PROJECT DOCUMENTATION

Yakabikaj GUI documentation

ERC-Project | Universität Hamburg
COBHUNI
Contemporary Bioethics and the
History of the Unborn in Islam

COBHUNI - ERC-Project | Universität Hamburg
Asien-Afrika-Institut
Prof. Dr. Thomas Eich

Version 1.0

Status as of 21/12/2021

Revisions

Date	Update	Editor
19/12/2021	Initial creation of the documentation	Sven Längler
21/12/2021	Revision of OS specific information	Florian Jäckel
21/12/2021	Additions to the Developer Notes	Sven Längler

Table of contents

Requirements

Docker

Docker in version v20.0 or higher must be installed on your computer.

More information can be found here: https://docs.docker.com/get-docker/

Docker-Compose

In addition, *Docker-Compose* in version 1.29 or higher is required. Information on installation and setup can be found here: https://docs.docker.com/compose/install/

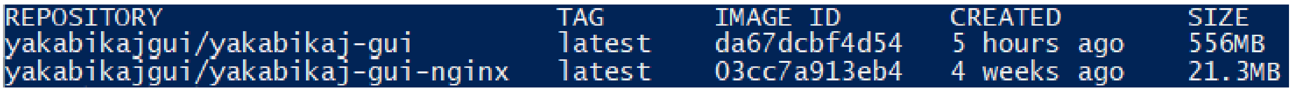
How to start: Installation

Build the Docker image

Unzip the archive containing the project files to any directory on your local storage. Then open your terminal (under Microsoft Windows, you can use PowerShell or command prompt „CMD“; under Mac use the Terminal) and navigate to the path where you saved the unzipped project files (e.g., "cd path/to/yakabikaj/GUI"). Then enter this command into your terminal to build the Docker image:

```
docker-compose build --pull
```

Verification of the build can be done by using the command `docker image ls` to output the list of all available images. The images *yakabikaj-gui-nginx* and *yakabikaj-gui* should be included in this list.



```
REPOSITORY                          TAG       IMAGE ID       CREATED        SIZE
yakabikajgui/yakabikaj-gui          latest    da67dcbf4d54   5 hours ago    556MB
yakabikajgui/yakabikaj-gui-nginx    latest    03cc7a913eb4   4 weeks ago    21.3MB
```

Run the application

Open your terminal (under Microsoft Windows, you can use PowerShell or command prompt „CMD“; under Mac use the Terminal) and navigate to the path where you saved the project files (e.g., "cd path/to/yakabikaj/GUI"). Then enter this command into your terminal to start the built Docker image.

```
docker-compose up
```

Now you can open the search interface in the browser of your choice via the local network interface through port 80 or 8000. To do this, enter one of the following URLs:

http://localhost

http://localhost:8000

http://127.0.0.1

http://127.0.0.1:8000

Updating the application

Update the local project files manually, i.e., save the new zip file and unzip it, and rebuild the Docker image (see Build the Docker image) or pull the latest image from Docker Hub (login with the `docker login` command may be required).

```
docker pull yakabikajgui/yakabikaj-gui:latest
```

User interface

tbd

Notes for developers

Directory structure & project files

File / directory	Description	Belongs to
django	Root folder of Django application	application
yakabikaj	Home folder of the Yakabikaj search algorithm	algorithm
data.py	Required by search.py	algorithm
search.py	query.py uses the export ***search***	algorithm
yakabikaj_gui	Home folder of the Yakabikaj GUI (Django framework)	Django
static	Static files including assets, CSS, fonts and JavaScript files	Django
templates	All HTML view markups	Django
asgi.py	Deployment of ASGI to NGINX	Django
models.py	Must exist due to compatibility reasons (Djano database)	Django
result.py	Handles search queries initiated in GUI	Django
settings.py	Django‘s project settings	Django
urls.py	Django‘s routes configuration	Django
views.py	All Django views	Django
wsgi.py	Deployment of WSGI to NGINX	Django
manage.py	Django’s command-line utility for administrative tasks	Django
nginx	Root folder of NGINX container	NGINX
default.conf	Default configuration of NGINX server	NGINX
Dockerfile	Build NGINX container from *nginx:1.19.0-alpine*	NGINX
.env	Environment settings	Docker
docker-compose.yml	Build the project including Gunicorn and NGINX service	Docker
Dockerfile	Build Yakabikaj GUI container from *python:3.8.5-alpine*	Docker
entrypoint.sh	Entry script that is executed on loading the Yakabikaj GUI container	Docker
requirements.txt	All Python requirements	Docker

Package dependencies

Django (Version 3.0 or higher)

Gunicorn (Version 20.0 or higher)

psycopg2-binary (Version 2.9 or higher)

more-itertools (Version 8.8 or higher)

Editing and extending settings for search algorithm

File django/yakabikaj_gui/query.py

Class *QueryForm* holds all Django form setting fields.
Add Django forms object (e.g. BooleanField) if neccessary.

Also update *get_config()* method of class *Query*. This method maps GUI values against parameters for the search algorithm call.

File django/yakabikaj_gui/templates/form_settings.html

Finally, the control element for the new setting must be rendered on the user interface. To do this, we adjust the file "form_settings.html" and include a setting template from "form_setting.html", for example:

```
{% include „form_setting.html“ with field=form.MY_SETTING label=‘My new setting‘ id=123%}
```

The ***field*** value must be corresponding to the setting created in *QueryForm*.
The description of the setting is passed with the parameter ***label***.
The parameter ***id*** contains an numerical ID, which should allow the identification of the setting option.

Optionally a reference to help content that is provided as Bootstrap modal (see also *django/yakabikaj_gui/templates/help/**) can be included using parameter **help_target**

Additional configuration

Apart from the Django settings in *settings.py*, the application uses some own settings.

MIN_CONTEXT_OFFSET	describes the length of the minimum context (characters) for abbreviated hit output, with content separated only at spaces (default 15).
MARK_MORE_CONTEXT	defines the character string to be prepended or suffixed to indicate abbreviated content in the match output (default '[...]')
HIGHLIGHT_PATTERN	defines the HTML pattern for highlighting the search hits. Curly brackets are to be entered in this order as placeholders for the following texts and values: 1.) for text before highlighting 2.) CSS class(es) for highlighted text 3.) highlighted text 4.) for text after highlighting (default value '{}<span class="{}">{}</span>{}')