SAP BTP - CAP RAG - AI Workshop
Hands-on Guide
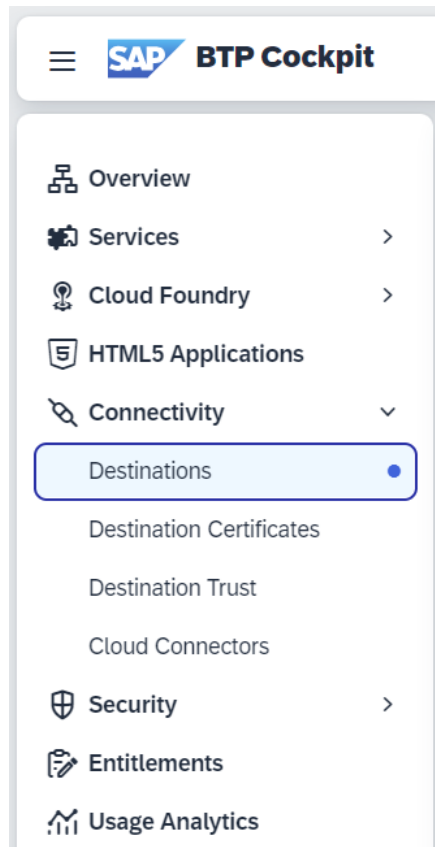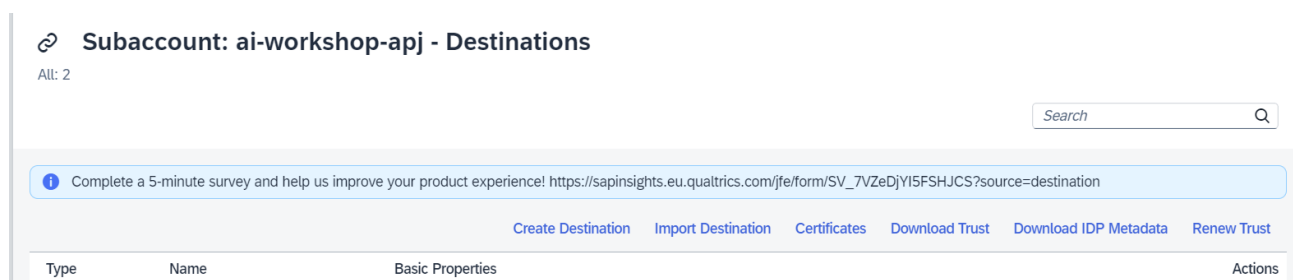
Contents

Create AI Core Destination

SAP BTP Destinations: https://help.sap.com/docs/connectivity/sap-btp-connectivity-cf/managing-destinations

In the SAP BTP Cockpit, navigate to **Destinations**.

Click on **Create Destination**.

Create a new destination with the properties below. The placeholders (<serviceurls.AI_API_URL>, <clientid>, <clientsecret>, <url>) must be replaced with the values from the **AI Core credentials** below-mentioned:

Name=GENERATIVE_AI_HUB

Type=HTTP

URL=<serviceurls.AI_API_URL>/V2

ProxyType=Internet

Authentication=OAuth2ClientCredentials

clientId=<clientid>

clientSecret=<cliensecret>

tokenServiceURL=<url>/oauth/token

tokenServiceURLType=Dedicated

HTML5.Timeout=300000

HTML5.DynamicDestination=true

The information required to create the destination will be find in the AI Core's Service Key instance.

In the SAP BTP Cockpit, navigate to **Services** > **Instances and Subscriptions**.
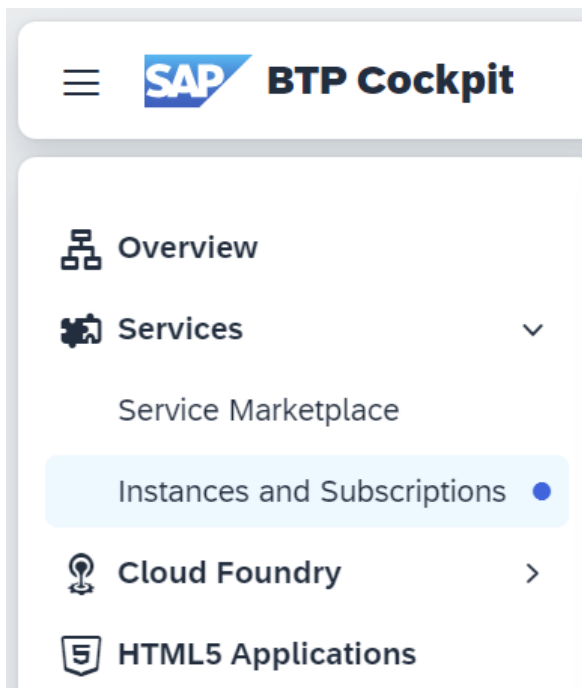
On **Instances tab**, click on the AI Core service instance (**default_aicore** in the image).

Click on **View Credentials** at the top, or, on the **Service Key Name** to open a dialog with the information required in the **Destination**.

default_aicore

Instance ID: **75459d4f-2c2d-4712-b3b6-abded177d9f5**

Service: SAP AI Core
Service Technical Name: **aicore**
Status: **Created**

Plan Display Name: **Extended**
Plan: **extended**
Runtime Environment: **Cloud Foundry**
Scope: dev

Created On: **20 Aug 2024**
Changed On: **20 Aug 2024**

View Credentials ...

Bound Applications (0)   Service Keys (1)   Labels (0)

Create

Name	Created On	Modified On	Status
No bound applications.			

Service Keys (1)

Create

Name	Created On	Modified On	Status	
default_aicore-service-key	26 Aug 2024	26 Aug 2024	Created	...

Credentials

default_aicore-service-key ⌄

Form | **JSON**

```
1  {
2      "clientid": "sb-75459d4f-2c2d-█████████████████████",
3      "clientsecret": "6058f16f-de36-4c5e-9e4e
       █████████████████████████",
4      "url": "https://████████authentication.ap10.hana.ondemand.com",
5      "identityzone": "███████████",
6      "identityzoneid": "b6b20812-███████████████",
7      "appname": "75459d4f-2c2d████████████████████████████",
8      "serviceurls": {
9          "AI_API_URL": "https://api.ai.prod.ap-southeast-2.aws.ml.hana.ondemand.com"
10     }
11 }
```

Copy JSON   Download   Close

Configure LLM for Usage in AI Launchpad

We are going to create two configurations, one for a chat model and another for embedding model.

In the SAP BTP Cockpit, navigate to **Services** > **Instances and Subscriptions**.
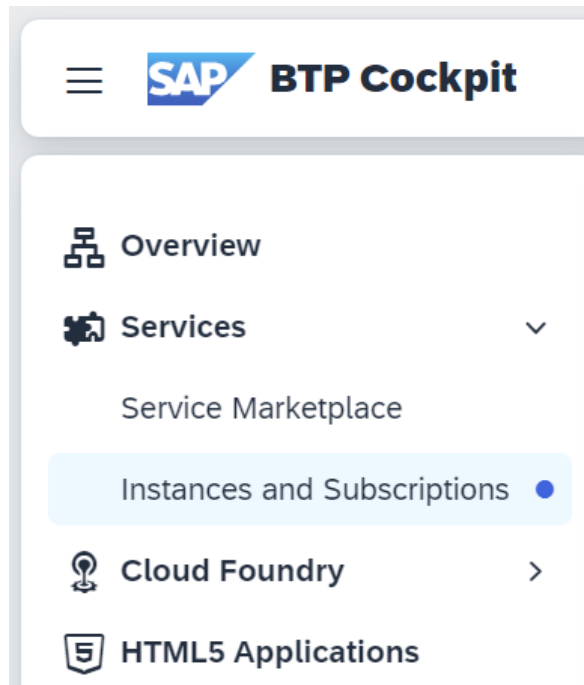
In the **Subscriptions tab**, click on **SAP AI Launchpad**. **SAP AI Launchpad** will open in a new tab, navigate to it.

Select the **Resource Group** (default) created before.
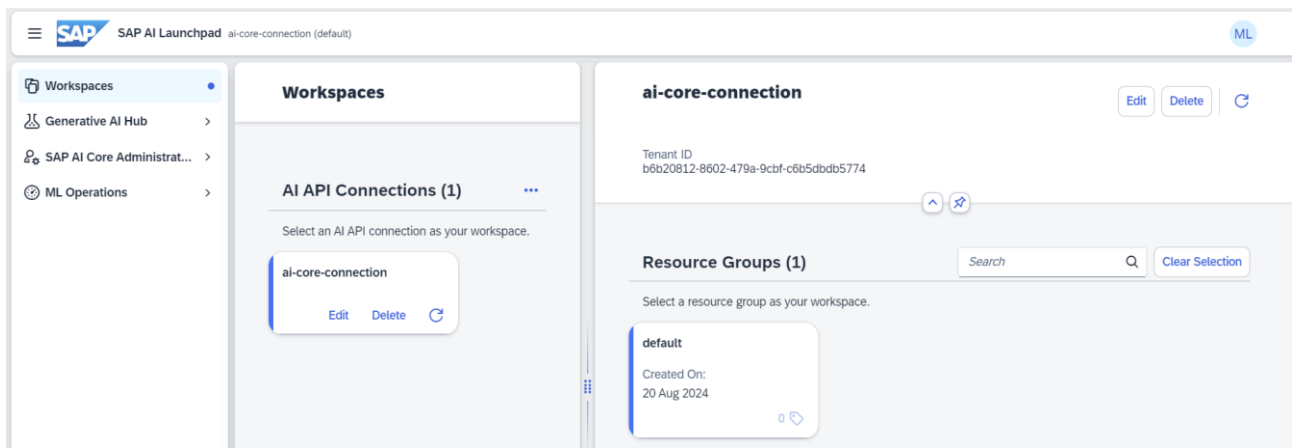
Expand the **ML Operations** menu and click on **Configurations**.,
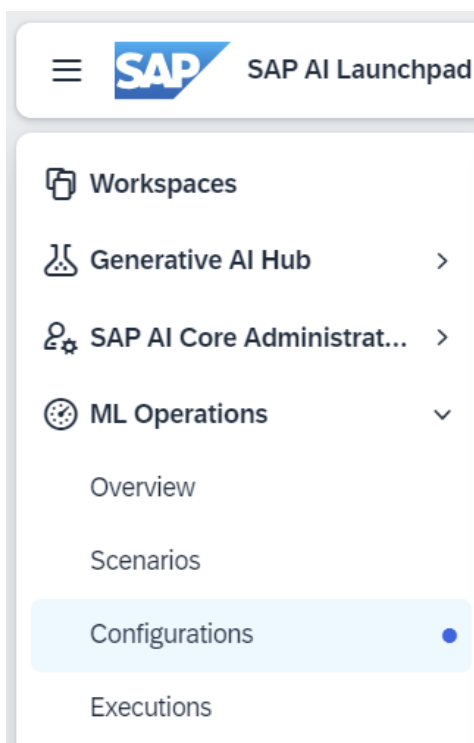
Click on **Create button**.
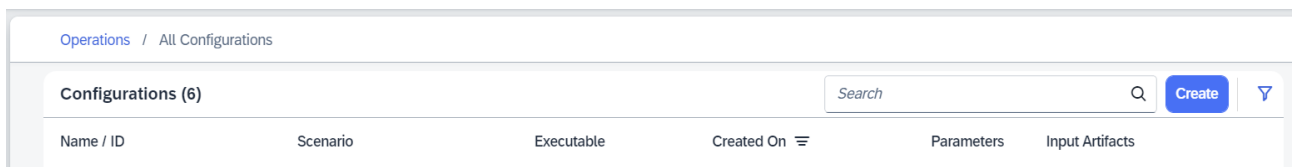
Enter a **Configuration Name** (gpt-4o), select the **Scenario** (foundation-models), select the **Version** (0.0.1), select the **Executable** (azure-openai). Click on **Next button**.

Enter **modelName** (gpt-4o) and **modelVersion** (latest). Click on **Next button**.

Click on **Review button**.

Click on **Create button**.

From the newly create configuration, click on **Create Deployment button**.

Select the **Standard** option for **Duration**. Click on **Review** button.

1 Select Scenario ———— 2 Select Executable ———— 3 Select a configuration ———— **4** **Duration** (Optional) ———— 5 Review

4. Duration

Choose a timeframe for the deployment to be active.

◉ Standard   Use the standard duration which defaults from your environment.

◯ Custom   [ ] [⌄]

Click on **Create button**.

1 Select Scenario ———— 2 Select Executable ———— 3 Select a configuration ———— 4 Duration (Optional) ———— **5** **Review**

5. Review

1.Scenario

Scenario Name: foundation-models   Edit

2.Executable

Executable Name: azure-openai
Scenario Version: 0.0.1   Edit

3.Configuration

Configuration Name: gpt-4o   Edit

4.Duration

Duration: Standard   Edit

Previous   **Create**   Cancel

The **Deployment** will be created, and it'll be available for usage when its status is **RUNNING**. The **Deployment ID** (dc9a9804353026a2) is the key used later by the applications you build.

dc9a9804353026a2

Update   Stop   Delete   ↻

Current Status:	Created On:	Changed On:	Submitted On:	Started On:	Duration:
🕐 RUNNING	today 7:35:23 pm	today 7:37:38 pm	today 7:36:08 pm	today 7:37:13 pm	0 minutes 29 seconds

Target Status:	Running Until:	URL:
🕐 RUNNING	Unlimited	https://api.ai.prod.ap-southeast-2.aws.ml.hana.ondemand.com/v2/inference/deployments/dc9a9804353026a2 🗐

Now, repeat the same **Configuration** and **Deployment** steps to create the **embedding model**.

Operations / All Configurations / Configuration Details

text-embedding-ada-002

Create Deployment

Configuration ID:
cc43bf55-6309-454b-9b3d-fa32344fc146 ⧉

Scenario:
foundation-models
Version 0.0.1

Created On:
26 Aug 2024, 10:34:57 pm

Executable:
azure-openai

Parameters (2)		Search
Name	Description	Value
Optional Parameters		
modelVersion		**latest** Default: latest
modelName	supportedModels: gpt-35-turbo, gpt-4o, gpt-4, gpt-35-turbo-16... Show More	**text-embedding-ada-002** Default: gpt-35-turbo

Operations / All Deployments / Deployment Details

dc8d095f102905b1

Update Stop Delete ↻

Current Status:
⏱ RUNNING

Created On:
today 7:40:44 pm

Changed On:
today 7:48:53 pm

Submitted On:
today 7:42:10 pm

Started On:
today 7:43:20 pm

Duration:
5 minutes 51 seconds

Target Status:
⏱ RUNNING

Running Until:
Unlimited

URL:
https://api.ai.prod.ap-southeast-2.aws.ml.hana.ondemand.com/v2/inference/deployments/dc8d095f102905b1 ⧉

Clone Template Project from GitHub

SAP BAS Dev Space: https://help.sap.com/docs/bas/sap-business-application-studio/dev-spaces-in-sap-business-application-studio

In the SAP BTP Cockpit, navigate to **Services** > **Instances and Subscriptions**.
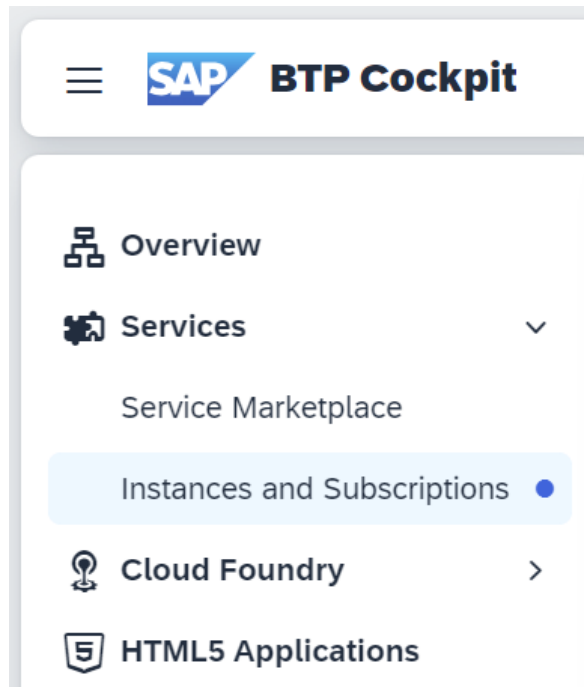
On the **Subscriptions tab**, click on SAP Business Application Studio (BAS) to launch it in a new tab.

If it's the first time you are accessing it, the welcome page will be displayed, and you'll need to create a **Developer Space**. Click on the **Create Dev Space button**. This step is executed one time only.
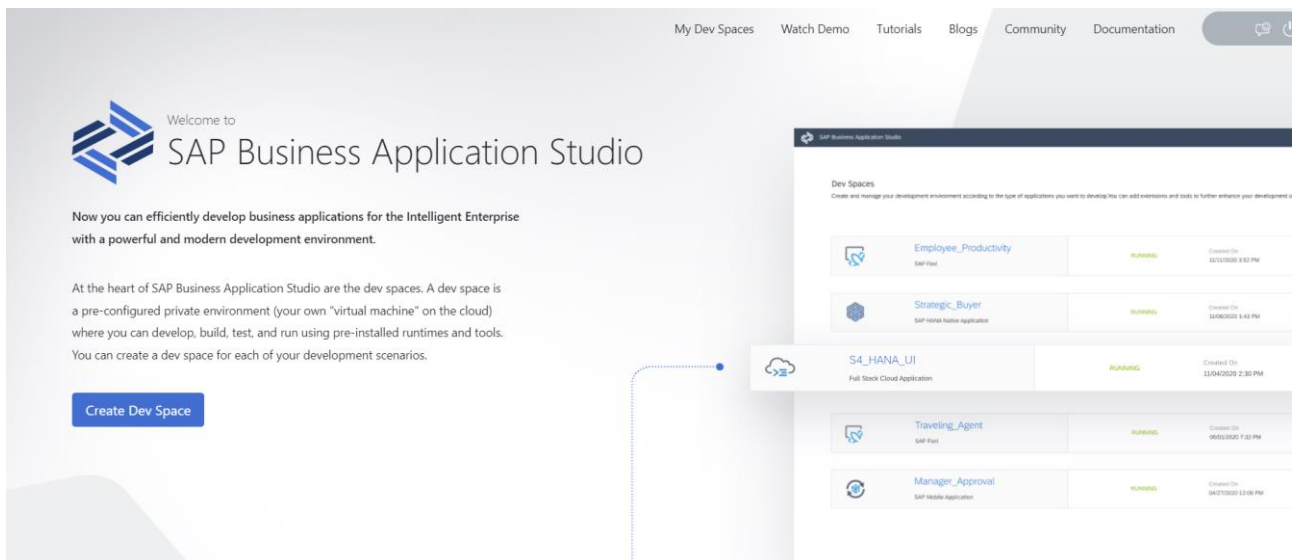
Give the **Dev Space** a **name**, select the **Full Stack Cloud Application** option, and click on **Create Dev Space button**.
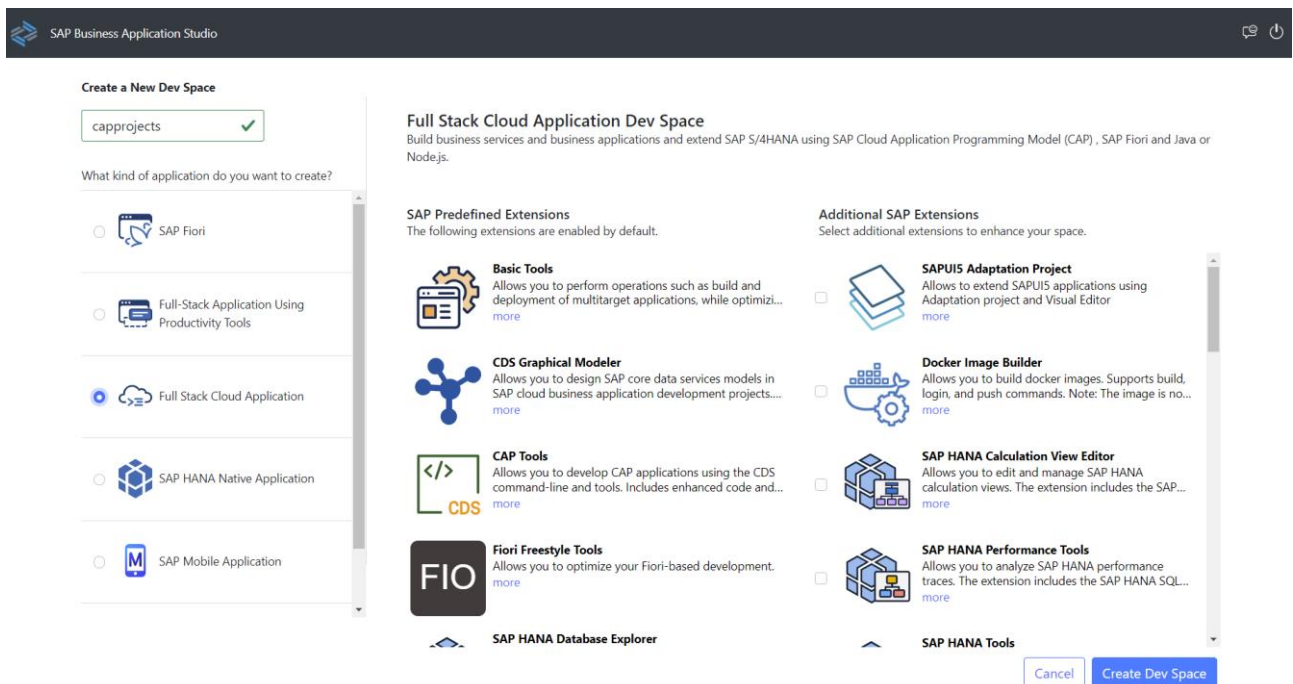
Once the **Dev Space** is created and it's **RUNNING**, click on its name to launch **BAS**.
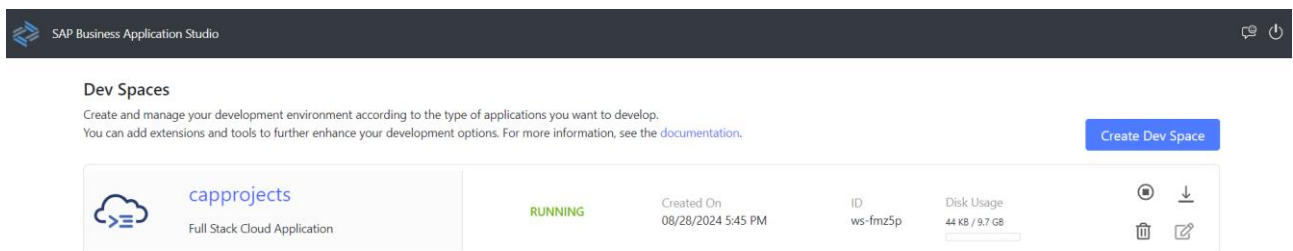
We will not start a project from scratch. We will clone a template project from GitHub.

Click on **Clone from Git**.
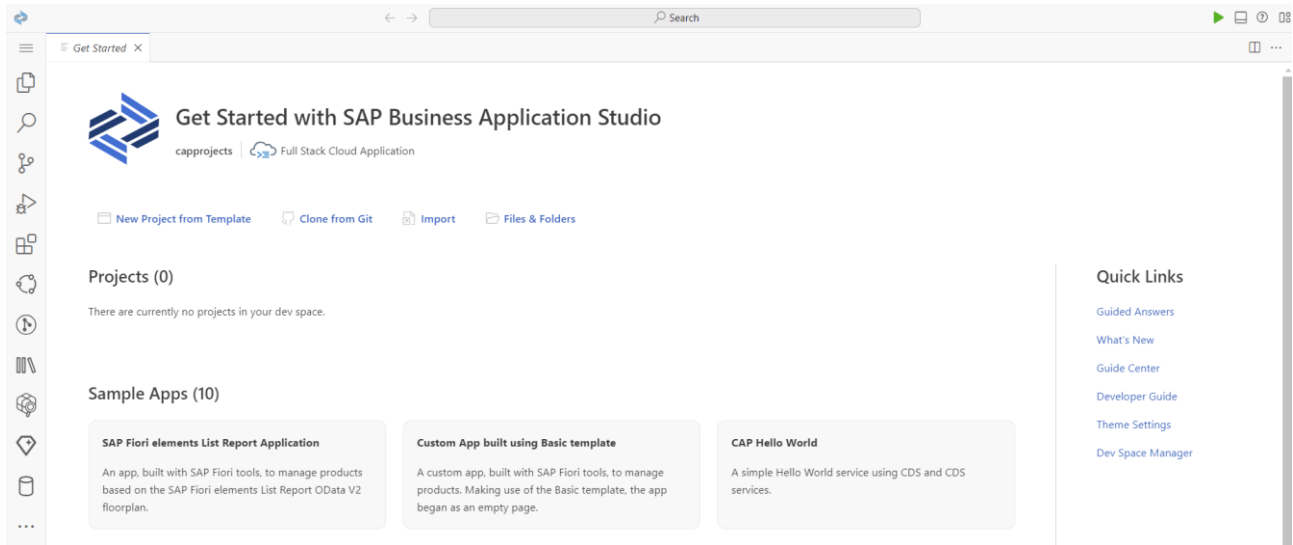
Enter the **git repository** URL (https://github.com/mauriciolauffer/btp-cap-rag-ai-workshop.git) and press **ENTER**.
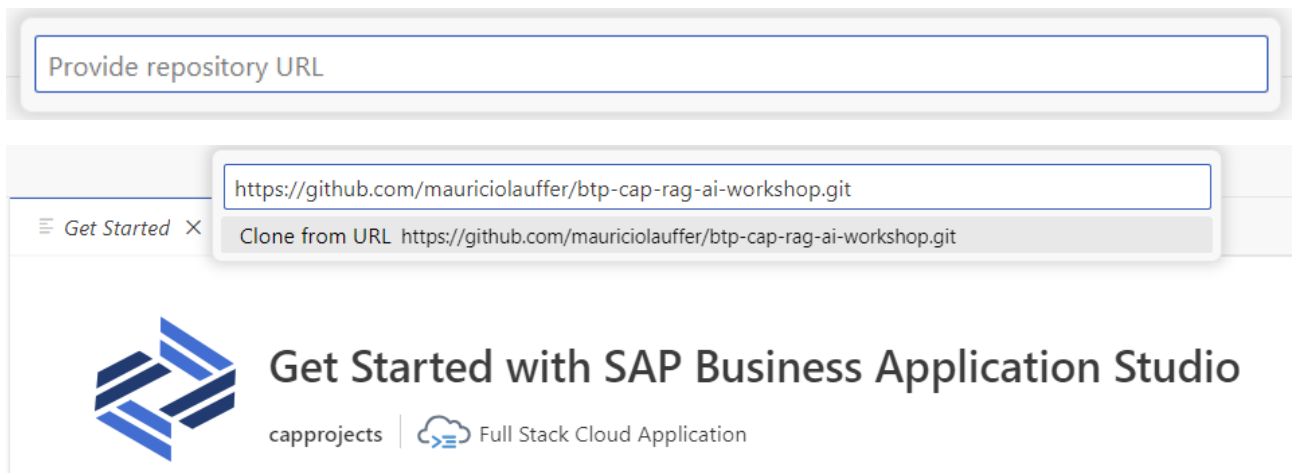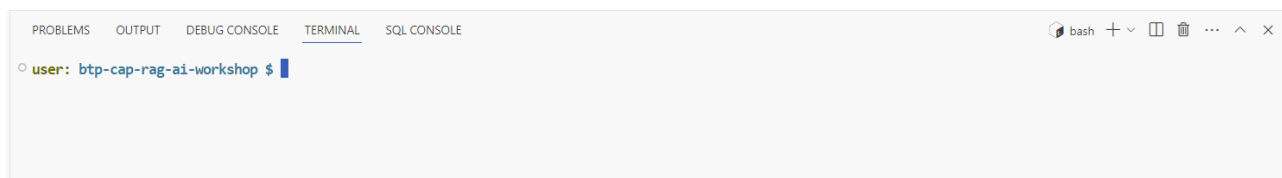
The template project will be cloned and opened. You can start exploring the codebase in the **Explorer** section.
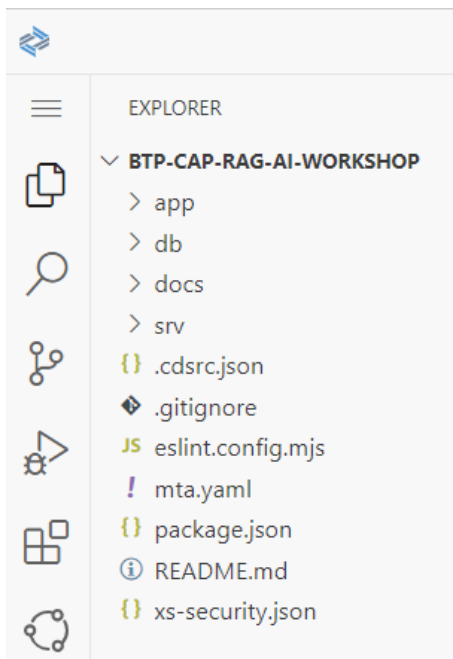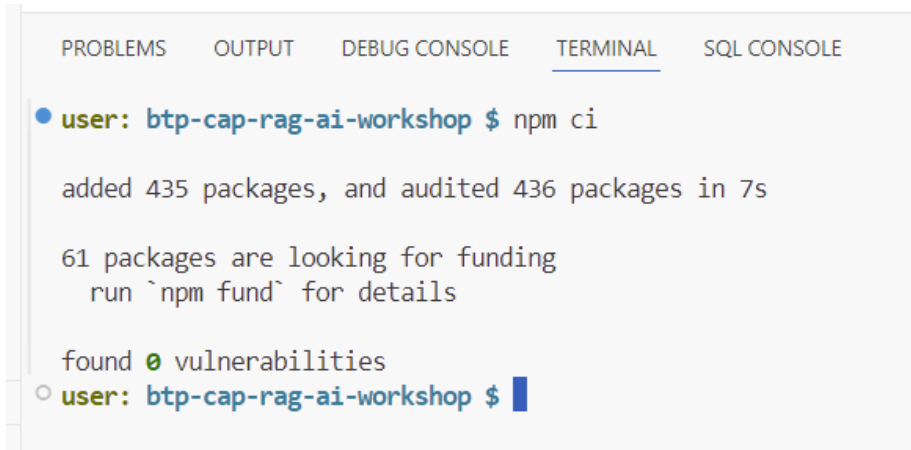
Open the Terminal to install the project dependencies. Click on the **hamburger button** (3 stacked lines icon), then **Terminal** > **New Terminal**.
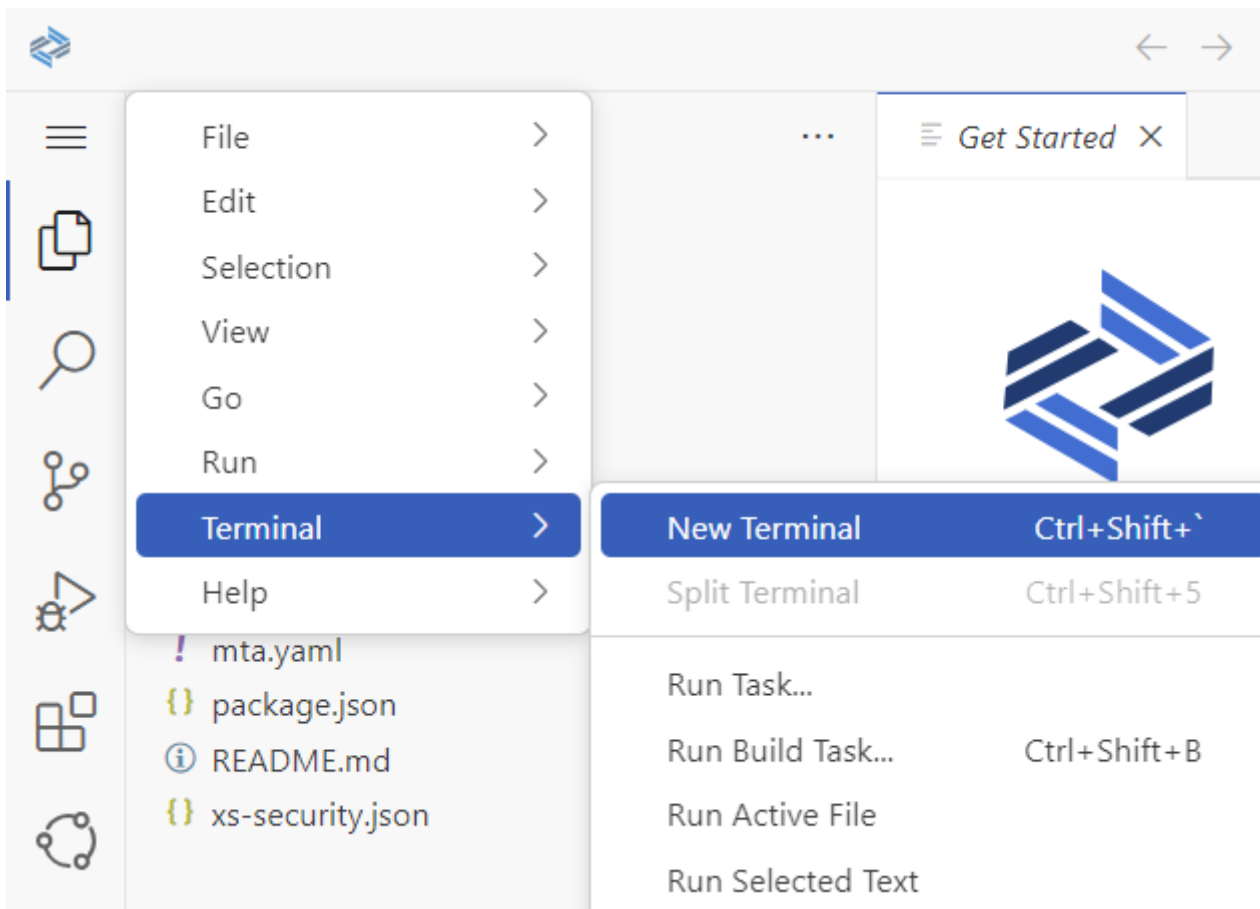
The Terminal should open at the bottom of the screen, install the project dependencies with the command
$ npm ci

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   SQL CONSOLE   bash + ∨ ⬚ 🗑 ⋯ ∧ ✕

user: btp-cap-rag-ai-workshop $ ▊

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   SQL CONSOLE

● **user: btp-cap-rag-ai-workshop $** npm ci

  added 435 packages, and audited 436 packages in 7s

  61 packages are looking for funding
    run `npm fund` for details

  found **0** vulnerabilities
○ **user: btp-cap-rag-ai-workshop $** ▊

---

user: btp-cap-rag-ai-workshop $ ▊

Update AI Core Access Details in the Codebase

Open the file **.cdsrc.json** which contains CAP related configuration. The **genAI Hub** access details are also there under **GENERATIVE_AI_HUB** key.

Replace the placeholder ___PLACEHOLDER_CHAT___ with the **Deployment ID** from the **chat model** in your **AI Launchpad Service**.

Replace the placeholder ___PLACEHOLDER_EMBED___ with the **Deployment ID** from the **embedding model** in your **AI Launchpad Service**.

```
13        "GENERATIVE_AI_HUB": {
14          "CHAT_MODEL_DESTINATION_NAME": "AICoreAzureOpenAIDestination",
15          "CHAT_MODEL_DEPLOYMENT_URL": "/inference/deployments/___PLACEHOLDER_CHAT___",
16          "CHAT_MODEL_RESOURCE_GROUP": "default",
17          "CHAT_MODEL_NAME": "gpt-4o",
18          "CHAT_MODEL_API_VERSION": "2023-05-15",
19          "EMBEDDING_MODEL_DESTINATION_NAME": "AICoreAzureOpenAIDestination",
20          "EMBEDDING_MODEL_DEPLOYMENT_URL": "/inference/deployments/___PLACEHOLDER_EMBED___",
21          "EMBEDDING_MODEL_RESOURCE_GROUP": "default",
22          "EMBEDDING_MODEL_NAME": "text-embedding-ada-002",
23          "EMBEDDING_MODEL_API_VERSION": "2023-05-15"
24        },
25        "AICoreAzureOpenAIDestination": {
26          "kind": "rest",
27          "credentials": {
28            "destination": "GENERATIVE_AI_HUB",
29            "requestTimeout": "300000"
30          }
31        }
```

Create the Database Layer

Open the file **db/schema.cds** and create the tables:

```
namespace btpcapragai;

using {
    cuid,
    managed
} from '@sap/cds/common';

entity Conversation : cuid, managed {
    userId    : String;
    title     : String;
    messages  : Composition of many Message
                    on messages.conversation = $self;
}

entity Message : cuid, managed {
    conversation : Association to Conversation;
    role         : String;
    content      : LargeString;
}

entity DocumentChunk : cuid {
    text_chunk      : LargeString;
    metadata_column : LargeString;
    embedding       : Vector(1536);
}

entity Files : cuid, managed {
    @Core.MediaType  : mediaType  @Core.ContentDisposition.Filename: fileName
    content   : LargeBinary;
    @Core.IsMediaType: true
    mediaType : String;
    fileName  : String;
    size      : String;
}
```

Create the Embeddings OData Service

Open the file **srv/embedding-service.cds** and create the OData Service:

```
using {btpcapragai as db} from '../db/schema';
service EmbeddingService {
  entity DocumentChunk as
    projection on db.DocumentChunk
    excluding {
      embedding
    };
  entity Files        as projection on db.Files;
  action deleteEmbeddings() returns String;
}
annotate EmbeddingService with @(requires: 'authenticated-user');
```

Open the file **srv/embedding-service.js** and create the code to handle the service. The OData Action **deleteEmbeddings** and the OData Entity **Files** need to be implemented. However, the **OData Entity Files** will only have a custom handler for the **ON UPDATE** event.

The **OData Entity Files** ON UPDATE event will have to:

- read the filename for the current File ID being updated
- prepare PDF to be split in chunks of text for embedding
- split the PDF in chunks of text
- get the configuration to use the embedding model
- convert the chunks of text to vectors (embedding)
- insert the vectors into the **DocumentChunk** table

The OData Action **deleteEmbeddings** will have to:

- delete data from Files and DocumentChunk tables

The whole implementation can be found and copied from here: https://github.com/mauriciolauffer/btp-cap-rag-ai-workshop/blob/main/srv/embedding-service.js

Create the Chat OData Service

Open the file **srv/chat-service.cds** and create the OData Service:

```cds
using {btpcapragai as db} from '../db/schema';

type RagResponse_AdditionalContents {
    score       : String;
    pageContent : String;
}

type RagResponse {
    role               : String;
    content            : String;
    timestamp          : String;
    additionalContents : array of RagResponse_AdditionalContents;
}

service ChatService {
    entity Conversation as projection on db.Conversation;
    entity Message      as projection on db.Message;
    action getAiResponse(sessionId : String, content : String, timestamp :
Timestamp) returns RagResponse;
    action deleteChatSession(sessionId :
UUID)                                          returns String;
}

annotate ChatService with @(requires: 'authenticated-user');
```

Open the file **srv/chat-service.js** and create the code to handle the service. The OData Actions **getAiResponse** and **deleteChatSession** need to be implemented.

The OData Action **getAiResponse** will have to:

- get the configuration to use the embedding model
- get the configuration to use the chat model
- handle the chat session/history
- get the RAG response from HANA Vector + LLM
- process the response
- return response to user

The OData Action **deleteChatSession** will have to:

- delete the chat session data

The whole implementation can be found and copied from here: https://github.com/mauriciolauffer/btp-cap-rag-ai-workshop/blob/main/srv/chat-service.js

Build and Deploy Application to SAP BTP Cloud Foundry Runtime

Access the Terminal to execute pre-defined NPM scripts to build and deploy the application to SAP BTP Cloud Foundry Runtime.

Build

In the terminal, execute $ **npm run build**

Log in to Cloud Foundry

Before deploying the application to the **Cloud Foundry Runtime**, you need to log in to the target system. You'll need the target system **CF API Endpoint**. This information can be found in the **SAP BTP Cockpit Overview** page. Click on the **Overview** menu to navigate to it.
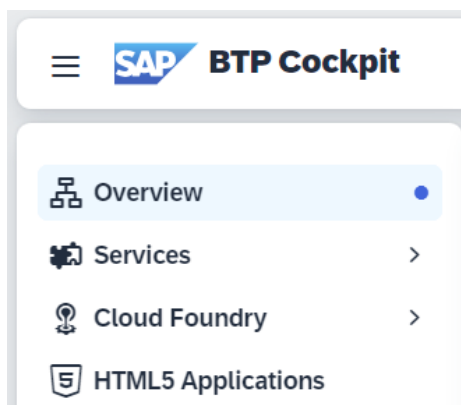
In the Cloud Foundry Environment tab, copy the **API Endpoint** value
(https://api.cf.ap10.hana.ondemand.com). The value will vary based on the subaccount region.

Now, go back to **BAS** to set the **CF API Endpoint** to be used for deployment. In the terminal, execute the command **$ cf api YOUR_API_ENDPOINT_GOES_HERE**

After setting the API Endpoint you can log into the SAP BTP Subaccount Cloud Foundry runtime. In the terminal, execute the command **$ cf login --sso**

This will show you a link to the **Login page**. Click on the link to open it in a new tab.

The **Login page** will be displayed. Log into the **SAP BTP Subaccount** with your credentials.

Copy the **Temporary Authentication Code**.

Go back to **BAS**. In the terminal, paste or type the **Temporary Authentication** Code and press Enter. A list of CF ORGs will be listed, select the one used for the workshop, type its number and press **ENTER**.

```
user: btp-cap-rag-ai-workshop $ cf login --sso
API endpoint: https://api.cf.ap10.hana.ondemand.com/

Temporary Authentication Code ( Get one at https://login.cf.ap10.hana.ondemand.com/passcode ):
Authenticating...
OK


Select an org:
1. academy
2. ai-workshop-apj
3. api-workshop-apj
4. apj-as-team
5. Arrow Energy Pty Ltd_arrow-task-center-dev-2uu2nf58
6. aws11
7. btpcsp
8. fretools-test
9. P&T CEE
10. sandbox
11. SAP CP APJ CustomerSuccessTeam_cspindev
12. SAP CP APJ CustomerSuccessTeam_lfx-dev-tech-academy-x1-ibsoby8d
13. SAP CP APJ CustomerSuccessTeam_recap2024-lqt9ojqs

Org (enter to skip): 2
```

```
Org (enter to skip): 2
Targeted org ai-workshop-apj.

Targeted space dev.

API endpoint:    https://api.cf.ap10.hana.ondemand.com
API version:     3.167.0
user:            mauricio.lauffer@sap.com
org:             ai-workshop-apj
space:           dev
user: btp-cap-rag-ai-workshop $
```

Deploy

Now, execute the command $ **npm run deploy**

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   SQL CONSOLE                                                    bash

user: btp-cap-rag-ai-workshop $ npm run deploy
Uploading application "btp-cap-rag-ai-srv"...
Started async upload of application "btp-cap-rag-ai-srv"
Stopping application "btp-cap-rag-ai-srv"...
Staging application "btp-cap-rag-ai-srv"...
Application "btp-cap-rag-ai-db-deployer" staged
Executing task "deploy" on application "btp-cap-rag-ai-db-deployer"...
Application "btp-cap-rag-ai-srv" staged
Starting application "btp-cap-rag-ai-srv"...
Application "btp-cap-rag-ai-srv" started and available at "ai-workshop-apj-dev-btp-cap-rag-ai-srv.cfapps.ap10.hana.ondemand.com"
Process finished.
Use "cf dmol -i cfb12fbf-6541-11ef-869f-eeee0a96d929" to download the logs of the process.
user: btp-cap-rag-ai-workshop $
```

Use the Embeddings App

Go to **HTML5 Applications** and click on the **capgenairagembedding** link.
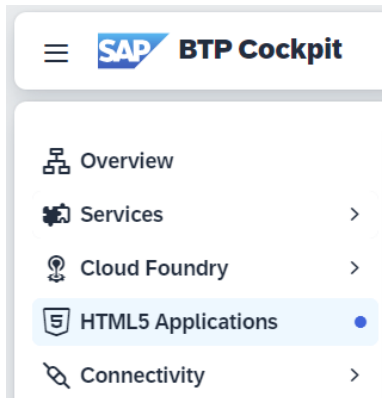
Upload PDF files to convert them into vectors and store in HANA Cloud. This is the embedding process.

Use the Chat App

Go to **HTML5 Applications** and click on the **capgenairagchat** link.
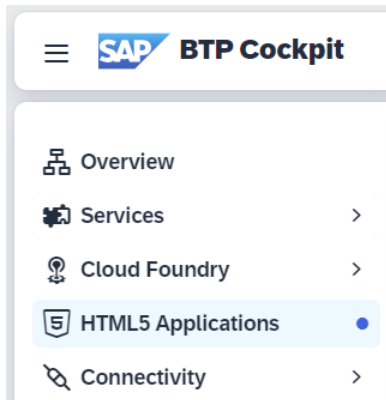
Have a chat with the chatbot. Ask questions regarding the PDF files you have previously uploaded.

Also, ask questions about PDF files you have not embedded yet. The chatbot should say it doesn't know the answer rather than hallucinate.

hi
user · 2024-08-29T15:05:48.764Z

Hello! How can I assist you today?
system · 2024-08-29T15:05:50.413Z

when is my next trip to japan?
user · 2024-08-29T15:05:57.627Z

Your next trip to Japan is on February 13, 2025. You will depart from Cairns at 11:20 AM and arrive at Tokyo Narita at 5:45 PM.
system · 2024-08-29T15:06:02.798Z

Type a message....

hi
user · 2024-08-29T15:05:48.764Z

Hello! How can I assist you today?
system · 2024-08-29T15:05:50.413Z

when is my next trip to japan?
user · 2024-08-29T15:05:57.627Z