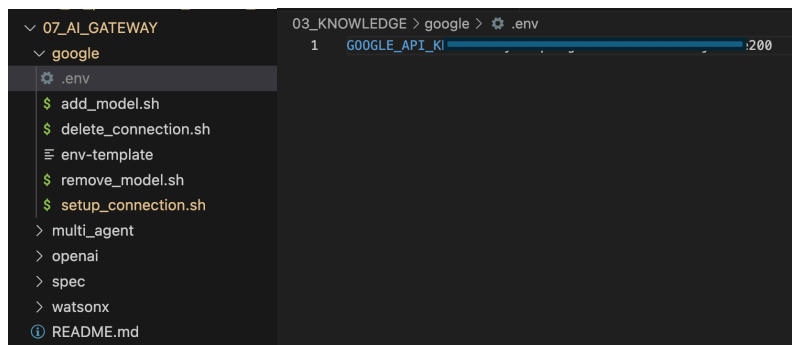


Lab 03_KNOWLEDGE RAG

Add google gemini model

1. Cd in to **03_KNOWLEDGE/google** Folder
2. Add `.env` file in google folder
3. In `.env` file, add the `GOOGLE_API_KEY`
 - To obtain Google API go to this [link](#)



4. Create virtual environment `.venv`

```
python3.11 -m venv .venv
source .venv/bin/activate
pip install ibm-watsonx-orchestrate
```

5. Add virtual environment to Orchestrate instance

```
orchestrate env add -n .venv -u <YOUR_ORCHESTRATE_URL>
orchestrate env activate .venv
```

```
((.venv) → google git:(knowledge) orchestrate env activate .venv ]
[Please enter WXO API key: ]
[WARNING] - Using IBM IAM Auth Type. If this is incorrect please use the '--type' flag to explicitly choose one of 'mcsp', 'mcsp_v1', 'mcsp_v2' or 'cpd'
[INFO] - Environment '.venv' is now active
```

6. Create connection by running `create_connection.bash`

```
bash setup_connection.sh
```

```
[(.venv) → google git:(knowledge) bash setup_connection.sh
[INFO] - Creating connection 'gg_creds'
[INFO] - Successfully created connection 'gg_creds'
[INFO] - Creating configuration for connection 'gg_creds' in the 'draft' environment
[INFO] - Configuration successfully created for 'draft' environment of connection 'gg_creds'.
[INFO] - Creating configuration for connection 'gg_creds' in the 'live' environment
[INFO] - Configuration successfully created for 'live' environment of connection 'gg_creds'.
[INFO] - Setting credentials for environment 'draft' on connection 'gg_creds'
[INFO] - Credentials successfully set for 'draft' environment of connection 'gg_creds'
[INFO] - Setting credentials for environment 'live' on connection 'gg_creds'
[INFO] - Credentials successfully set for 'live' environment of connection 'gg_creds'
```

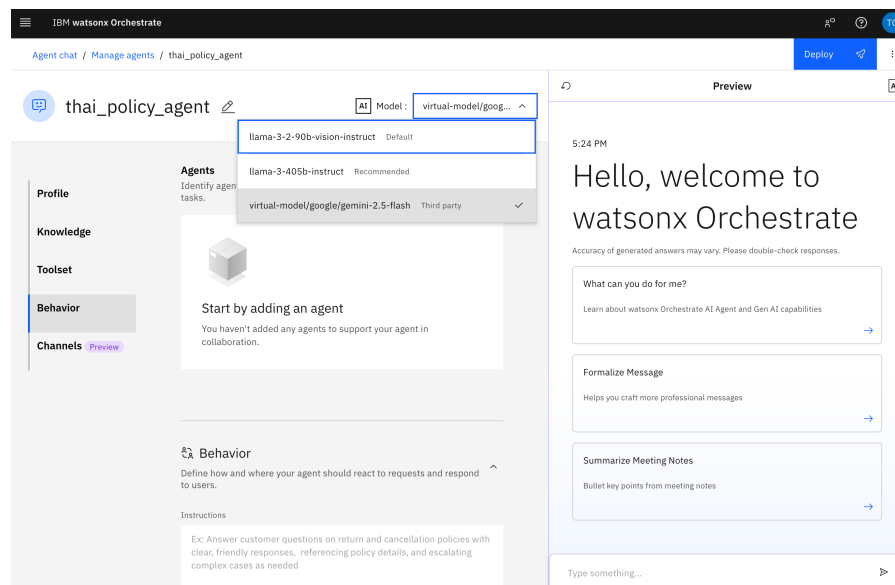
7. Add model using add **add-model.bash**

```
bash add_model.sh
```

```
[(.venv) → google git:(knowledge) bash add_model.sh
[INFO] - Be sure to include the following required fields for provider 'google' in the connection 'gg_creds':
- api_key

[INFO] - Successfully added the model 'virtual-model/google/gemini-2.5-flash'
```

8. After successfully add the connection, the model will show as below. Choose gemini model



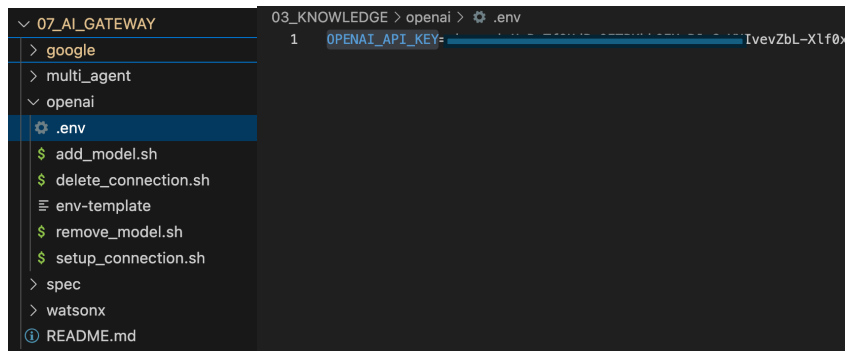
Add Openai model

9. Cd in to **03_KNOWLEDGE/openai** Folder

10. Add .env file in google folder

11. In .env file, add the OPENAI_API_KEY

- To obtain OpenAI API go to this [link](#)



12. Activated virtual env you hae created

orchestrate env activate .venv

```
[(.venv) → google git:(knowledge) orchestrate env activate .venv ]
[Please enter WXO API key: ]
[WARNING] - Using IBM IAM Auth Type. If this is incorrect please use the '--type' flag to explicitly choose one of 'mcsp', 'mcsp_v1', 'mcsp_v2' or 'cpd'
[INFO] - Environment '.venv' is now active
```

13. Create connection by running **create_connection.bash**

bash setup_connection.sh

```
[(.venv) → openai git:(knowledge) * bash setup_connection.sh
[INFO] - Creating connection 'openai_creds'
[INFO] - Successfully created connection 'openai_creds'
[INFO] - Creating configuration for connection 'openai_creds' in the 'draft' environment
[INFO] - Configuration successfully created for 'draft' environment of connection 'openai_creds'.
[INFO] - Creating configuration for connection 'openai_creds' in the 'live' environment
[INFO] - Configuration successfully created for 'live' environment of connection 'openai_creds'.
[INFO] - Setting credentials for environment 'draft' on connection 'openai_creds'
[INFO] - Credentials successfully set for 'draft' environment of connection 'openai_creds'
[INFO] - Setting credentials for environment 'live' on connection 'openai_creds'
[INFO] - Credentials successfully set for 'live' environment of connection 'openai_creds'
```

14. Add model using add **add-model.bash**

bash add_model.sh

```
[(.venv) → openai git:(knowledge) * bash add_model.sh
[INFO] - Be sure to include the following required fields for provider 'openai' in the connection 'openai_creds':
- api_key
[INFO] - Successfully added the model 'virtual-model/openai/text-embedding-3-small'
[INFO] - Be sure to include the following required fields for provider 'openai' in the connection 'openai_creds':
- api_key
[INFO] - Successfully added the model 'virtual-model/openai/gpt-4o'
```

15. After successfully add the connection, the model will show as below. Choose openai model

The screenshot displays the IBM watsonx Orchestrate web interface. The top navigation bar shows the logo and the text "IBM watsonx Orchestrate". Below the navigation bar, the breadcrumb trail indicates the current location: "virtual-model/openai/gpt-4o" > "AI agents" > "thai_policy_agent_openaiemb".

The main content area is divided into two sections. On the left, the "Profile" tab is active, showing a sidebar with options: Profile, Knowledge, Toolset, Behavior, and Channels (with a "Preview" link). The "Profile" section includes a "Description*" field with the text "You are an helpful agent that has k questions related to them." and a "What makes a good description?" link.

On the right, the "Model" dropdown menu is open, showing a list of available models:

- llama-3-2-90b-vision-instruct (Default)
- llama-3-405b-instruct (Recommended)
- virtual-model/google/gemini-2.5-flash (Third party)
- virtual-model/openai/gpt-4o (Third party)** (Selected, indicated by a checkmark)
- virtual-model/openai/text-embedding-3... (Third party)
- virtual-model/watsonx/meta-llama/Lla... (Third party)

On the far right, the "Preview" section shows a chat interface with the message "Hello, welcome to watsonx Orchestrate" and a timestamp of "1:37 PM". Below the message, there are two sections: "What can you do for me?" and "Formalize Message".