# Technical User Documentation

### Contents

The document contains technical details which will be useful to maintain and extend existing application. It also covers some aspect of design/architecture of the application. Scope of improvement section explains ideas the way application can be enhanced based on future requirements.

### Motivation

The document is aimed to help understand existing design and let other developers to give better understanding of dependencies and constraints if any.

### Assumptions

It is assumed that the requirement has by mistake mentioned service url (<https://api.lloydsbank.com/open-banking/v2.2/atms>) to be a request parameter, and hence it is not added in request parameters of exposed service but configured at the application level.

### Technical specification

Below is the list of libraries and plugins used to build the application. The dependencies of these libraries and plugins are found in **build.gradle** file.

|  |  |  |
| --- | --- | --- |
| Name | Version | Usage |
| Openjdk | 11 | Core library used to build and run the application |
| gradle | 6.8.3 | To build the application |
| springboot | 2.6.2 | Framework used for application configuration and development |
| Apache HTTP Client | 4.5.13 | To create cacheable HTTP client |
| Springfox swagger | 2.7.0 | To generate client code from swagger |
| Springdoc openapi | 1.6.4 | To generate swagger from API |
| Eclipse JDT | 2.2.0 | To use Nullness annotations |

### Features

* Application generates client from swagger provided by service provider and also generates swagger documentation for services exposed by application.
* The application has dockerfile which lets you build an image and run under docker container.

### Scope of extension

The application is developed under very limited scope to demonstrate functionality as requested. As application grows, based on business and technical requirement, some of below points can be considered for future extension.

* Currently caching is provided at the level of HTTP client, which can be extended at the data level to reduce requests to server when data is not expected to be volatile from service provider.
* Logging is configured to write logs on console, which needs to be extended

### Installation

**Check installation of gradle and Java** on your system using below commands,

> gradle -v

Above command should work if gradle and java is correctly set on Path in Windows.

Now **check installation of docker** in your system with below command.

> docker --version

You can also check running status of containers using below docker command

> docker ps –a

To **build the application**, move to directory RestClientBuilder under the project root directory

> cd <Root dir>\RestClientBuilder

> gradle clean build

This will **create client from swagger** of Lloyds bank swagger specification.

Once client is built, move to root directory and **build the project**

> cd <Root dir>

> gradle clean build

Now **build a docker image** using below command,

> cd <Root dir>

> docker build -t "atm-locator-app"

**Run container and load above image** using below command,

> docker run -it -p 8080:8080 atm-locator-app

Send GET request with URL parameter as below to test the application,

<http://localhost:8080/locator-service/atms/LFFFBC11>

You can access swagger UI on below localhost URL

<http://localhost:8080/swagger-ui/index.html>