Build Document

Date: 30.05.2023

This is a textual writedown of the information of following video: Deployment Video and some additional information.

Before Deployment:

Before you can use the data scraping from PDF files, you need to install TODO: Felix please explain Tesseract and the PATH that needs to be added

Deployment

1. Clone the Github Repo:

```
git clone https://github.com/amosproj/amos2023ss03-qachat
```

2. Set Python Path:

Go to the cloned repository and there set the Python-Path. This will look something like:

Windows:

```
$env:PYTHONPATH += "PATH_TO_THE_REPOSITORY"
```

Unix:

```
export PYTHONPATH=$PYTHONPATH:PATH_TO_THE_REPOSITORY
```

3. Create python environment, activate it & load required dependencies

First create the Virtual Python Enviroment:

Windows:

```
python.exe -m venv venv
```

Unix:

```
python -m venv venv
```

After this activate it with:

Windows:

```
.\venv\Scripts\activate
```

Unix:

```
./venv/bin/activate
```

Now you activated the environment and need to download the required dependencies:

Windows:

```
pip install -r .\requirements.txt
```

Unix:

```
pip install -r ./requirements.txt
```

4. Add the tokens to the project

For the project you need a variety of tokens, to get the project running. These are 3 Slack, 2 Supabase, 1 DeepL, 1 Tessdata and 3 Confluence Tokens. Add them by creating a file called 'tokens.env' in the 'QAChat' Folder. The tokens you either need to get from the services themselves, as described in the associated Wiki Pages, or get them from the Sync.com Website. The exception is the Tessdata "token", which is simply the path to your tessdata directory. The directory can be found within the virtual environment you created for this project.

5. (Optional) - Testing

For starting the implemented test first switch to the 'Testing' folder. Now you can run:

Windows:

```
pytest -s .\qa_agent_unit_test.py
```

Unix:

```
pytest -s ./qa_agent_unit_test.py
```

6. (Optional) - Loading Data into the Supabase Database

This will be automated in the end and after the initial setup, won't be necessary. For completeness of this guide, this still should be mentioned. First Change into the 'QAChat\DataProcessing' folder and there run the main.py file by entering following into your terminal:

Windows:

```
python.exe .\main.py
```

Unix:

```
python ./main.py
```

This can take quite some time, as there are already Machine Learning Models being used here. This will take even more time on the first time you run the program, as some models may need to be downloaded first.

When the process is finished, the database is filled with data, that can then be used for answering the question of users.

7. Starting the Slack-Bot

To start the Slack Bot change to the 'QA_Chat\Slack_Bot' folder.

Then run:

Windows:

```
python.exe .\qa_agent.py
```

Unix:

```
python ./qa_agent.py
```

As soon as you see the message: 'Bolt app is running!' in your terminal, the Slack Bot will react to any message it gets sent on Slack in its message tab. On how to use the Slack Bot on Slack, take a look at the User Documentation