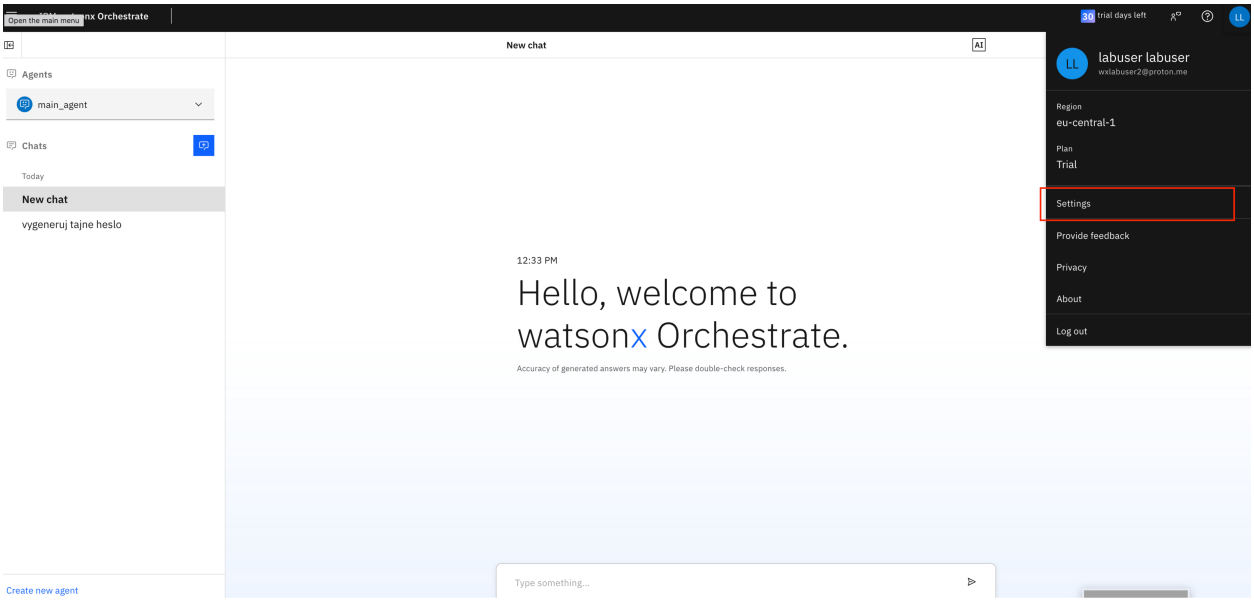
Watsonx Orchestrate ADK lab

We will setup agent acting as financial advisor through Orchestrate ADK.

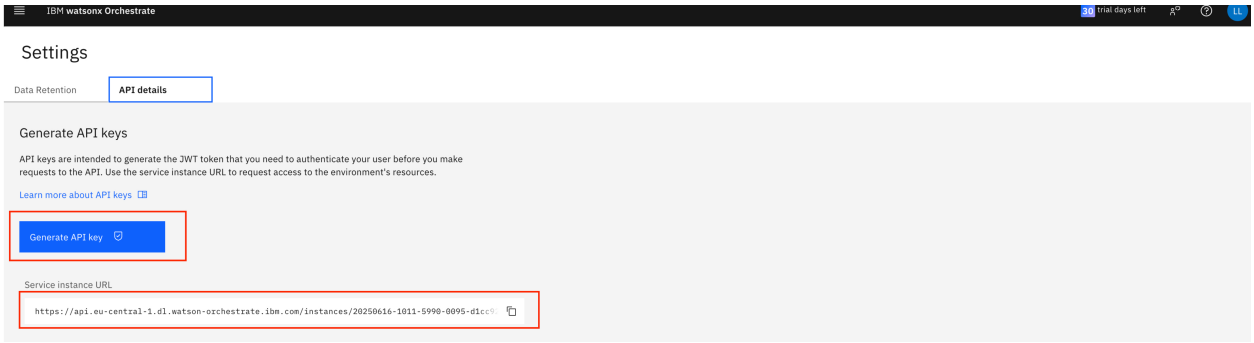
ADK documentation: https://developer.watson-orchestrate.ibm.com

Prerequisites: Get watsonx Orchestrate credentials

To connect to watsonx Orchestrate instance using ADK, it's required to get watsonx Orchestrate **URL** and **API key.**



Step #1



Step #2

1) Local ADK with remote WxO (preferred)

For remote wxO use **URL** and **API key** from instance provisioned before. For other environment provision options, see appendix.

ADK setup

1) You need to install Orchestrate ADK. You can do it with console pip command if you have Python installed (3.11-3.13):
2)

```
pip install ibm-watsonx-orchestrate
orchestrate --version
```

3) Then you need to log-in to remote WxO environment:

```
orchestrate env add -n remote_wxo -u <your_url>
orchestrate env activate remote_wxo --api-key <your_api_key>
```

4) If you did something wrong, delete the environment using following command and start again:

```
orchestrate env remove -n remote_wxo
```

2) Building the agent

1) Available models and agent types

```
orchestrate models list
```

2) Creating tools, agent and its collaborator

To create agents and tools, you can use YAML file (.yaml) or Python file (.py) – in this lab we use Python files, which you can download from here:

https://github.com/IBM-AIBootcamps/NL-watsonx-Orchestrate-workshop/tree/master/ADK_lab/agent

Folder contains these Python files:

- **tool_translate.py** – a Python function calling translation service.
- **tool_mortgage_calculator.py** – a Python function acting as mortgage calculator.
- **tool_wiki_search.py** – a Python function using Wikipedia API to retrieve information.
- **mortgage_advisor.py** – an agent definition in python, acting as mortgage advisor using a mortgage calculator. Changing description can change how the main agent would use this agent as collaborator.
- **main_agent.py** – an agent definition in python, acting as the main agent, which we will be deploying, using Wikipedia search tool and mortgage advisor as collaborator. Changing instructions can change how the agent is behaving.

3) Importing the tools and agents

```
$ orchestrate tools import -k python -f tools/tool_mortgage_calculator.py
$ orchestrate tools import -k python -f tools/tool_wiki_search.py
$ orchestrate agents import -f agents/mortgage_advisor.py
$ orchestrate agents import -f agents/main_agent.py
$ orchestrate tools import -k python -f tools/tool_translate.py -r \
requirements.txt
```

Note: the last command contains *requirements.txt* as a parameter to install required dependencies

4) Verifying deployments

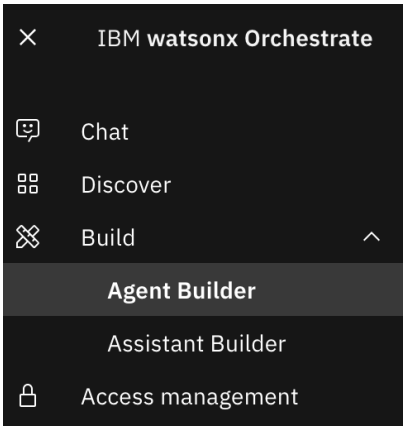
```
orchestrate agents list -v
```

3) Testing the agent

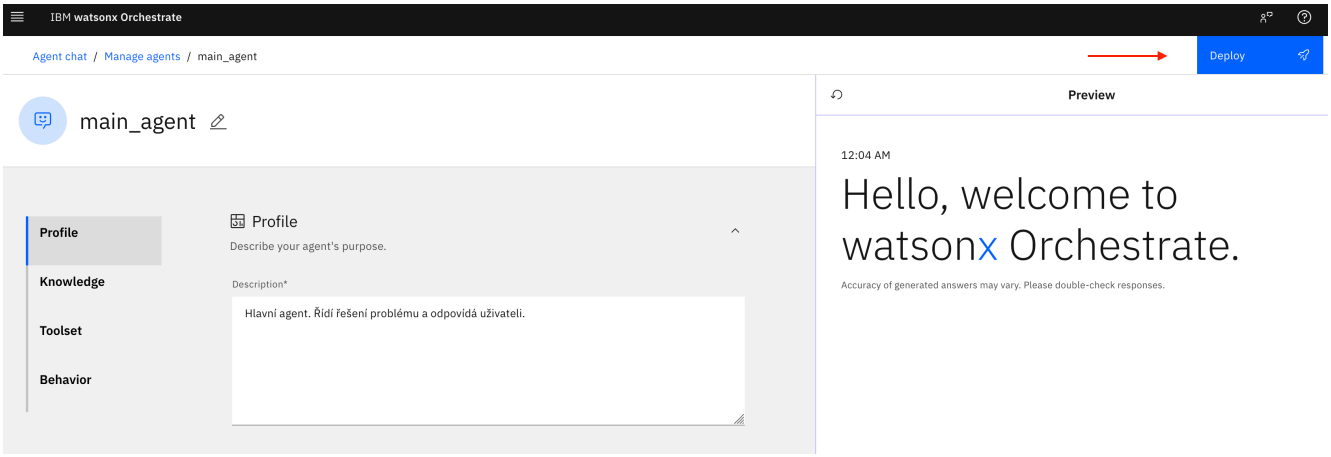
1) Go to the https://dl.watson-orchestrate.ibm.com and log-in with the credentials you provided at the beginning.

2) Open hamburger menu at the left top

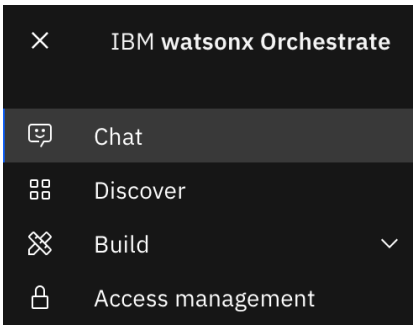
3) Open Agent Builder



4) Select the agent we just imported (*main_agent*) and click Deploy



5) Go back to the chat:



6) Try ask following at the bottom of the chat field:
    a. “Wat is de reporente?”
    b. “Ik wil graag mijn hypotheek berekenen”.

4) Bonus: Import BeeAI agent

1) To import BeeAI researcher agent and to create its native main agent called *native_deep_researcher* run command:

```
orchestrate agents import -f agents/beeai_agent.py
```

2) Then go to step **4) Testing the agent** and replicate steps.

3) Once you finish, try following examples in the chat:
    a. Wie is het staatshoofd van Nederland?
    b. En wie is zijn vrouw*?*

Appendix

2b) Remote environment with ADK and WxO lite

IBM watsonx Orchestrate ADK - US East Only

https://techzone.ibm.com/my/reservations/create/682e3ae02e45f4d814f815d4

Use VM remote desktop. Start Apache Ranger on the desktop and wait till it is finished (right bottom corner). Run batch file on the desktop with Orchestrate init shell script.

2c) Local environment with ADK and WxO lite (docker needed and a key)

1) pip install orchestrate
2) create .env file containing following:

```
DOCKER_IAM_KEY=<docker iam key>
WATSONX_SPACE_ID=<watsonx space id>
WATSONX_APIKEY=<watsonx api key>
OPENSOURCE_REGISTRY_PROXY=us.icr.io/watson-orchestrate-private
WO_DEVELOPER_EDITION_SOURCE=internal
```

3) orchestrate server start --env-file=.env
4) orchestrate env activate local
5) orchestrate chat start