Integration framework – Master Thesis Project – Dominik Merih Aker

Preparations

Before running the project, the chatbot frameworks must be prepared. First, create chatbots as shown in the appendix screenshots of the thesis. You can also request access to the existing chatbots from Dominik Aker (dominik.aker@gmx.com).

If using a SAP CAI chatbot, the config file has to be modified to include some of the chatbot settings:

```
{
    "developerToken":"",
    "authorizationToken":"",
    "botslug":"",
    "version":"",
    "userslug":""
}
```

Both token can be found in the chatbot settings, botslug is the name of the chatbot, and userslug is the name of the chatbot owner. The version is generally “v1” but can be changed to another version. To work with a webchat, the index.html file has to be modified to include the webchat script:

```
<script src="https://cdn.cai.tools.sap/webchat/webchat.js"
    channelId=""
    token=""
    id=""
    ></script>
```

This script with actual values is created automatically when adding a webchat as a channel in the channel settings of an SAP CAI chatbot and can be copy & pasted into the index.html file.

How to run the project

The integration framework can either be started locally or deployed on a cloud platform.

The thesis project is a node.js application. Therefore, node.js has to be installed.

Afterwards, the command *npm install* needs to be run via command line to install all the required packages. Once done, the application can be started by using the command “*node index.js”*.

When used locally, the node.js application will run on port 5000. To allow chatbot frameworks to send requests to a local framework, a tunnel has to be used – one possibility to do so is ngrok. When used, it redirects requests to a ngrok URL to a localhost server. See the ngrok documentation for more details. The https ngrok URL can then be used in the chatbot framework settings – see appendix of the thesis.

When opening the URL http://localhost:5000/ in a browser (while the application is running locally), the index.html is opened. If the preparations were done correctly, a webchat can be found in the bottom right corner of the screen.

When deploying to a cloud platform, the project can be cloned from the git repository. The preparations have to be done using the cloud platform. Then, the cloud URL can be used in the chatbot framework settings.

Example version of the project

A version of the framework that is deployed in the cloud can be found on https://www.aker.tech/

It also contains a short explanation of the different functionalities. The project runs on a DigitalOcean node.js droplet and the chat window is the webchat. It uses the SAP CAI chatbot mentioned in the thesis.

The Amazon Alexa and Google Assistant versions have to be rebuilt based on the screenshots in the appendix of the thesis – they are linked permanently to the authors account and can not be transferred over. To test them, simply rebuild them, and set the https://www.aker.tech/alexa and https://www.aker.tech/google endpoints in the settings. Then test them with any smart device and/or the in-browser test functionality of the frameworks.