iMagineFrame deployment manual

iMagineFrame Backend

The backend part of the project for iMagineLab called iMagineFrame.

Deployment

Before you can start you need to clone this git repository.

1. Configuring

In this repository there is a file called `imf-backend.env.example`, you need to copy this file to `imf-backend.env`.

Next you need to adjust all variables declared in that file to suit your needs.

> ⚠ **Important**
> Please DO change at least the passwords!

2. Building

Next we need to build a docker image we can run later. To build an image a Dockerfile is provided.

To build an image run:

```
sudo docker build --no-cache -t imagineframe-backend:latest .
```

2. Running

To run the application a docker compose file is provided.

> ⚠ **Important**
> You do need to change one thing, the docker image of the imf-backend service on line 17 to `imagineframe-backend:latest`.

To then start the application run these two commands:

```
sudo docker compose -f compose-imf-backend.yml --env-file imf-
backend.env pull
sudo docker compose -f compose-imf-backend.yml --env-file imf-
backend.env up -d
```

Project structure

Packages in src.main.java.be.uantwerpen.fti.se.imagineframe_backend

controller

Hold the controller classes where the REST endpoints are defined.

exceptionHandling

Holds all class related to exception handling. Including the global exception handler for all endpoints.

label

Enums used in the project.

model

All the models used in the project and a subpackage where all DTO's are defined.

repository

Defines the interfaces for the repositories, communication to the database.

security

Classes that involve security.

service

All services of the application. The logic that makes the application work and test to check correct values.

Other packages

- **src.main.resources**: properties files used when building and iMagineLab logo for PDF generation.
- **src.test.java.xxxx**: all the tests written to test nothing breaks during development.

Technologies

- Java
- Spring boot
- Maven

Getting Started - Development

Let Maven install all dependencies.

Run the project, the backend will be available on http://localhost:8080.

iMagineFrame Frontend

The fronted part of the project for iMagineLab called iMagineFrame.

Features

The application has many pages, listed below. These pages are designed to be as easy to use possible.

- Public pages
  - Welcoming page
  - Registration page
  - Login page
- User home page
- Users can edit their profile
- Management pages
  - User management
  - Group management
  - Registrations overview to approve or decline
- Management and user pages
  - Products
  - Events
  - Projects

Deployment

Before you can start you need to clone this git repository.

1. Configuring

In this repository there is a file called `.env.production` in this file you need to set the URL the backend will be running on.

2. Building

Next we need to build a docker image we can run later. To build an image a Dockerfile is provided.

To build an image run:

```
sudo docker build --no-cache -t imagineframe-frontend:latest .
```

2. Running

To run the image use the following command or adjust some parameter.

> ⚠ **Caution**
> Only adjust the port for the outside, NEVER change the inside port.

```
sudo docker run -d -p 3002:3000 --restart unless-stopped --name imf-frontend imagineframe-frontend:latest
```

Technologies

- Vue
- Vuetify
- NPM

Getting Started - Development

First install the dependencies:

```
npm install
```

This will create a node_modules folder in the project root. Do not commit this folder to git!

Then start the development server:

```
npm run dev
```

This will start a development server on port 3000. You can access the application at http://localhost:3000.