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

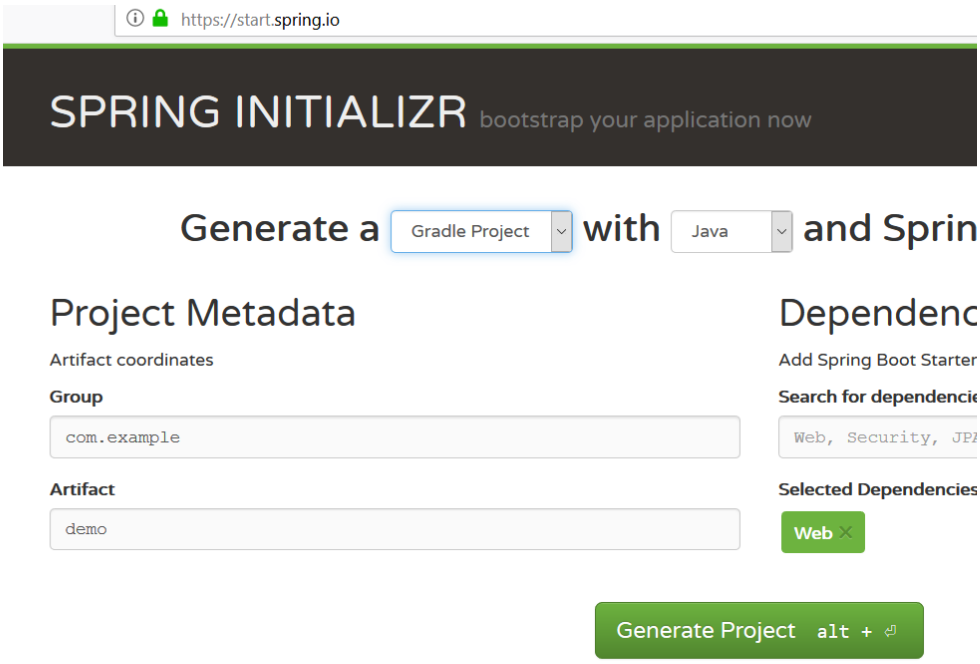
Java (JDK version)

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)



https://start.spring.io

SPRING INITIALIZR bootstrap your application now

Generate a Gradle Project with Java and Sprin

Project Metadata

Artifact coordinates

Group: com.example

Artifact: demo

Dependenc

Add Spring Boot Starter

Search for dependencie: Web, Security, JPA

Selected Dependencies: Web ×

Generate Project alt + ⏎

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

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

-	gradlew eclipse	under windows
-	./gradlew eclipse	under Linux

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

Rest Web Service design (see the slides of t course)

Define JSon structures modeling a car to be rented and a list of car.

Define URI for a list of cars to be rented.

Accordingly, to Restful architecture define the protocol (using HTTP request get, put, post and delete) for a car rental service.

Spring boot coding

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

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{
        }
```

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.