Technical University of Denmark

Technical University of Denmark

02267 Software Development of Web Services

DTUPay - Installation Guide

Tobias Rydberg (s173899)
Daniel Larsen (s151641)
Emil Kosiara (s174265)
Troels Lund (s161791)
Sebastian Lindhard Budsted (s135243)
Kasper Lindegaard Stilling (s141250)

Group 11

January 19, 2021

Introduction

This guide describes how to build, install, run and test the DTUPay system. Additionally, source code repositories and detailed descriptions of the tools used for development and project management is available together with common troubleshooting steps.

Source code

The source code for the project is version controlled using Git and hosted on GitHub. You can find the source code at the following link.

- GitHub repository: `https://github.com/02267-SDWS/dtu-pay`

The latest and stable build is residing in the `main` branch of the repository, so this is what you should clone or download when installing the system.

The repository holds seven different projects.

- account-service
- client
- payment-service
- report-service
- rest
- token-service
- utils

Docker files, docker-compose configuration, build scripts and OpenAPI specifications reside in the root of the repository as well. For more information, it is advised to take a look at the `README.md` at the root of the repository.

Installation Requirements

Before proceeding with the actual installation of the system, a few requirements need to be satisfied for it to succeed. You should make sure that these required dependencies are installed on the system before starting the next step.

OpenJDK 11 As the project is written in the Java programming language, it is necessary to download a Java Development Kit to be able to build the system. OpenJDK 11 has been used to develop the system, so it is advised that you download the same version. Download and installation instructions can be found here: https://openjdk.java.net/install/

Apache Maven The build system used for the project is Apache Maven, so installation of Maven is required to be able to run the build scripts. Download and installation instructions can be found here:
https://maven.apache.org/download.cgi

Docker The project uses Docker for containerization of each of the microservices. Installation instructions for Docker can be found here: https://docs.docker.com/get-docker/

docker-compose To automate the build and execution of the Docker containers, docker-compose is being used. Download and install instructions can be found here: https://docs.docker.com/compose/install/

Once all of the above dependencies are installed correctly, you can proceed to the next step.

Installation & Run Procedure

Once the requirements in the previous section are satisfied, the system can be installed by following these steps.

1) Download or clone the GitHub repository at https://github.com/02267-SDWS/dtu-pay. Make sure that it is done from the `main` branch.

2) Open up a local terminal and change directory to the root of the downloaded or cloned repository.

3) Once inside the root of the repository, execute the `build_and_run.sh` script. This script will first build all the binaries (`.jar` files) then build the Docker containers and run them with docker-compose. This will most likely take a few minutes to finish as some docker images will be downloaded as well.

4) If everything has succeeded, you should now be able to access the REST interface service on `localhost:8080`. You can access its SwaggerUI and play around with the system by appending `/swagger-ui` to the end of the URL. You can access the report-service at `localhost:8083`. You can use its SwaggerUI as well. You can find OpenAPI specifications for both of these at the root of the repository.

You can now proceed to the test part of the guide.

Test Procedure

After installation and run of the system has been successful, you are now able to run the system tests.

1) Just like before, open up a local terminal and change directory to the root of the downloaded or cloned source code repository.

2) Once inside the root of the repository, execute the `mvn_test_all.sh` script. This script will run individual tests for all of the projects and show the result in the terminal.

If all of the tests have succeeded, the system is stable and ready to use.

Troubleshooting

Here are a few of the most common issues that one might run into when installing, running or testing the system.

Docker issues If you have any issues with running docker commands, you might need to add yourself to the `docker` group if you are running Linux. An alternative could be to use `sudo` to run the command as root user.

Script execution issues If the scripts are not able to be executed on Linux, you might need to change their execution permission. You can do that by running `chmod +x <scriptfile>` to make the logged in user get execution rights.

Tools

A myriad of different editors and IDEs, build, containerization and automation tools have been used during development. Below you will find a detailed description of the tools that have been used together with their versions and download links. It is advised to consult the respective installation instructions for your specific operating system and architecture.

Software Development Kits

- OpenJDK (Version 11)

Editors & IDEs

- IntelliJ IDEA Ultimate (Version 2020.3.1, 2020.2.2)
- Visual Studio Code (Version 1.52.1)
- Neovim (Version 0.4.4)
- Vim (Version 8.2)

Other

- docker-compose (Version 1.27.4)
- Apache Maven (Version 3.6.3)
- OpenSSH (Version 8.4p1)
- Jenkins (Version 2.263.1)
- Docker (Version 20.10.2)
- Git (Version 2.30.0)
- GitHub Projects
- GitHub

Downloads

- OpenDJK: https://openjdk.java.net/install/
- IntelliJ IDEA Ultimate: https://www.jetbrains.com/idea/download
- Visual Studio Code: https://code.visualstudio.com/download
- Neovim: https://github.com/neovim/neovim/wiki/Installing-Neovim
- Vim: https://www.vim.org/download.php
- docker-compose: https://docs.docker.com/compose/install/
- Apache Maven: https://maven.apache.org/install.html
- OpenSSH: https://www.openssh.com/
- Jenkins: https://www.jenkins.io/download/
- Docker: https://docs.docker.com/get-docker/
- Git: https://git-scm.com/downloads
- GitHub Projects: https://github.com/features/project-management/ (No Download)
- GitHub: https://github.com/ (No Download)