Installation Guide

This is a repo containing 7 sub-projects for the DTUPay Web Service:

REST application: DTUPay REST Application

Microservices: Account Service Payment Service Token Service Report Service

Message Queue: Message Utilities

End-to-end Tests: End-to-end Tests

Installing Java

You will need Java (=version 11.0.13). To check whether it's already installed on a UNIX-like system, open up a terminal window (e.g. Terminal on OS X) and type `java --version` at the command prompt. For example, you should see something similar to the following:

```
 $ java --version
java 11.0.13 2021-01-19
Java(TM) SE Runtime Environment (build 11.0.13+8-Ubuntu-0ubuntu1.20.04)
Java HotSpot(TM) 64-Bit Server VM (build 11.0.13+8-Ubuntu-0ubuntu1.20.04, mixed mode, sharing)
```

Installing Maven

You will need Maven (>=version 3.6.3). To check whether it's already installed on a UNIX-like system, open up a terminal window (e.g. Terminal on OS X) and type `mvn --version` at the command prompt. For example, you should see something similar to the following:

```
 $ mvn --version
Apache Maven 3.6.3
Maven home: /usr/local/Cellar/maven/3.6.3/libexec
Java version: 11.0.13, vendor: Oracle Corporation, runtime: /Library/Java/JavaVirtualMachines/jdk-11.0.13.jdk/Contents/Home
Default locale: en_CN, platform encoding: UTF-8
OS name: "mac os x", version: "10.15.7", arch: "x86_64", family: "mac"
```

Installing Docker

Get Docker (version 20.10.7, build 20.10.7-0ubuntu5~20.04.2)

Installing Jenkins

If you would like to work with Continuous Integration. A recommodation is Jenkins (version 2.319.1). Or you can start the Specific Runner (Not the Shared Runner) of gitlab pipeline and remove the `.gitlab-ci.yml` from `.gitignore` to use the Gitlab CI/CD.

Build And Run

All the required dependencies are included in the pom.xml for each sub-project and will be pulled automatically. The `Message Utilities` is an abstraction of the RabbitMq and should be installed as a libarary for other sub-projects.

Each sub-project can be built independently and dockerized into docker images.

The `build_and_run.sh` is all you need for building and running all the tests, it includes

- compile maven (java) projects
- build docker images
- run level test for microservices
- deploy the Web Service (docker compose)
- run the end-to-end test

```
 $ ./build_and_run.sh
```

The ouput should be as follows with all the end-to-end tests passed:

```
 22 Scenarios (22 passed)
146 Steps (146 passed)
0m9.261s
```

The following warning may appear for the code does not close the client directly.

> WARN: RESTEASY004687: Closing a class org.jboss.resteasy.client.jaxrs.engines.ApacheHttpClient43Engine instance for you. Please close clients yourself.

README Author @Bingkun