InstallationGuide

January 21, 2022

1 Requirements

You need to have downloaded maven in order to build and run the webservices. Apache-maven version 3.8.4 which it is known to work with can be downloaded from https://maven.apache.org/download.cgi. And java jdk 11 is also needed and can be found at https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html

2 Build And Run

To test, build and run the application simply run the build-and-run.sh script file. This will call build.sh and start.sh. build.sh builds images of all services and start.sh launches them in docker containers. Build-and-run.sh calls these scripts in order. Once the images has been tested and built once the start.sh can be used to launch the application. This will expose the facade through localhost on port 8080.

Listing 1: build and run

```
#!/bin/bash
./build_and_run.sh
```

3 Testing

To run the tests for the microservices (All projects except integrationTest) navigate to the folder for the repository eg. Reporting and then to the service folder eg. reporting-service (this is the repository name - service). In here there will be a build.sh script this will run the tests and build an image. If you only wish to run the tests in the same path you can use the mvn test command.

Listing 2: test script

```
#!/bin/bash
cd DTUPay-Payment/payment-service
./build.sh

OR

mvn test
```

To run the integration tests navigate to the repository folder for IntegrationTest. Here there will be 3 scripts, deploy.sh which deploys all services in docker containers, test.sh which runs the integration tests and shuts down the services and build-and-run.sh which calls these scripts in order. Note for these tests to work there need to be images of all the services on your computer so we recommend running the build.sh in the root folder of the zip file which builds all images first.

Listing 3: test script

```
#!/bin/bash
cd IntegrationTest
./build_and_run.sh
```

4 Git Repo

All repositories are public so you should be able to clone them without authentication. All repositories can be found below. Since the build scripts used above uses a relative path on order for it to work you need to clone release and then navigate to that folder and then clone each repository there. Eg. git clone "release" - cd "release" - git clone ...

Release: https://gitlab.gbar.dtu.dk/s194618/Release.git
IntegrationTests: https://gitlab.gbar.dtu.dk/s213685/IntegrationTest.git
Facade: https://gitlab.gbar.dtu.dk/s213685/Facade.git
Payment: https://gitlab.gbar.dtu.dk/s213685/DTUPay-Payment.git
Account: https://gitlab.gbar.dtu.dk/s213685/DTUPay-AccountManagement.git
Token: https://gitlab.gbar.dtu.dk/s213685/DTUPay-TokenManagement.git
MessageQueue: https://gitlab.gbar.dtu.dk/s213685/MessageQueue.git
Reporting: https://gitlab.gbar.dtu.dk/s213685/Reporting.git

it can be done with the following shell script:

Listing 4: Clone Script

```
#!/bin/bash
```

```
git clone https://gitlab.gbar.dtu.dk/s194618/Release.git

cd Release

git clone https://gitlab.gbar.dtu.dk/s213685/IntegrationTest.git
git clone https://gitlab.gbar.dtu.dk/s213685/Facade.git
git clone https://gitlab.gbar.dtu.dk/s213685/DTUPay-Payment.git
git clone https://gitlab.gbar.dtu.dk/s213685/DTUPay-AccountManagement.git
git clone https://gitlab.gbar.dtu.dk/s213685/DTUPay-TokenManagement.git
git clone https://gitlab.gbar.dtu.dk/s213685/MessageQueue.git
git clone https://gitlab.gbar.dtu.dk/s213685/Reporting.git
```