Danmarks Tekniske Universitet

DTU

Installation Guide

02267 Software Development of Web Services

Group 1

Student no.	Authors
s184477	Frederik Emil Schibelfeldt
s212434	Kári Sveinsson
s010219	Peter Tran
s222961	João Afonso Alves Henriques E Silva
s223186	João Luís Gonçalves Mena
s222963	Tiago Azevedo Rodrigues Silverio Machado
s212953	Þorfinnur Pétursson

Teachers:

Hubert Baumeister

January 20, 2023

Contents

1 Installation of DTU-pay

Make sure you are in the parent folder called DTU_Pay. For building, testing and running the DTU Pay application you simply need to run *build-and-run.sh* script file. To run the script execute the command:

```
$ bash build_and_run.sh
```

The script will build all of the images and install the maven dependencies that are required for DTU Pay application to function. After the script has finish it will expose the facade through the server localhost on port 8080

2 Software and tools

2.1 IDE, SDK and Build Tool

Individuals have been using their own favorite IDE, IntelliJ and Eclipse. Before starting the development, we had to agree on version used for Java SDK and the build tool Maven, to make sure we would not have development conflicts later on.

Versions used:

IntelliJ IDEA version 2022.3

Eclipse IDE 2022-12

Java SDK version 11.0.17

Apache Maven version 3.8.7

Downloads:

IntelliJ IDEA version 2022.3 can be downloaded from:

- https://www.jetbrains.com/idea/download/

Eclipse IDE 2022-12 can be downloaded from:

- https://www.eclipse.org/downloads/

Java SDK version 11.0.17 can be downloaded from:

- https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html

Apache Maven version 3.8.7 can be downloaded from:

- https://maven.apache.org/download.cgi

2.2 Source Code Management and Continuous Integration

We have been using GitHub, to keep track of our cource code. We also create a continuous integration environment, by installing Jenkins on our server and setting it up with our GitHub repository.

Versions used:

Jenkins 2.375.1

Installation:

Install Jenkins on a Debian-based distributions like Ubuntu using command line:

```
$ curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | \
sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null

$ echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null

$ sudo apt-get update

$ sudo apt-get install jenkins
```

2.3 Message broker

We have been using RabbitMQ to establish communication between microservices.

Versions used:

RabbitMQ 5.10.0

Installation:

We make use of RabbitMQ Java client by including it using the central Maven repository. Including it in the pom.xml file as following:

```
<dependencies>
    <dependency>
        <groupId>com.rabbitmq</groupId>
        <artifactId>amqp-client</artifactId>
        <version>5.10.0</version>
    </dependency>
</dependencies>
```

2.4 Container Application

Every services in this project has been dockerized. This is done with Docker-compose.

Versions used:

Docker 1.29.2

Installation:

```
$ sudo curl -L "https://github.com/docker/compose/releases/download/ \
1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o \
/usr/local/bin/docker-compose

$ sudo chmod +x /usr/local/bin/docker-compose
```

3 Access to tools

3.1 Github access

The account **hubertbau** has been added as a developer to our Github repository, which can be found at:

- https://github.com/FiloGrilo/DTU_Pay

3.2 Jenkins access

To access our Jenkins Server please use the following URL and credential:

- http://138.68.65.135:8282/
- Username: hubert
- Password: comeonin