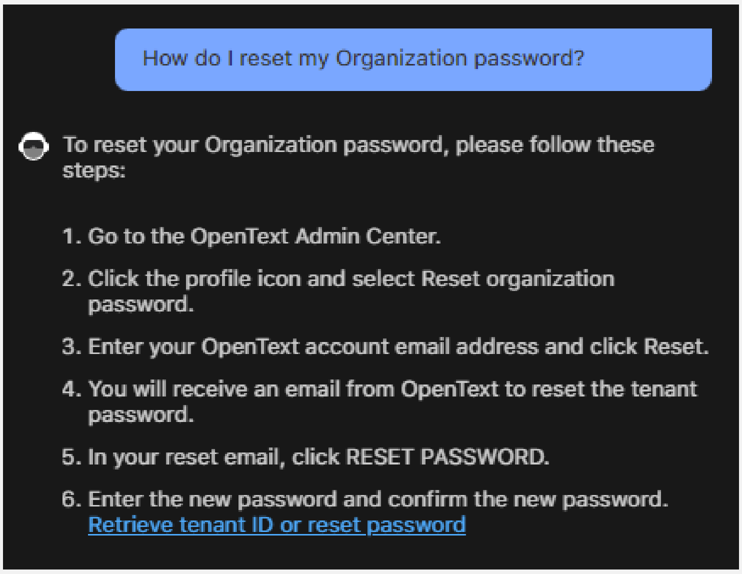
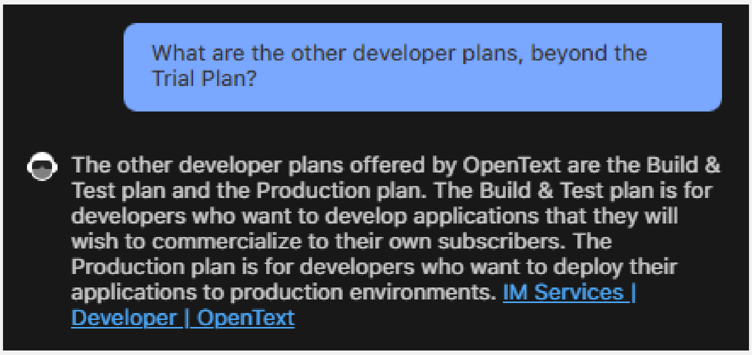
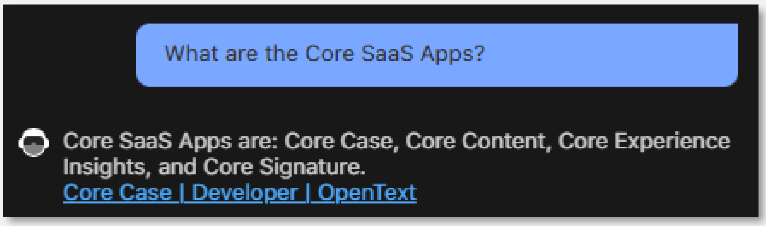
Please ask the following questions to the Thrust Studio Aviator to see how it responds. We have also provided the responses (as “Our test answer”) we have observed on our test run for your convenience.

<table>
<tr><td>Question</td><td>What is OpenText Developer Tools?</td></tr>
<tr><td>Our test answer</td><td>What is OpenText Developer Tools?
OpenText Developer Tools is a website that provides developers with access to Swagger documentation, Postman collections, tutorials, and samples. It also provides access to OpenText Information Management Services APIs based on the entitlement. The website also provides support tools in the form of Visual Studio (VS) Code extension packs. Getting started | Developer | OpenText</td></tr>
<tr><td>Question</td><td>What are the OpenText Cloud Developer Tools for VS Code?</td></tr>
<tr><td>Our test answer</td><td>What are the OpenText Cloud Developer Tools for VS Code?
The OpenText Cloud Developer Tools for VS Code are a set of low-code modelers that work on top of the OpenText Cloud Platform APIs. They also include an Application Lifecycle Management (ALM) service providing deployment capabilities and a command line utility to help you automate the deployment actions in your CI/CD pipeline. Developer Tools</td></tr>
<tr><td>Question</td><td>Can you help me with a list of APIs provided by OpenText that I can use?</td></tr>
<tr><td>Our test answer</td><td>Can you help me with a list of APIs provided by OpenText that I can use?
The following APIs are provided by OpenText: Active Documents Visibility REST Services, Business Workspaces 21.4 REST API, Business Workspaces 22.2 REST API, Business Workspaces 22.3 REST API, Capture Service, Content Metadata Service, Content Storage Service, Decision, and Documentum. Is there anything else I can help you with? APIs | Developer | OpenText</td></tr>
</table>

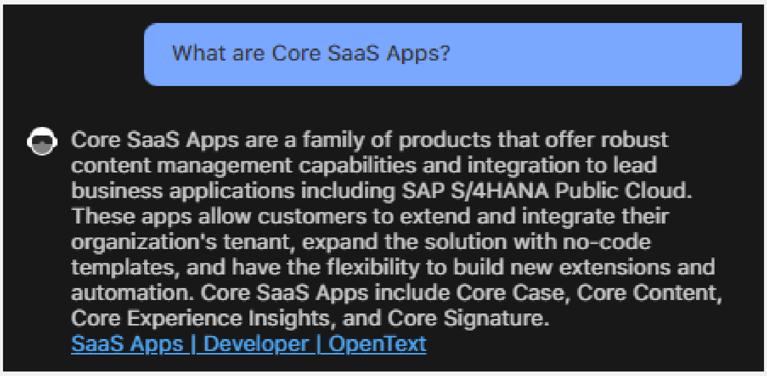
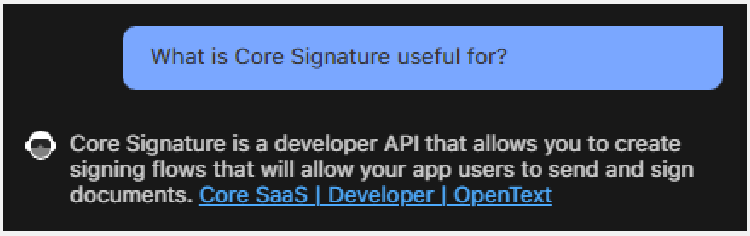
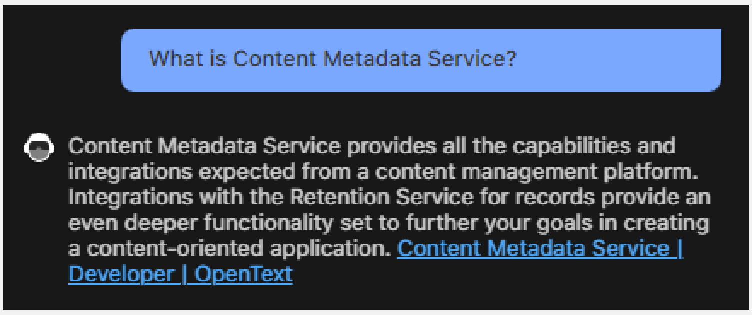
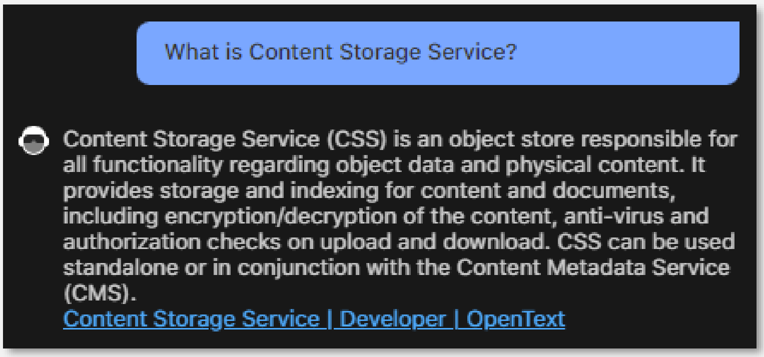
Question	How do I start working with OpenText APIs?
Our test answer	 How do I start working with OpenText APIs? To start working with OpenText APIs, you can visit the OpenText Developer website and sign up for a free trial. Once you have signed up, you will be able to access the API library and try, test, and evaluate the service. Developer
Question	How do I Sign up for a trial account to set up an Organization?
Our test answer	 How do I Sign up for a trial account to set up an Organization? To sign up for a trial account to set up an Organization, you can sign in with your OpenText account and add the trial. The trial is free for 90 days and you will get first-hand experience developing applications that leverage the power of OpenText Information Management Service APIs. Sign up for Trial | Developer | OpenText
Question	Can you help me with a list of APIs under IM Services?
Our test answer	 Can you help me with a list of APIs under IM Services? The list of APIs under IM Services can be found on the OpenText Developer website. To get access token, you can select the APIs tab and then select IM Services to refine the list. Make your first call - Swagger | Create your first application | Developer | OpenText
Question	Provide me a list of all IM Services.
Our test answer	 Provide me a list of all IM Services. The IM Services are Capture Service, Content Metadata Service, Content Storage Service, Decision Service, Risk Guard Service, and Viewing & Transformation Services. IM Services | Developer | OpenText

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

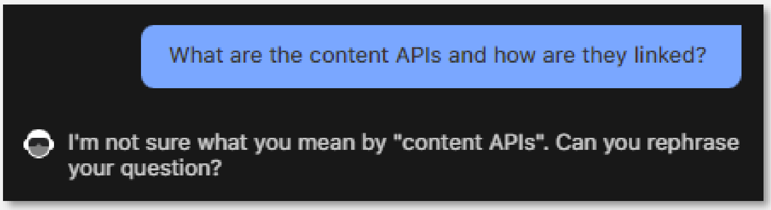
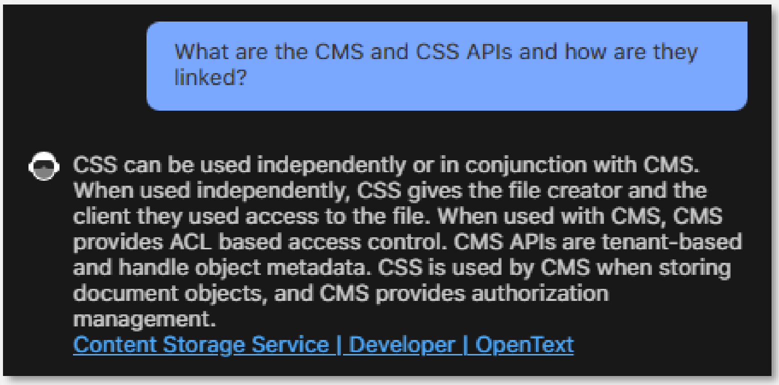
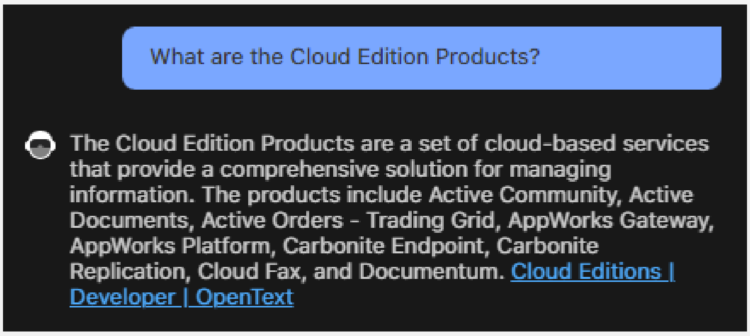
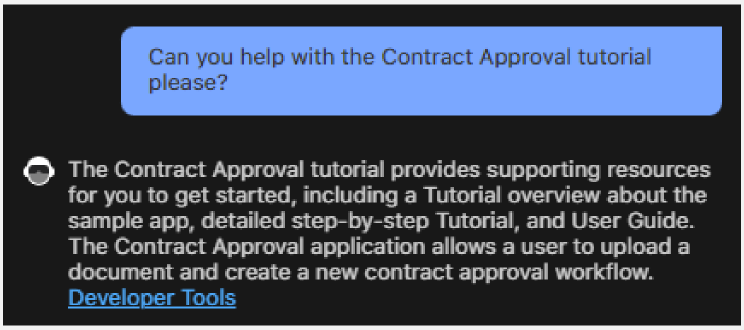
Question	How do I create an app in Admin Center?
Our test answer	 How do I create an app in Admin Center? To create an app in Admin Center, sign in to developer.opentext.com, click Console and select the organization you created during the trial sign-up. 2. In the Admin Center, select Apps and click Add. 3. From the list, select Add new standalone app. Create an app in Admin Center | Create your first application | Developer | OpenText
Question	What do I do with an app created in Admin Center?
Our test answer	 What do I do with an app created in Admin Center? After you have created an app, you can create, view, and manage your applications, manage your application service client, and manage your tenant. Admin Center | Application Administration | Developer | OpenText
Question	How do I create tenants under my organization in Admin Center?
Our test answer	 How do I create tenants under my organization in Admin Center? To create tenants under your organization in Admin Center, you need to sign in to the OpenText Developer website, click Console and select the organization and region combination to launch the Admin Center. In Admin Center, you can build client apps by integrating with Information Management Service APIs using your own integrated development environment (IDE). Admin Center | Application Administration | Developer | OpenText

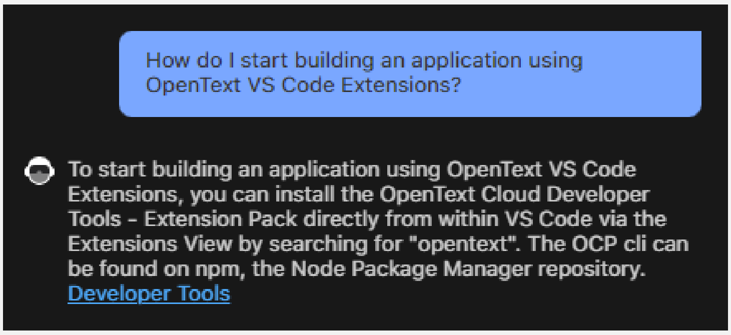
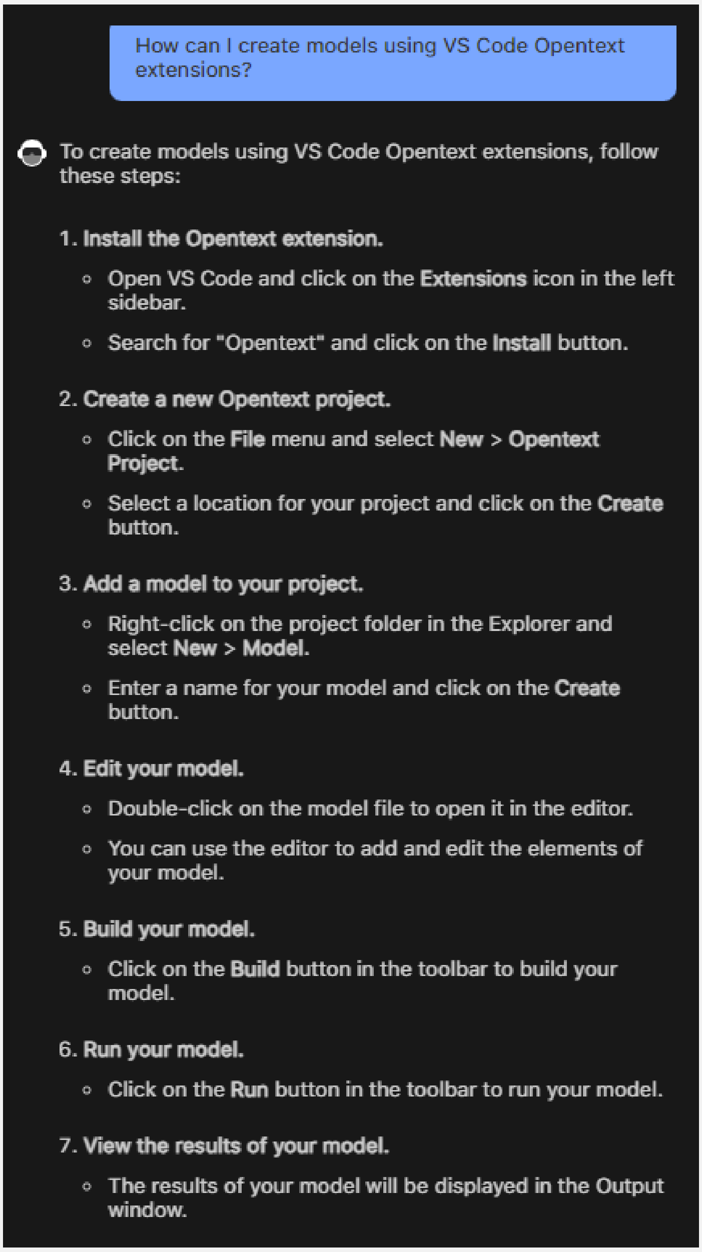
Question	How do I reset my Organization password?
Our test answer	 How do I reset my Organization password? To reset your Organization password, please follow these steps: 1. Go to the OpenText Admin Center. 2. Click the profile icon and select Reset organization password. 3. Enter your OpenText account email address and click Reset. 4. You will receive an email from OpenText to reset the tenant password. 5. In your reset email, click RESET PASSWORD. 6. Enter the new password and confirm the new password. Retrieve tenant ID or reset password
Question	What are the other developer plans, beyond the Trial Plan?
Our test answer	 What are the other developer plans, beyond the Trial Plan? The other developer plans offered by OpenText are the Build & Test plan and the Production plan. The Build & Test plan is for developers who want to develop applications that they will wish to commercialize to their own subscribers. The Production plan is for developers who want to deploy their applications to production environments. IM Services | Developer | OpenText
Question	What are the Core SaaS Apps?
Our test answer	 What are the Core SaaS Apps? Core SaaS Apps are: Core Case, Core Content, Core Experience Insights, and Core Signature. Core Case | Developer | OpenText

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

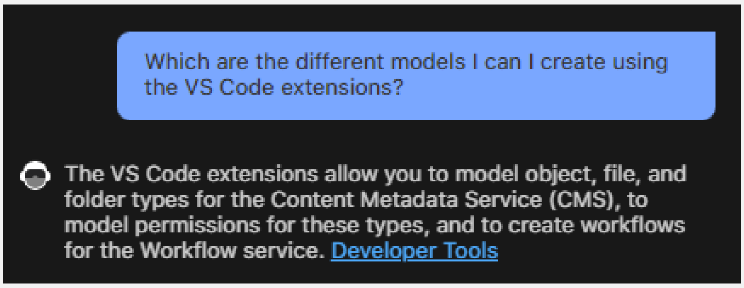
<table>
<tr><td>Question</td><td>What are Core SaaS Apps?</td></tr>
<tr><td>Our test answer</td><td>
What are Core SaaS Apps?
Core SaaS Apps are a family of products that offer robust content management capabilities and integration to lead business applications including SAP S/4HANA Public Cloud. These apps allow customers to extend and integrate their organization's tenant, expand the solution with no-code templates, and have the flexibility to build new extensions and automation. Core SaaS Apps include Core Case, Core Content, Core Experience Insights, and Core Signature.
SaaS Apps | Developer | OpenText</td></tr>
<tr><td>Question</td><td>What is Core Signature useful for?</td></tr>
<tr><td>Our test answer</td><td>
What is Core Signature useful for?
Core Signature is a developer API that allows you to create signing flows that will allow your app users to send and sign documents. Core SaaS | Developer | OpenText</td></tr>
<tr><td>Question</td><td>What is Content Metadata Service?</td></tr>
<tr><td>Our test answer</td><td>
What is Content Metadata Service?
Content Metadata Service provides all the capabilities and integrations expected from a content management platform. Integrations with the Retention Service for records provide an even deeper functionality set to further your goals in creating a content-oriented application. Content Metadata Service | Developer | OpenText</td></tr>
<tr><td>Question</td><td>What is Content Storage Service?</td></tr>
<tr><td>Our test answer</td><td>
What is Content Storage Service?
Content Storage Service (CSS) is an object store responsible for all functionality regarding object data and physical content. It provides storage and indexing for content and documents, including encryption/decryption of the content, anti-virus and authorization checks on upload and download. CSS can be used standalone or in conjunction with the Content Metadata Service (CMS).
Content Storage Service | Developer | OpenText</td></tr>
</table>

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

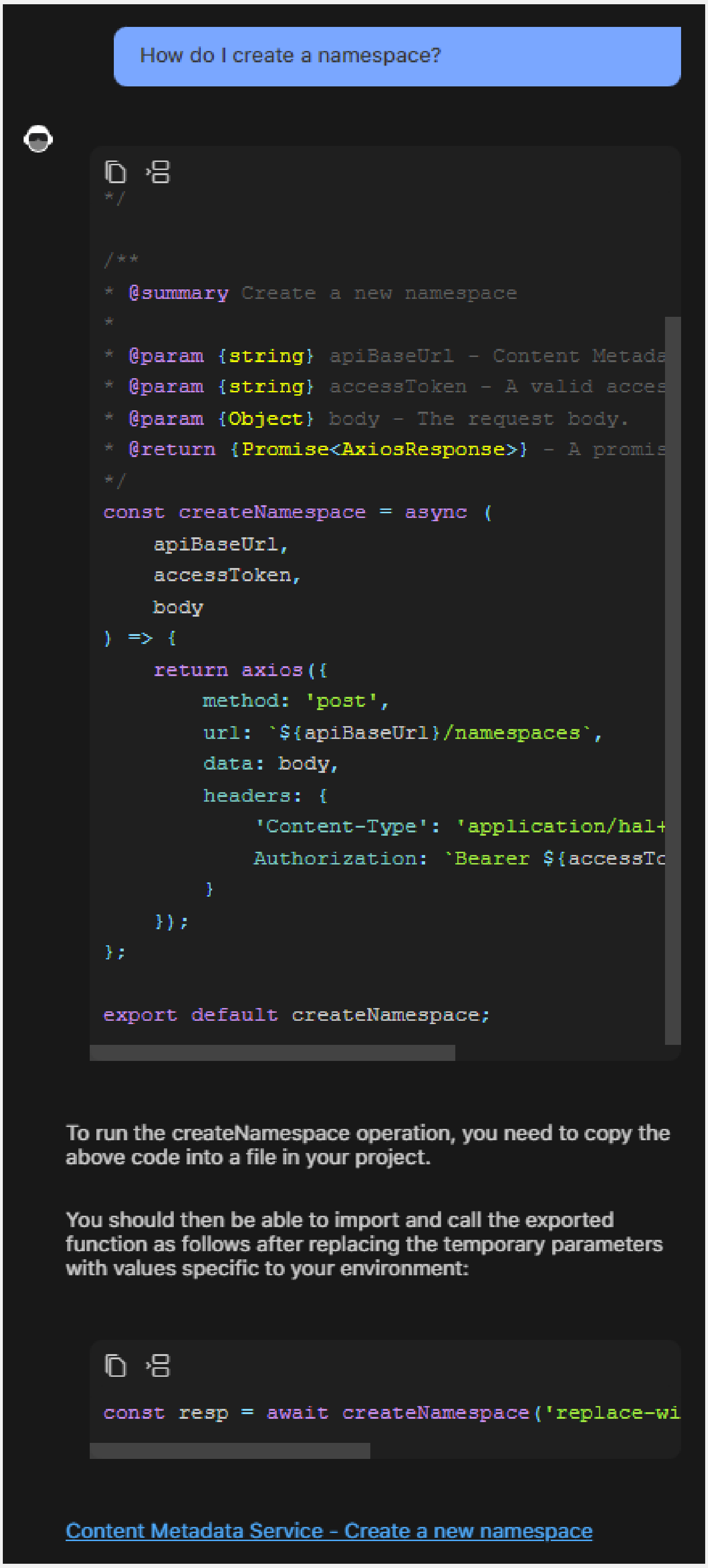
Question	What are the content APIs and how are they linked?
Our test answer	 What are the content APIs and how are they linked? I'm not sure what you mean by "content APIs". Can you rephrase your question?
Question	What are the CMS and CSS APIs and how are they linked?
Our test answer	 What are the CMS and CSS APIs and how are they linked? CSS can be used independently or in conjunction with CMS. When used independently, CSS gives the file creator and the client they used access to the file. When used with CMS, CMS provides ACL based access control. CMS APIs are tenant-based and handle object metadata. CSS is used by CMS when storing document objects, and CMS provides authorization management. Content Storage Service | Developer | OpenText
Question	What are the Cloud Edition Products?
Our test answer	 What are the Cloud Edition Products? The Cloud Edition Products are a set of cloud-based services that provide a comprehensive solution for managing information. The products include Active Community, Active Documents, Active Orders - Trading Grid, AppWorks Gateway, AppWorks Platform, Carbonite Endpoint, Carbonite Replication, Cloud Fax, and Documentum. Cloud Editions | Developer | OpenText
Question	Can you help with the Contract Approval tutorial please?
Our test answer	 Can you help with the Contract Approval tutorial please? The Contract Approval tutorial provides supporting resources for you to get started, including a Tutorial overview about the sample app, detailed step-by-step Tutorial, and User Guide. The Contract Approval application allows a user to upload a document and create a new contract approval workflow. Developer Tools

Question	How do I start building an application using OpenText VS Code Extensions?
Our test answer	 How do I start building an application using OpenText VS Code Extensions? To start building an application using OpenText VS Code Extensions, you can install the OpenText Cloud Developer Tools - Extension Pack directly from within VS Code via the Extensions View by searching for "opentext". The OCP cli can be found on npm, the Node Package Manager repository. Developer Tools
Question	How can I create models using VS Code Opentext extensions?
Our test answer	 How can I create models using VS Code Opentext extensions? To create models using VS Code Opentext extensions, follow these steps: 1. **Install the Opentext extension.** ∘ Open VS Code and click on the **Extensions** icon in the left sidebar. ∘ Search for "Opentext" and click on the **Install** button. 2. **Create a new Opentext project.** ∘ Click on the **File** menu and select **New** > **Opentext Project**. ∘ Select a location for your project and click on the **Create** button. 3. **Add a model to your project.** ∘ Right-click on the project folder in the Explorer and select **New** > **Model**. ∘ Enter a name for your model and click on the **Create** button. 4. **Edit your model.** ∘ Double-click on the model file to open it in the editor. ∘ You can use the editor to add and edit the elements of your model. 5. **Build your model.** ∘ Click on the **Build** button in the toolbar to build your model. 6. **Run your model.** ∘ Click on the **Run** button in the toolbar to run your model. 7. **View the results of your model.** ∘ The results of your model will be displayed in the Output window.

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

Question	Which are the different models I can I create using the VS Code extensions?
Our test answer	 Which are the different models I can I create using the VS Code extensions? The VS Code extensions allow you to model object, file, and folder types for the Content Metadata Service (CMS), to model permissions for these types, and to create workflows for the Workflow service. Developer Tools

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

Question	How do I create a namespace?
Our test answer	

How do I create a namespace?

```
*/


/**
* @summary Create a new namespace
*
* @param {string} apiBaseUrl - Content Metada
* @param {string} accessToken - A valid acces
* @param {Object} body - The request body.
* @return {Promise<AxiosResponse>} - A promis
*/
const createNamespace = async (
    apiBaseUrl,
    accessToken,
    body
) => {
    return axios({
        method: 'post',
        url: `${apiBaseUrl}/namespaces`,
        data: body,
        headers: {
            'Content-Type': 'application/hal+
            Authorization: `Bearer ${accessTo
        }
    });
};

export default createNamespace;
```

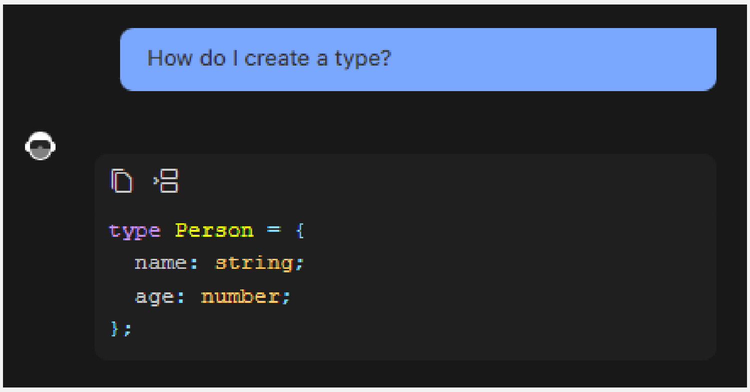
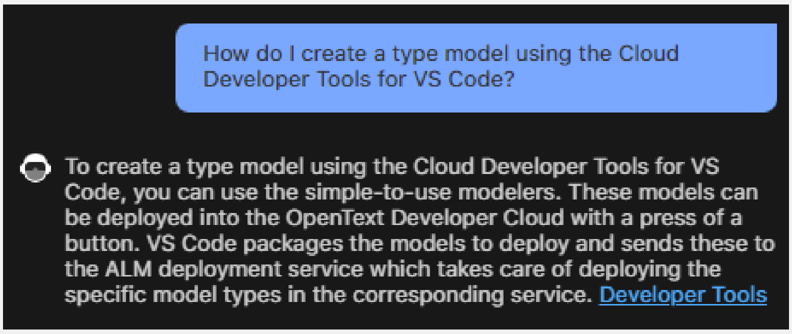
To run the createNamespace operation, you need to copy the above code into a file in your project.

You should then be able to import and call the exported function as follows after replacing the temporary parameters with values specific to your environment:

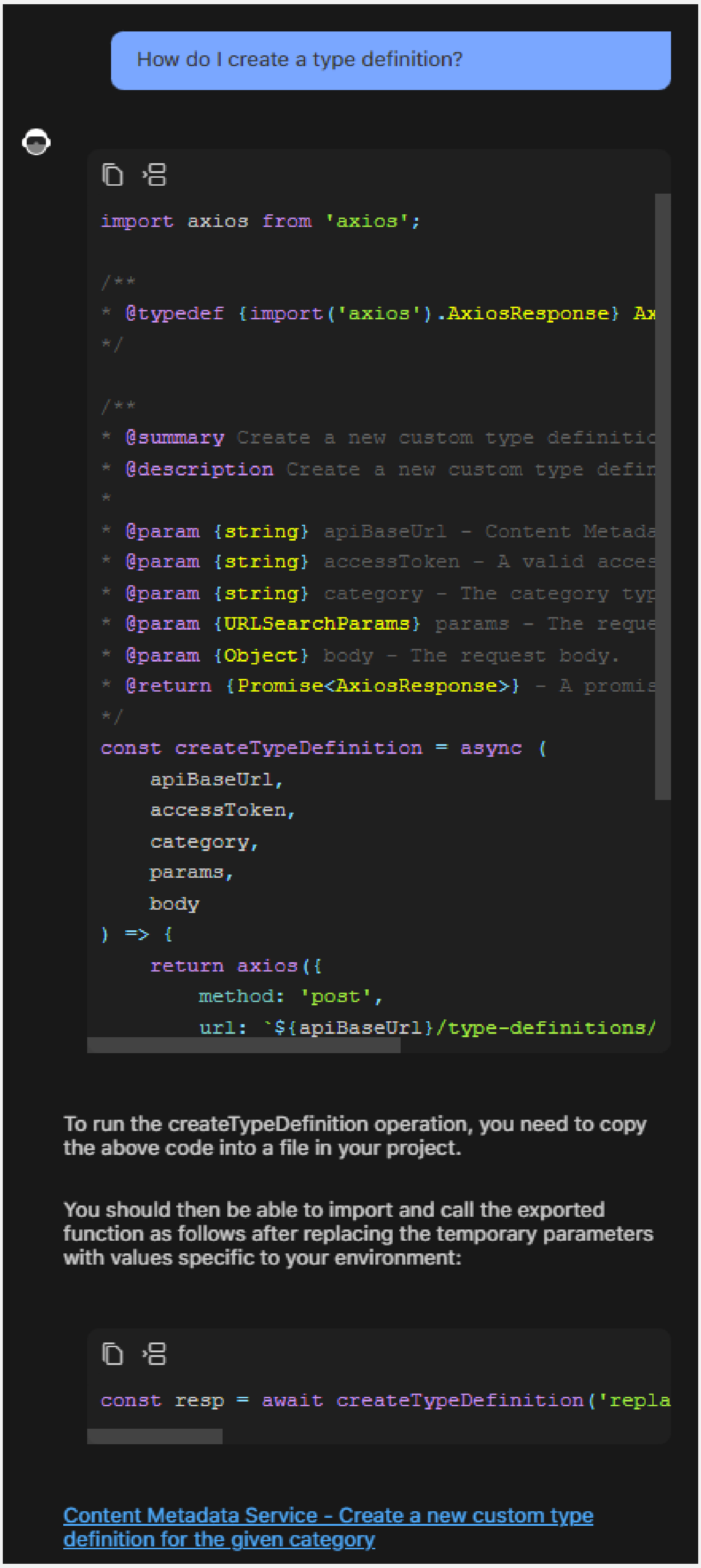
```
const resp = await createNamespace('replace-wi
```

Content Metadata Service - Create a new namespace

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

<table>
<tr><td>Question</td><td>How do I create a type?</td></tr>
<tr><td>Our test answer</td><td>
How do I create a type?
<pre><code>type Person = {
  name: string;
  age: number;
};</code></pre></td></tr>
<tr><td>Question</td><td>How do I create a type model using the Cloud Developer Tools for VS Code?</td></tr>
<tr><td>Our test answer</td><td>
How do I create a type model using the Cloud Developer Tools for VS Code?
To create a type model using the Cloud Developer Tools for VS Code, you can use the simple-to-use modelers. These models can be deployed into the OpenText Developer Cloud with a press of a button. VS Code packages the models to deploy and sends these to the ALM deployment service which takes care of deploying the specific model types in the corresponding service. Developer Tools</td></tr>
</table>

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

Question	How do I create a type definition?
Our test answer	

How do I create a type definition?

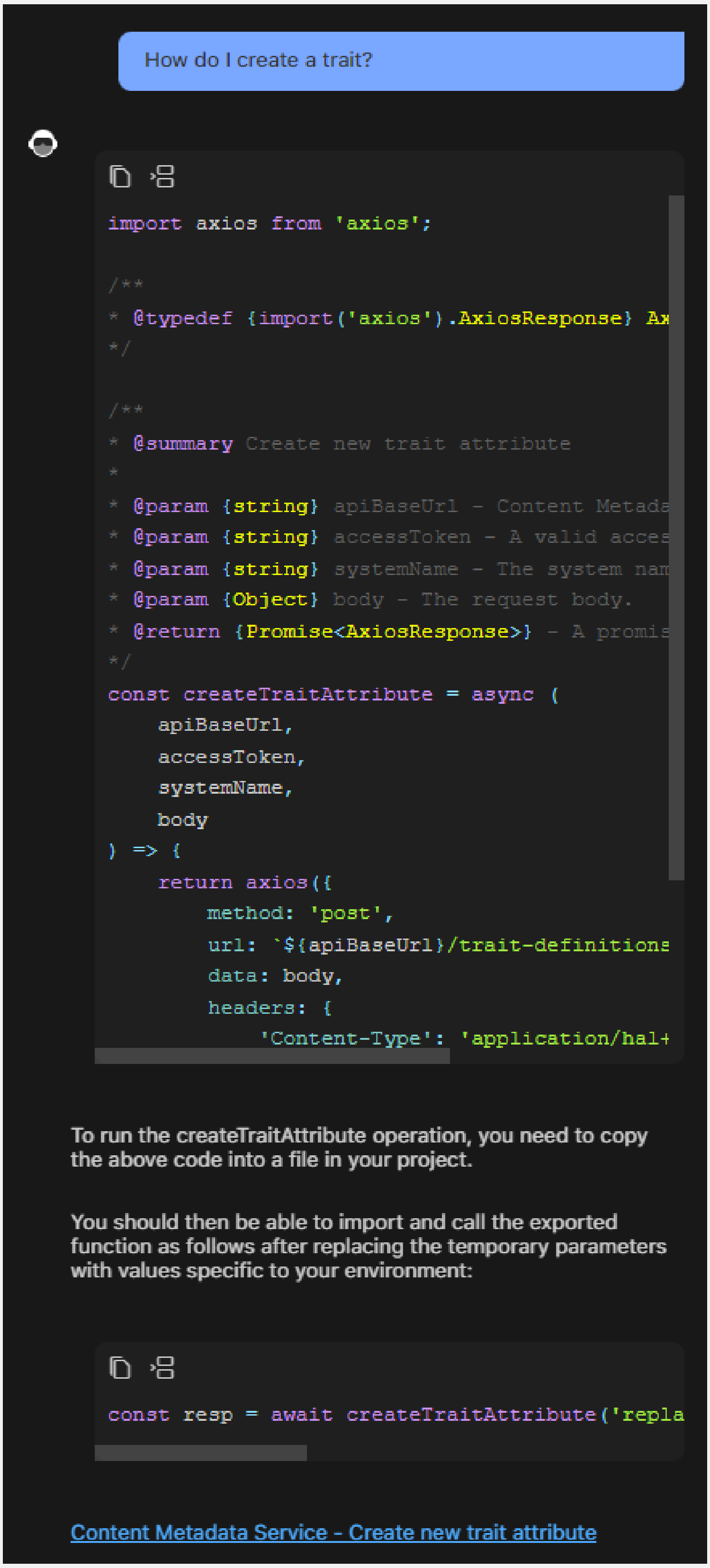
```
import axios from 'axios';

/**
* @typedef {import('axios').AxiosResponse} Ax
*/

/**
* @summary Create a new custom type definitic
* @description Create a new custom type defir
*
* @param {string} apiBaseUrl - Content Metada
* @param {string} accessToken - A valid acces
* @param {string} category - The category typ
* @param {URLSearchParams} params - The reque
* @param {Object} body - The request body.
* @return {Promise<AxiosResponse>} - A promis
*/
const createTypeDefinition = async (
    apiBaseUrl,
    accessToken,
    category,
    params,
    body
) => {
    return axios({
        method: 'post',
        url: `${apiBaseUrl}/type-definitions/
```

To run the createTypeDefinition operation, you need to copy the above code into a file in your project.

You should then be able to import and call the exported function as follows after replacing the temporary parameters with values specific to your environment:

```
const resp = await createTypeDefinition('repla
```

[Content Metadata Service - Create a new custom type definition for the given category]()

Question	How do I create a trait?
Our test answer	

How do I create a trait?

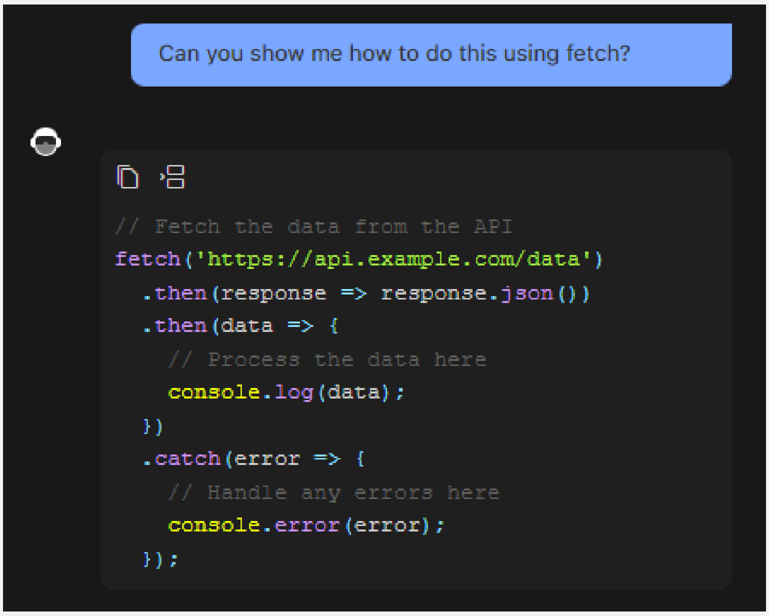
```
import axios from 'axios';

/**
* @typedef {import('axios').AxiosResponse} Ax
*/

/**
* @summary Create new trait attribute
*
* @param {string} apiBaseUrl - Content Metada
* @param {string} accessToken - A valid acces
* @param {string} systemName - The system nam
* @param {Object} body - The request body.
* @return {Promise<AxiosResponse>} - A promis
*/
const createTraitAttribute = async (
    apiBaseUrl,
    accessToken,
    systemName,
    body
) => {
    return axios({
        method: 'post',
        url: `${apiBaseUrl}/trait-definitions
        data: body,
        headers: {
            'Content-Type': 'application/hal+
```

To run the createTraitAttribute operation, you need to copy the above code into a file in your project.

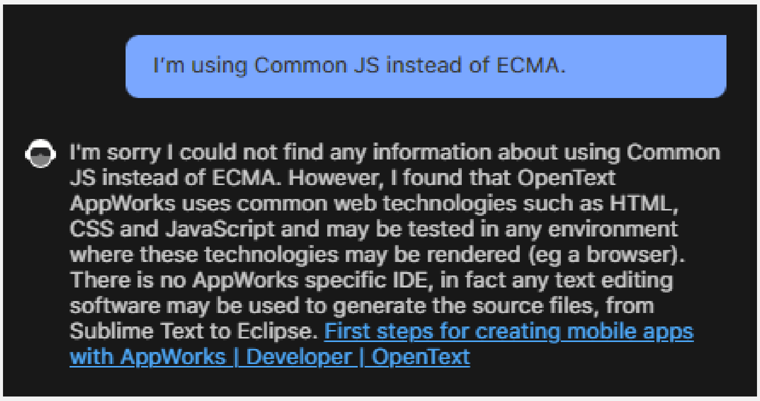
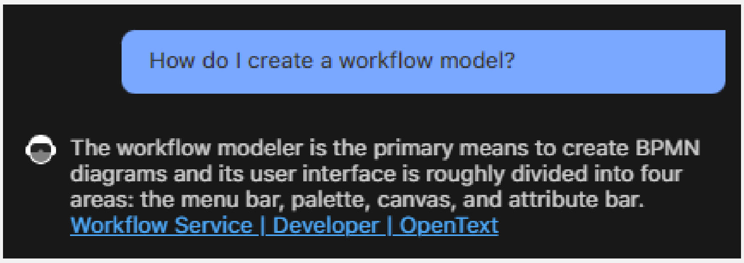
You should then be able to import and call the exported function as follows after replacing the temporary parameters with values specific to your environment:

```
const resp = await createTraitAttribute('repla
```

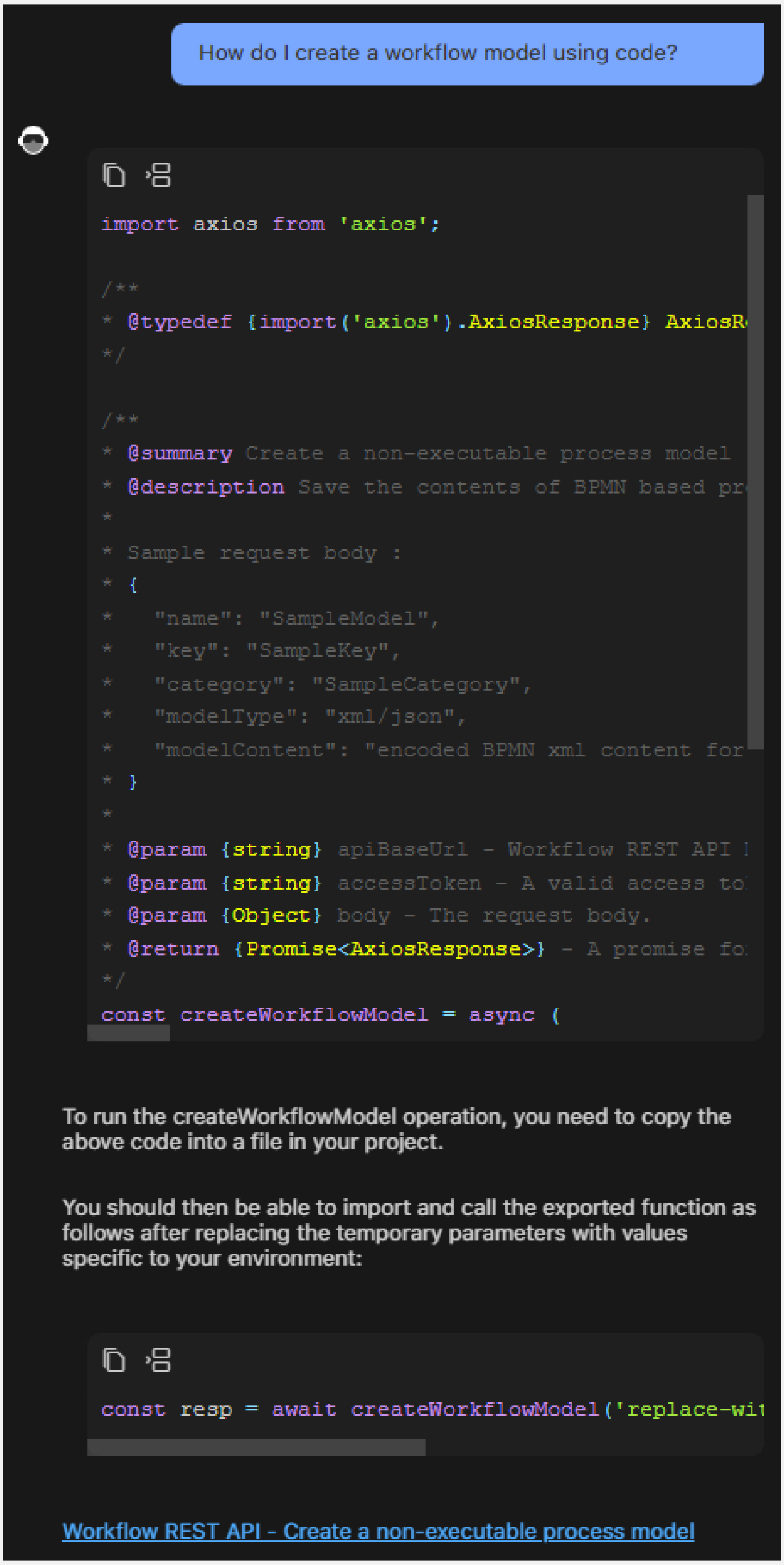
[Content Metadata Service - Create new trait attribute]()

Question	Can you show me how to do this using fetch?
Our test answer	 Can you show me how to do this using fetch?

```
// Fetch the data from the API
fetch('https://api.example.com/data')
  .then(response => response.json())
  .then(data => {
    // Process the data here
    console.log(data);
  })
  .catch(error => {
    // Handle any errors here
    console.error(error);
  });
```

Question	I’m using Common JS instead of ECMA.
Our test answer	 I’m using Common JS instead of ECMA. I'm sorry I could not find any information about using Common JS instead of ECMA. However, I found that OpenText AppWorks uses common web technologies such as HTML, CSS and JavaScript and may be tested in any environment where these technologies may be rendered (eg a browser). There is no AppWorks specific IDE, in fact any text editing software may be used to generate the source files, from Sublime Text to Eclipse. First steps for creating mobile apps with AppWorks | Developer | OpenText
Question	How do I create a workflow model?
Our test answer	 How do I create a workflow model? The workflow modeler is the primary means to create BPMN diagrams and its user interface is roughly divided into four areas: the menu bar, palette, canvas, and attribute bar. Workflow Service | Developer | OpenText

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

Question	How do I create a workflow model using code?
Our test answer	

How do I create a workflow model using code?

```
import axios from 'axios';

/**
* @typedef {import('axios').AxiosResponse} AxiosR
*/

/**
* @summary Create a non-executable process model
* @description Save the contents of BPMN based pr
*
* Sample request body :
* {
*   "name": "SampleModel",
*   "key": "SampleKey",
*   "category": "SampleCategory",
*   "modelType": "xml/json",
*   "modelContent": "encoded BPMN xml content for
* }
*
* @param {string} apiBaseUrl - Workflow REST API
* @param {string} accessToken - A valid access to
* @param {Object} body - The request body.
* @return {Promise<AxiosResponse>} - A promise fo
*/
const createWorkflowModel = async (
```

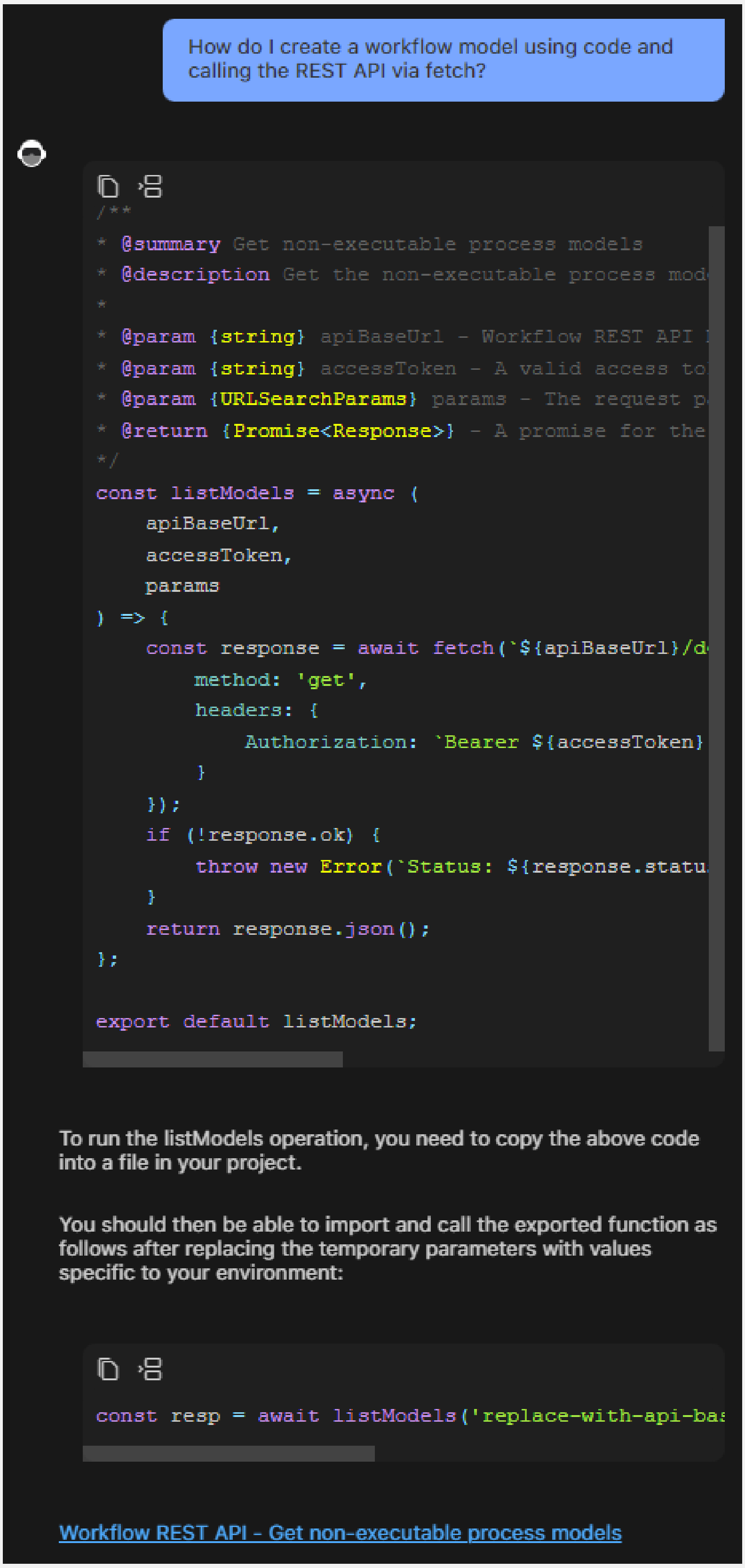
To run the createWorkflowModel operation, you need to copy the above code into a file in your project.

You should then be able to import and call the exported function as follows after replacing the temporary parameters with values specific to your environment:

```
const resp = await createWorkflowModel('replace-wit
```

[Workflow REST API - Create a non-executable process model]

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

Question	How do I create a workflow model using code and calling the REST API via fetch?
Our test answer	

How do I create a workflow model using code and calling the REST API via fetch?

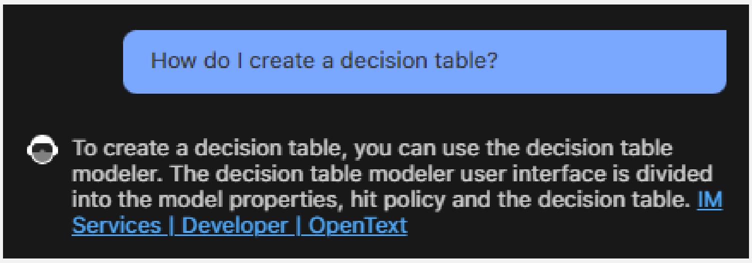
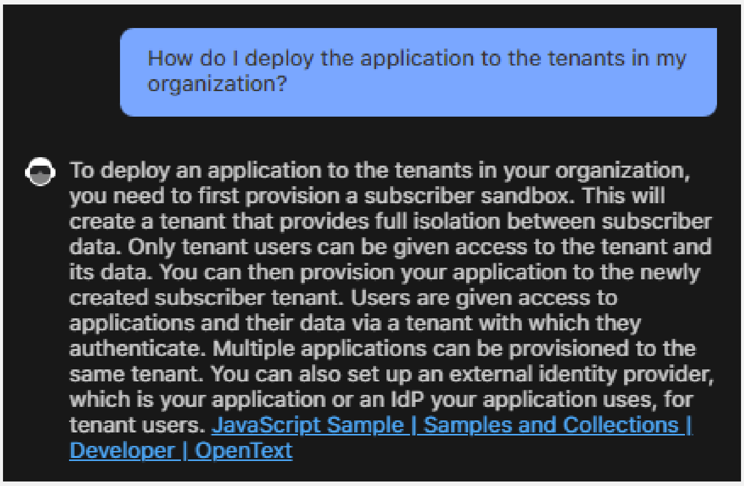
```
/**
* @summary Get non-executable process models
* @description Get the non-executable process mod
*
* @param {string} apiBaseUrl - Workflow REST API 
* @param {string} accessToken - A valid access to
* @param {URLSearchParams} params - The request p
* @return {Promise<Response>} - A promise for the
*/
const listModels = async (
    apiBaseUrl,
    accessToken,
    params
) => {
    const response = await fetch(`${apiBaseUrl}/d
        method: 'get',
        headers: {
            Authorization: `Bearer ${accessToken}
        }
    });
    if (!response.ok) {
        throw new Error(`Status: ${response.statu
    }
    return response.json();
};

export default listModels;
```

To run the listModels operation, you need to copy the above code into a file in your project.

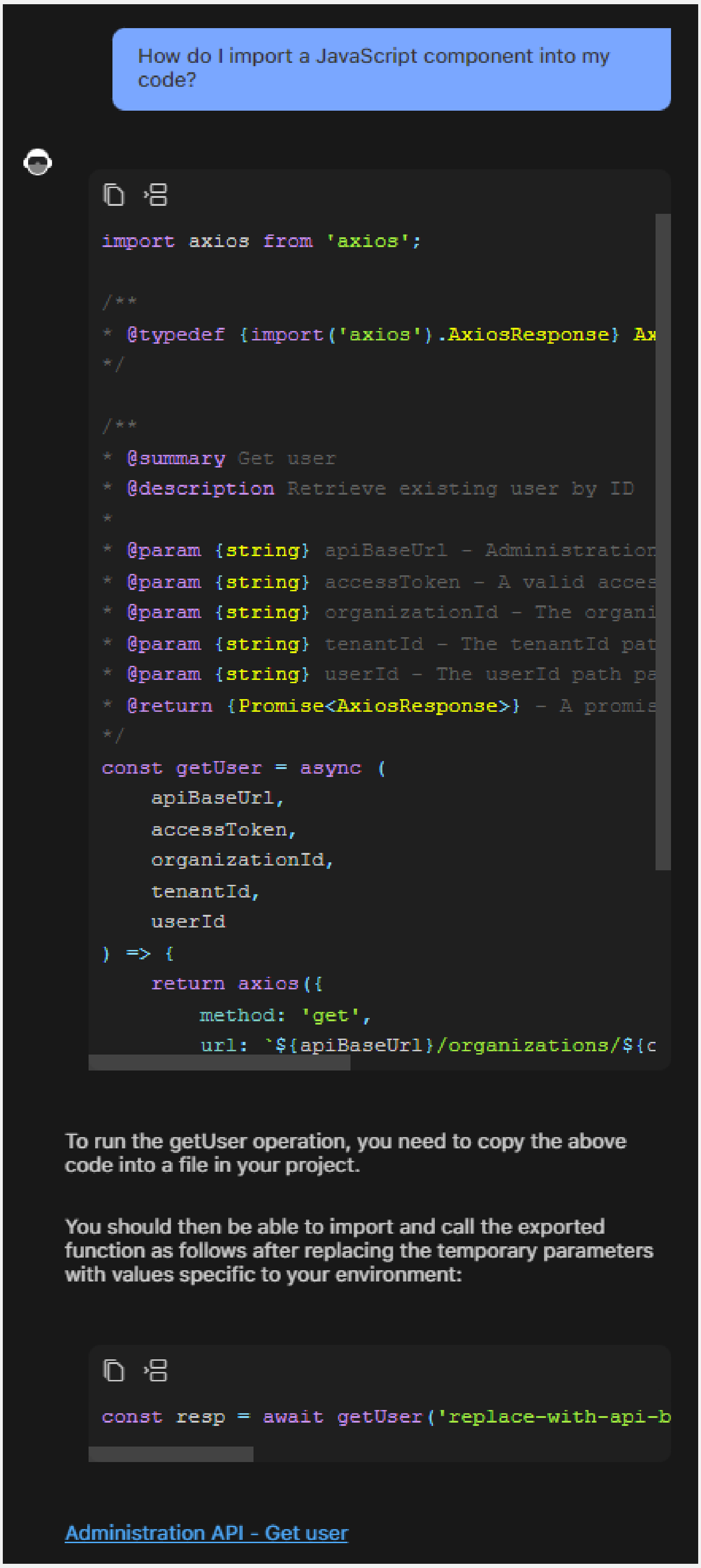
You should then be able to import and call the exported function as follows after replacing the temporary parameters with values specific to your environment:

```
const resp = await listModels('replace-with-api-ba
```

Workflow REST API - Get non-executable process models

Question	How do I create a decision table?
Our test answer	 How do I create a decision table? To create a decision table, you can use the decision table modeler. The decision table modeler user interface is divided into the model properties, hit policy and the decision table. [IM Services | Developer | OpenText]
Question	How do I deploy the application to the tenants in my organization?
Our test answer	 How do I deploy the application to the tenants in my organization? To deploy an application to the tenants in your organization, you need to first provision a subscriber sandbox. This will create a tenant that provides full isolation between subscriber data. Only tenant users can be given access to the tenant and its data. You can then provision your application to the newly created subscriber tenant. Users are given access to applications and their data via a tenant with which they authenticate. Multiple applications can be provisioned to the same tenant. You can also set up an external identity provider, which is your application or an IdP your application uses, for tenant users. [JavaScript Sample | Samples and Collections | Developer | OpenText]

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

Question	How do I import a JavaScript component into my code?
Our test answer	

How do I import a JavaScript component into my code?

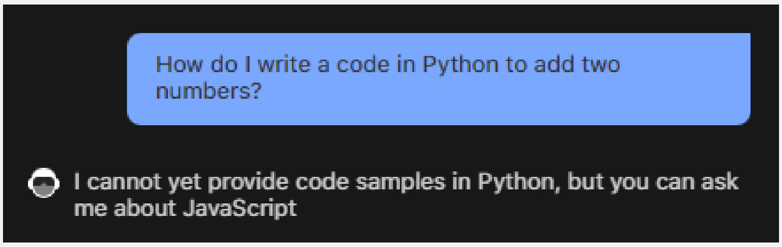
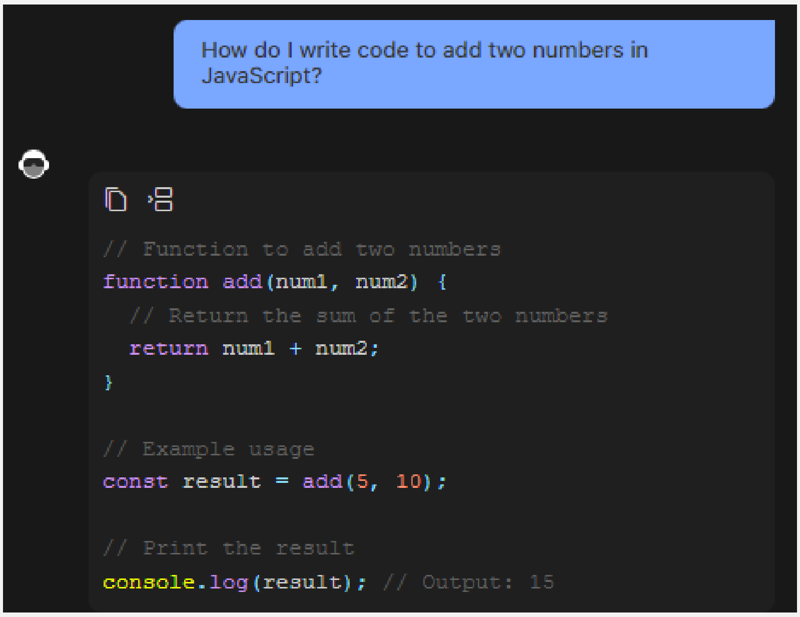
```
import axios from 'axios';

/**
* @typedef {import('axios').AxiosResponse} Ax
*/

/**
* @summary Get user
* @description Retrieve existing user by ID
*
* @param {string} apiBaseUrl - Administration
* @param {string} accessToken - A valid acces
* @param {string} organizationId - The organi
* @param {string} tenantId - The tenantId pat
* @param {string} userId - The userId path pa
* @return {Promise<AxiosResponse>} - A promis
*/
const getUser = async (
    apiBaseUrl,
    accessToken,
    organizationId,
    tenantId,
    userId
) => {
    return axios({
        method: 'get',
        url: `${apiBaseUrl}/organizations/${c
```

To run the getUser operation, you need to copy the above code into a file in your project.

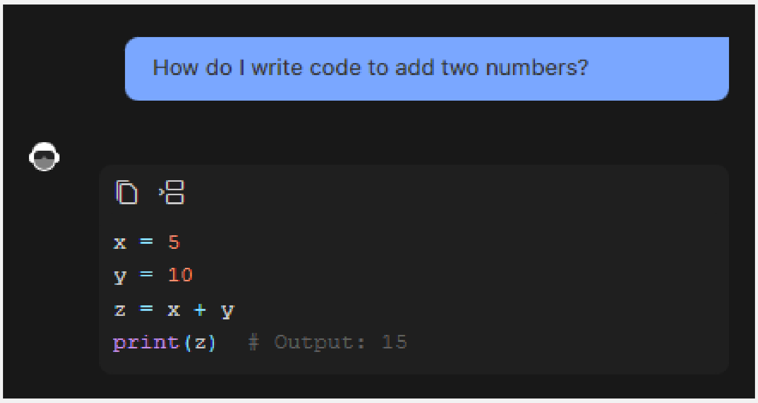
You should then be able to import and call the exported function as follows after replacing the temporary parameters with values specific to your environment:

```
const resp = await getUser('replace-with-api-b
```

Administration API - Get user

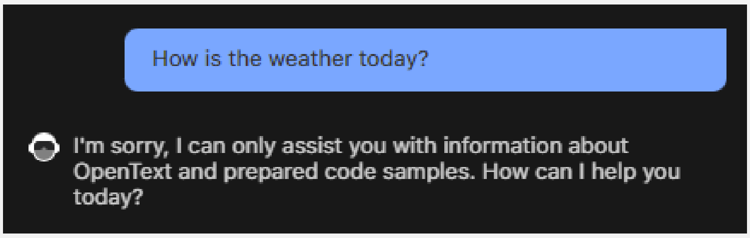
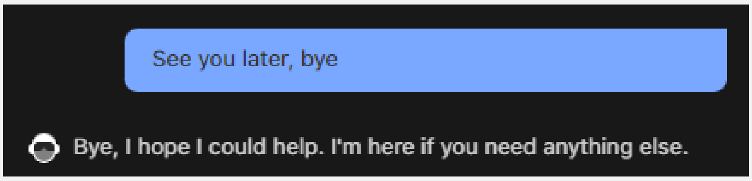
Question	How do I write a code in Python to add two numbers?
Our test answer	 How do I write a code in Python to add two numbers? I cannot yet provide code samples in Python, but you can ask me about JavaScript
Question	How do I write code to add two numbers in JavaScript?
Our test answer	 How do I write code to add two numbers in JavaScript?

```
// Function to add two numbers
function add(num1, num2) {
  // Return the sum of the two numbers
  return num1 + num2;
}

// Example usage
const result = add(5, 10);

// Print the result
console.log(result); // Output: 15
```

Question	How do I write code to add two numbers?
Our test answer	 How do I write code to add two numbers?

```
x = 5
y = 10
z = x + y
print(z)  # Output: 15
```

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

Question	How is the weather today?
Our test answer	 How is the weather today? I'm sorry, I can only assist you with information about OpenText and prepared code samples. How can I help you today?
Question	See you later, bye
Our test answer	 See you later, bye Bye, I hope I could help. I'm here if you need anything else.

Next step:

Building an application with the assistance of Thrust Studio Aviator

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

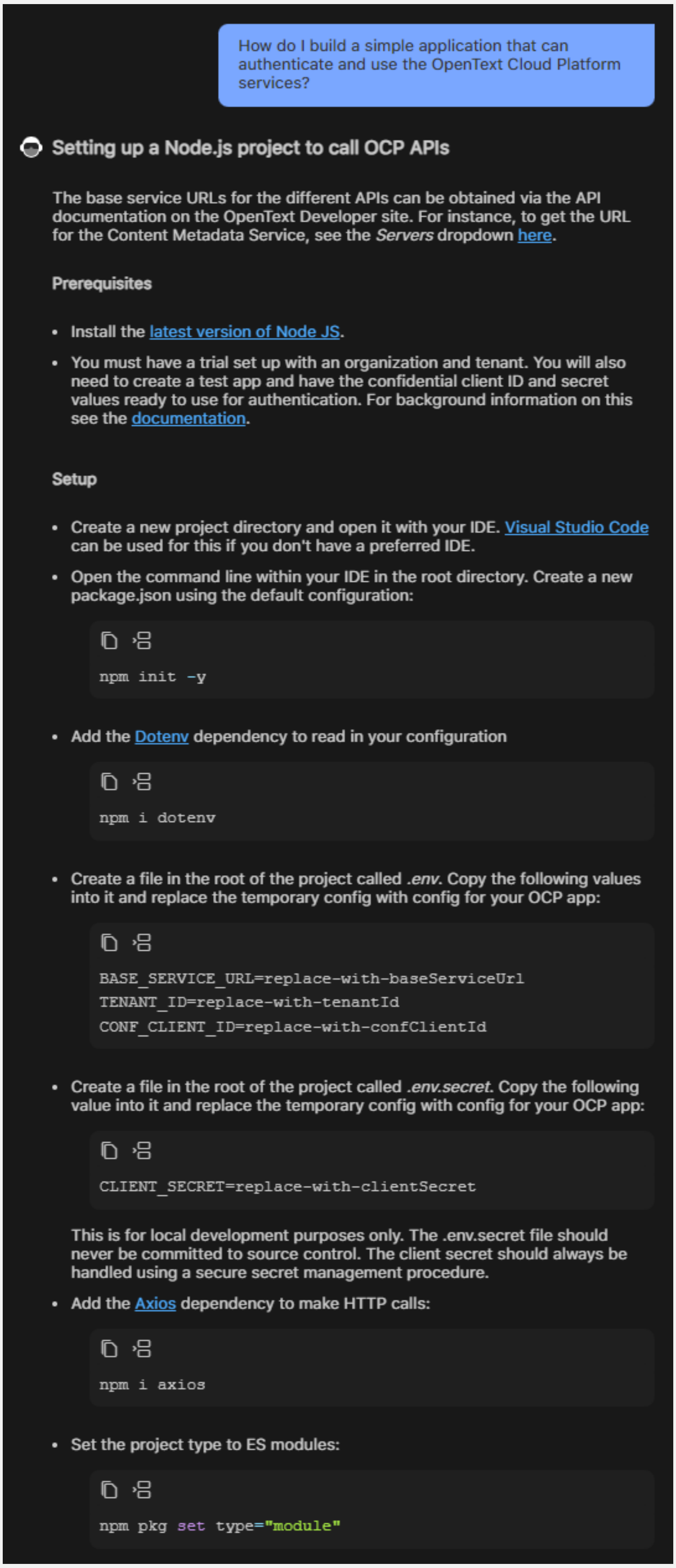
3.3 Building an application with the assistance of Thrust Studio Aviator

Please ask the following questions to the Thrust Studio Aviator. However, differently from the previous section, you will use the Thrust Studio Aviator responses to really build an application that consumes the OpenText™ Thrust Services.

Question	How do I build a simple application that can authenticate and use the OpenText Cloud Platform services?

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.

Our test answer



How do I build a simple application that can authenticate and use the OpenText Cloud Platform services?

Setting up a Node.js project to call OCP APIs

The base service URLs for the different APIs can be obtained via the API documentation on the OpenText Developer site. For instance, to get the URL for the Content Metadata Service, see the *Servers* dropdown here.

Prerequisites

- Install the latest version of Node JS.
- You must have a trial set up with an organization and tenant. You will also need to create a test app and have the confidential client ID and secret values ready to use for authentication. For background information on this see the documentation.

Setup

- Create a new project directory and open it with your IDE. Visual Studio Code can be used for this if you don't have a preferred IDE.
- Open the command line within your IDE in the root directory. Create a new package.json using the default configuration:

```
npm init -y
```

- Add the Dotenv dependency to read in your configuration

```
npm i dotenv
```

- Create a file in the root of the project called *.env*. Copy the following values into it and replace the temporary config with config for your OCP app:

```
BASE_SERVICE_URL=replace-with-baseServiceUrl
TENANT_ID=replace-with-tenantId
CONF_CLIENT_ID=replace-with-confClientId
```

- Create a file in the root of the project called *.env.secret*. Copy the following value into it and replace the temporary config with config for your OCP app:

```
CLIENT_SECRET=replace-with-clientSecret
```

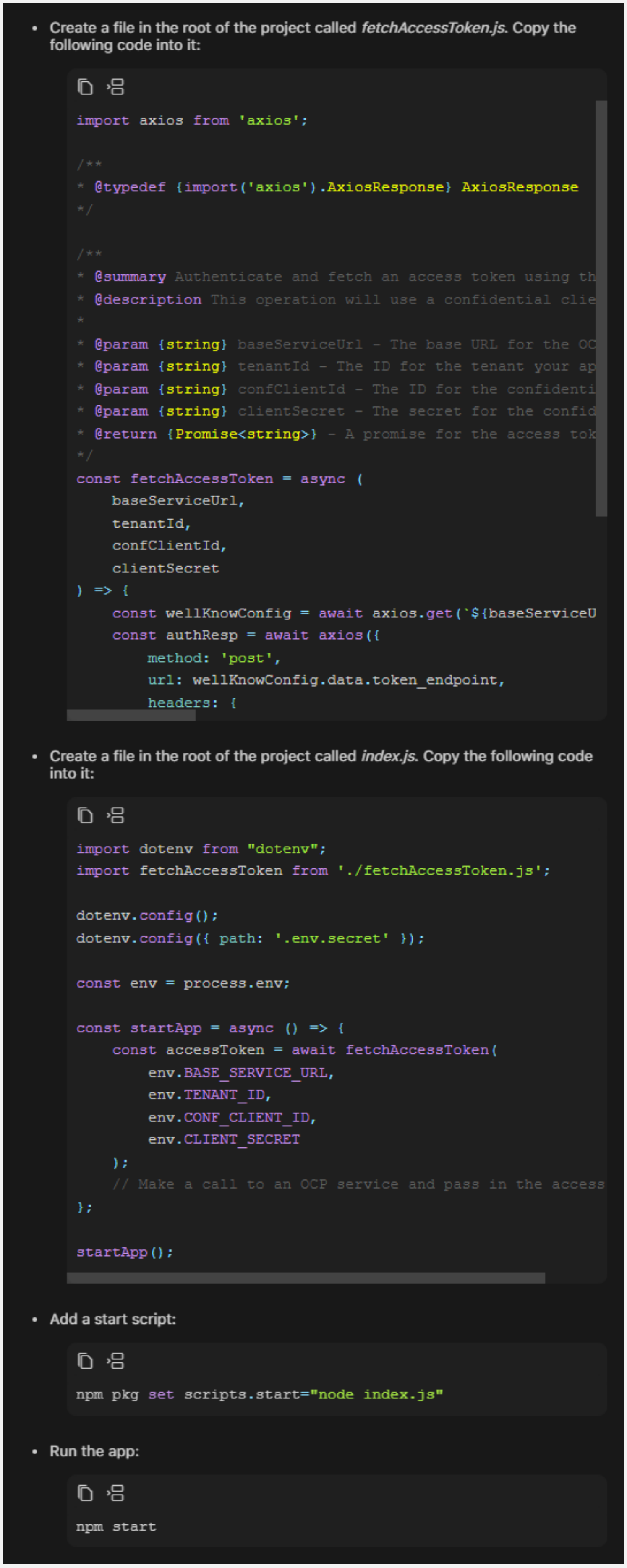
This is for local development purposes only. The .env.secret file should never be committed to source control. The client secret should always be handled using a secure secret management procedure.

- Add the Axios dependency to make HTTP calls:

```
npm i axios
```

- Set the project type to ES modules:

```
npm pkg set type="module"
```



- Create a file in the root of the project called *fetchAccessToken.js*. Copy the following code into it:

```
import axios from 'axios';

/**
* @typedef {import('axios').AxiosResponse} AxiosResponse
*/


/**
* @summary Authenticate and fetch an access token using th
* @description This operation will use a confidential clie
*
* @param {string} baseServiceUrl - The base URL for the OC
* @param {string} tenantId - The ID for the tenant your ap
* @param {string} confClientId - The ID for the confidenti
* @param {string} clientSecret - The secret for the confid
* @return {Promise<string>} - A promise for the access tok
*/
const fetchAccessToken = async (
    baseServiceUrl,
    tenantId,
    confClientId,
    clientSecret
) => {
    const wellKnowConfig = await axios.get(`${baseServiceU
    const authResp = await axios({
        method: 'post',
        url: wellKnowConfig.data.token_endpoint,
        headers: {
```

- Create a file in the root of the project called *index.js*. Copy the following code into it:

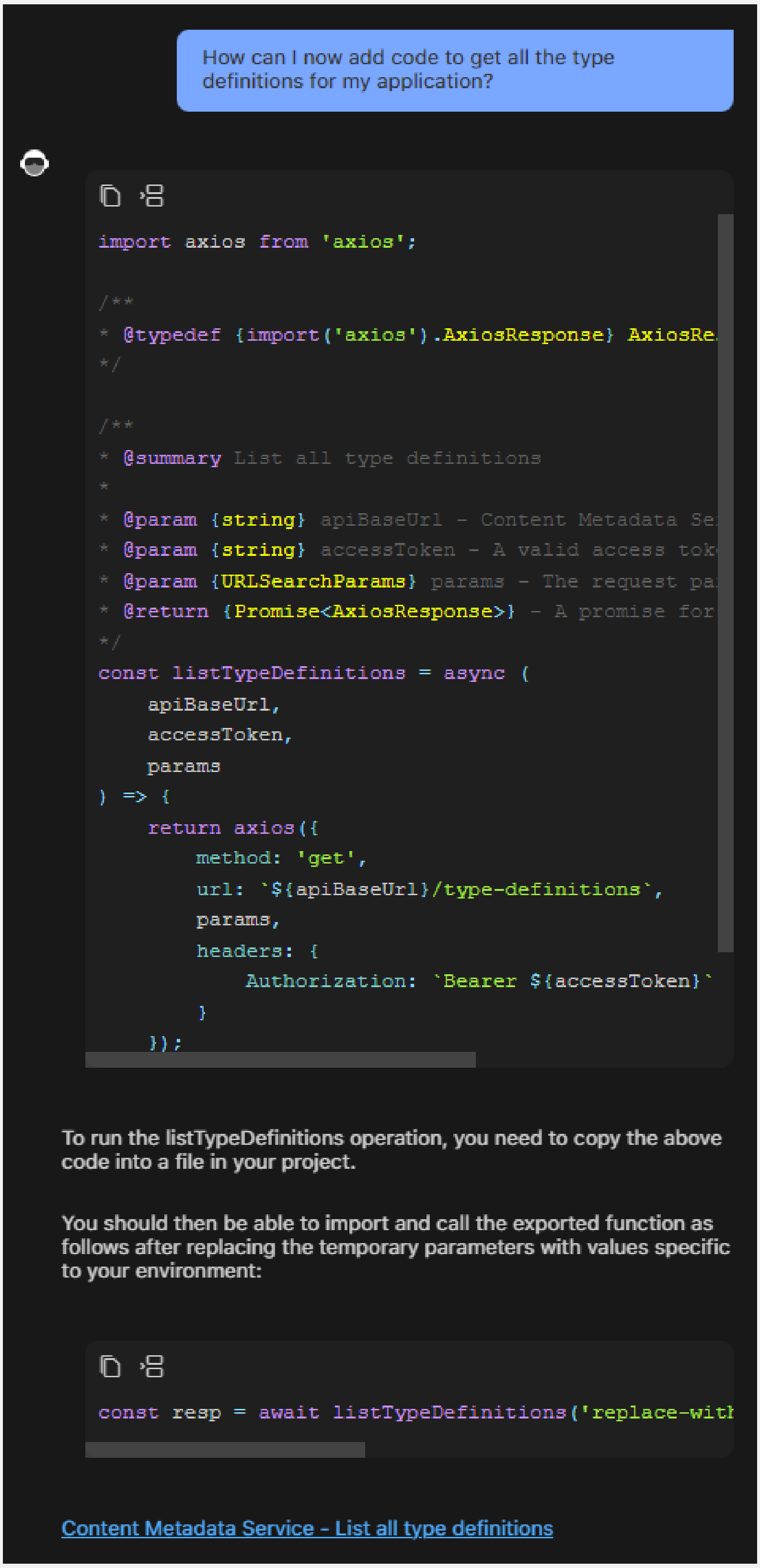
```
import dotenv from "dotenv";
import fetchAccessToken from './fetchAccessToken.js';

dotenv.config();
dotenv.config({ path: '.env.secret' });

const env = process.env;

const startApp = async () => {
    const accessToken = await fetchAccessToken(
        env.BASE_SERVICE_URL,
        env.TENANT_ID,
        env.CONF_CLIENT_ID,
        env.CLIENT_SECRET
    );
    // Make a call to an OCP service and pass in the access
};

startApp();
```

- Add a start script:

```
npm pkg set scripts.start="node index.js"
```

- Run the app:

```
npm start
```

Question	How can I now add code to get all the type definitions for my application?
Our test answer	

How can I now add code to get all the type definitions for my application?

```
import axios from 'axios';

/**
* @typedef {import('axios').AxiosResponse} AxiosRe
*/

/**
* @summary List all type definitions
*
* @param {string} apiBaseUrl - Content Metadata Se
* @param {string} accessToken - A valid access tok
* @param {URLSearchParams} params - The request pa
* @return {Promise<AxiosResponse>} - A promise for
*/
const listTypeDefinitions = async (
    apiBaseUrl,
    accessToken,
    params
) => {
    return axios({
        method: 'get',
        url: `${apiBaseUrl}/type-definitions`,
        params,
        headers: {
            Authorization: `Bearer ${accessToken}`
        }
    });
```

To run the listTypeDefinitions operation, you need to copy the above code into a file in your project.

You should then be able to import and call the exported function as follows after replacing the temporary parameters with values specific to your environment:

```
const resp = await listTypeDefinitions('replace-wit
```

Content Metadata Service - List all type definitions

About OpenText

OpenText enables the digital world, creating a better way for organizations to work with information, on-premises or in the cloud. For more information about OpenText (NASDAQ/TSX: OTEX), visit opentext.com.

Connect with us:

OpenText CEO Mark Barrenechea’s blog

Twitter | LinkedIn

Copyright © 2024 Open Text. All rights reserved. Trademarks owned by Open Text.