Rest Web Service with Spring boot

The goal of this lab work is to develop a small application for cars renting.

The functionalities to be implemented are:

- Get a list of unrented cars
- Rent a car
- Get back a car

Required software

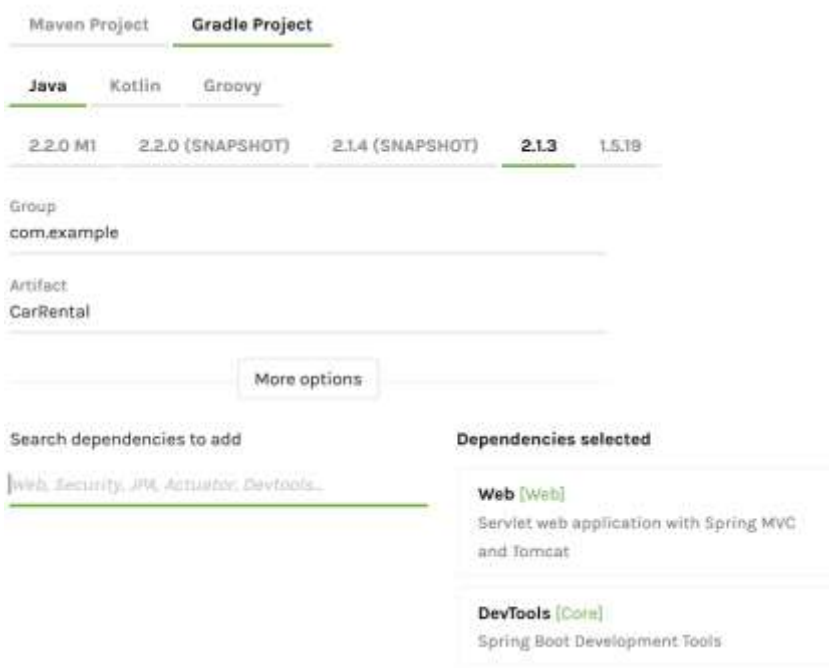
Java (JDK version)

Eclipse

Gradle project creation

Use the Spring Initializer (https://start.spring.io/) to create a project:

- Don't forget to choose Gradle Projetct
- Give it a name (Articfact)
- Add Web as dependencies (library)
- Add DevTools as dependencies



Download and unzip the project (outside the Eclipse workspace)

Open a command line window (project location).

Use the following command to build the project (download libraries, compilation…):

- gradlew build under windows
- ./gradlew build under Linux

Add the plugin Eclipse (or idea if you want to use Intellinj instead of Eclipse) to the configuration file build.gradle:

```
apply plugin: 'java'
apply plugin: 'eclipse'
apply plugin: 'idea'
```

Use the following command to convert the project in an Eclipse project:

- gradlew eclipse under windows or gradlew idea for Intellij
- ./gradlew eclipse under Linux or ./gradlew idea for Intellij

Import the project under Eclipse: File->Import->General-> Existing project into workspace ... select the project directory

Launch the main program: /src/main/java/package…/*Application.java

Spring boot coding

Write a class annotated with Controller implementing the car rental service (see https://github.com/charroux/CarService for an example).

Then code the following features:

- Get the features of a car:
  - URI: …/cars/plateNumber
  - http GET
  - Json response: { " plateNumber" : "11AA22" , " nomberOfSeats" : 5, "price" : 100 }
- Rent a car:
  - URI: …/plateNumber?rent=true
  - http PUT
  - Send Json inside the http body: { "begin" : "11/11/2017" , " end" : "1/1/2018" }
- Get back the car:
  - URI: …/plateNumber?rent=false
  - http PUT

Advice: you can use the following templates.

```
@RequestMapping(value = "/cars", method = RequestMethod.GET)
@ResponseStatus(HttpStatus.OK)
@ResponseBody
public List<Car> listOfCars(){
}
```

```
        @RequestMapping(value = "/cars/{plateNumber}", method = RequestMethod.GET)
        @ResponseStatus(HttpStatus.OK)
        @ResponseBody
        public Car aCar(@PathVariable("plateNumber") String plateNumber) throws Exception{
        }

        @RequestMapping(value = "/cars/{plateNumber}", method = RequestMethod.DELETE)
        @ResponseStatus(HttpStatus.OK)
        public void getBack(@PathVariable("plateNumber") String plateNumber) throws Exception{
        }

        @RequestMapping(value = "/cars/{plateNumber}", method = RequestMethod.PUT)
        @ResponseStatus(HttpStatus.OK)
        public void rent(@PathVariable("plateNumber") String plateNumber) throws Exception{
        }

        @RequestMapping(value = "/voiture/{plateNumber}", method = RequestMethod.PUT)
        @ResponseStatus(HttpStatus.OK)  public void
rentAndGetBack(@PathVariable("plateNumber") String plateNumber,
@RequestParam(value="rent", required = true)boolean rent) throws Exception{
        }

        @RequestMapping(value = "/cars/{plateNumber}", method = RequestMethod.PUT)
        public void rent(@PathVariable("plateNumber") String plateNumber, @RequestParam(value="rent",
required = true)boolean rent, @RequestBody Dates dates){

        }
```

Test your application

Test the web service inside a web browser: http://localhost:8080/cars

Use a plugin for web browser like RestClient or Postman to test your application.