This page contains the build and deploy guide of the Digital Identity Software.

Build & Deployment Process

Prerequisites

Docker will be required to build and deploy our application. On ubuntu you can install the required tools with:

```
sudo apt install docker-compose docker git nano
```

Download the latest version and change to the corresponding folder:

```
git clone https://github.com/amosproj/amos2022ss04-digital-identity.git && cd amos2022ss04-d
```

Alternatively you can download the latest release from: https://github.com/amosproj/amos2022ss04-digital-identity/releases/latest/

Configuration

For entering the data regarding our mail relay and the Lissi Agent, edit the docker-compose.yml:

```
nano docker-compose.yml
```

The following lines are needed to be adjusted:

```
    environment:
      MYSQL_HOST: mysql
      HOSTNAME: # <-
      # Mail Relay Configuration:
      MAIL_RELAY_HOST: smtp.gmail.com # <-
      MAIL_RELAY_PORT: 587 # <-
      MAIL_RELAY_USERNAME: myemail@gmail.com # <-
      MAIL_RELAY_PASSWORD: kjahdfklfjhsdk # <-
      MAIL_RELAY_SMTP: "true" # <-
      MAIL_RELAY_STARRTTLS: "true" # <-
      # Lissi API Configuration
      LISSI_API_URL: https://myagent.domain.com # <-
      LISSI_API_AUTH_CLIENT_ID: spring-boot-client # <-
      LISSI_API_AUTH_CLIENT_SECRET: sadfkjhsadkjh5klj1h231l2kjhsasal # <-
```

In the end only the ports of the frontend are exposed. If you want to change them you can do it here:

```
  caddy:
    image: caddy
    restart: unless-stopped
    ports:
      - "80:80" # <- Change the number before the ':'
```

```
      - "443:443" # <- Change the number before the ':'
    volumes:
      - ./caddy/Caddyfile:/etc/caddy/Caddyfile
      - ./src/digitalIdentity-frontend/dist/digital-identity:/srv
      - caddy_data:/data
      - caddy_config:/config
```

Enable https:// (optional)

For that you have to edit the Caddyfile:

```
nano caddy/Caddyfile
```

Change:

```
http:// { # <- replace http:// with your domain.
    encode gzip
    # ...
```

to e.g.

```
www.example.com {
    encode gzip
    # ...
```

Compose and Start

To finally start everything you only need to call docker-compose in the `amos2022ss04-digital-identity` directory:

```
docker-compose -f docker-compose.prod.build.yml up
docker-compose up -d
```

Creation of first user:

To do so click on 'Forgot Password?' and enter the email address of the first HR employee. He gets then an email with a initial password. (That works only once on new, fresh instances of the application)

Uninstall everything:

```
docker-compose down
docker-compose down --volumes # (Also delete the database)
cd ..
rm -rf amos2022ss04-digital-identity
apt purge docker-compose docker # (If you want to also uninstall docker)
apt autoremove --purge
```

Development Environment

Docker

Follow the the instructions above. Instead of the `docker-compose.yml` file you have to edit the `docker-compose.dev.yml` file accordingly. Build up the docker stack with:

```
docker-compose -f docker-compose.dev.yml up
```

or

```
docker-compose -f docker-compose.dev.yml up -d # (For running in the background)
```

Manual

Prerequisites

For setting the development server up following modules are required: 1. Node.js 2. Angular CLI 3. Maven 4. Docker

You have to clone the project using the following command in a Git Bash terminal:

```
git clone https://github.com/amosproj/amos2022ss04-digital-identity.git
```

Frontend

Run Frontend

```
cd src/digitalIdentity-frontend/
npm install
ng serve --open
```

Access URL: `http://localhost:4200`

Testing Frontend

- For Testing the Frontend see more information here.

Backend

Init mysql database

```
sudo apt install mysql-common # or similar
sudo mysql --password  # type in your root password of linux machine
create database digital_identity;
create user 'didentity'@'%' identified by 'aosai6aH';
grant all on digital_identity.* to 'didentity'@'%';
quit;
```

Set up mail relay:

- In `src/digitalIdentity-backend/src/main/resources/application.properties` you have to define the mail settings.
- For a gmail address you need to create an app password, which you need to enter in the application.properties file. (https://support.google.com/accounts/answer/185833)
- A valid section in application.properties could look like this:

```
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=myemail@gmail.com
spring.mail.password=wajorjsyogivfugt
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
```

Set up credential for accessing lissi api:

- In `src/digitalIdentity-backend/src/main/resources/application.properties` you have to define the Credential.
- You can find the Credential in the screenshot in the mail "Links and Information" (19.05.2022) (Field: Client Secret).

```
lissi.client.id= <to be filled>
```

Run Backend

```
cd src/digitalIdentity-backend/
./mvnw spring-boot:run
```

Testing Backend

- For Testing the Backend see more information here.