Cloud-based application: Assignment 1

Part 1: Building a Spring-boot web service

Introduction

For the first part of the assignment, I've decided to use `gradle` as the build tool for the application. The whole application was written in Java using the standard division of classes (service classes, controller classes, ...).

How does the application work

The application entry point starts from the api access point `/api/getPopulation?year=<X>`. When requesting this entry point the system will trigger the service class that will fetch all the needed information from the Statec API. Once the service will receive the information, another system (method) will be then responsible for analyzing the data and retrieving the population for the requested year. If a system does not find any population values for that specific year, it will retrieve the 2 nearest years that do have valid population numbers.

The web service will then retrieve an array with the respective year(s) population figures:

```
[
    {
        "year": 1991,
        "totalPopulation": 383421,
        "totalFemalePopulation": 195465,
        "totalMalePopulation": 187956,
        "totalLuxembourgishMalePopulation": 130592,
        "totalLuxembourgishFemalePopulation": 138677,
        "totalForeignMalePopulation": 57364,
        "totalForeignFemalePopulation": 56788
    },
    {
        "year": 2001,
        "totalPopulation": 439539,
        "totalFemalePopulation": 222999,
        "totalMalePopulation": 216540,
        "totalLuxembourgishMalePopulation": 135373,
        "totalLuxembourgishFemalePopulation": 141881,
        "totalForeignMalePopulation": 81167,
        "totalForeignFemalePopulation": 81118
    }
]
```

How to run the application locally with only gradle

In order to run the application using gradle, you shall run the folllowing comment at the root of the Spring-boot project:

```
# building the app
./gradlew build
# running the app
./gradlew bootRun
```