# Microservices lab 1

## Tools

Operating system : Linux-based, macOS or Windows (not recommended)

**JDK 11**
Mac users : I recommend you install it with sdkman https://sdkman.io/.

IDE : **IntelliJ IDEA Ultimate** https://www.jetbrains.com/idea/download is highly recommended (free licence for students https://www.jetbrains.com/shop/eform/students).
You can also use Eclipse or VSCode.

## Goal

Create a microservice exposing data through a Rest api.

## Steps

The next steps are assuming that you're going to build a Spring Boot based application.

1. Create a Spring Boot app using https://start.spring.io/
    a. Select a **Gradle** project
    b. Select a **Kotlin** or **Java** project
    c. Add dependencies on **Web** and **Actuator**

2. Run the following command to build an executable jar :
   `./gradlew build`

   All the dependencies are inside, including the app server (Tomcat by default).

3. Run the following command to start the Spring Boot app :
   `java -jar build/libs/my-app-0.1.0.jar`
   **or**
   `./gradlew bootRun`

4. The server should now be running on port 8080. Check that everything is working properly by doing a HTTP request on :
   `localhost:8080/actuator/health`

5. Create a Rest api (Create Read Update Delete) :
    a. You can imagine whatever you want while you can also stick with the movies service and create an api about movies
    b. Create an endpoint that allows to requests all movies
    c. Create an endpoint that allows to requests one specific movie with its unique id
    d. Create an endpoint that allows to create a new movie
    e. Create an endpoint that allows to update a movie
    f. Create an endpoint that allows to delete a movie

6. Test your api with a REST client, Postman https://www.getpostman.com/ for example.

7. Add a dependency on spring-data-jpa and H2 (in-memory database). Feel free to use any other database relational or not.

8. Plug your REST api to a spring-data repository.

9. Test your api.