SPRING BOOT WITH REST SDJ3 (REF [Spring Boot Restful Web Services CRUD Example - YouTube](https://www.youtube.com/watch?v=YYwFhclOlqs&ab_channel=JavaGuides))

1. Create the project

Graphical user interface

Description automatically generated

1. We need now to search for springboot dependencies.

(JPA +spring web and H2 for local db )

Those dependencies will be added on maven file and will force java to use those.

Graphical user interface, text

Description automatically generated

1. Locate Main file and run

Like on normal java application the RUN command for the program it will be located in “SpringbootRestApplication”

A screenshot of a computer screen

Description automatically generated with medium confidence

By default the program will run Tomcat (web hosting server) on port 8080 so if you want to access the application you need to use “localhost:8080” or <http://localhost:8080/>

Graphical user interface, text, application, email

Description automatically generated

1. Create a model and controller

First we need a classic model (Employee or User up to you)

(In my case I’ve create a normal class with gets/sets and constructor)

Text

Description automatically generated

Because I want to use H2 (hibernate), I can pre-state a class to generate the table into database. (For using normal jdbc I will provide a different git repository with an example, but the principle will be the like in sep2)

For me example:

Text

Description automatically generated

I notify the class that will be in table “employees” and after serialize the variables for ID to generate the value auto and mention that the variable “id” it will be classify ID.

And for other variable just set the name for colums.

5.We also need to create a repository for it (a interface)

A screenshot of a computer

Description automatically generated with medium confidence

6.Now we need to create a controller from where we can handle all the REST API endpoints

Text

Description automatically generated

We create a specific function that will create/update/remove/get and serialize the function with GetMapping/PutMapping/PostMapping and DeleteMapping

7.Test with postman

First I want to create a user and stored into hibernate database, using postman I will reach to the specific endpoint (what we set in controller), and to use post, after that for the body because the POST method need a body in JSON format, I will use the format that is in the Employee class in java where attributes are the variables and the value of those.

A screenshot of a computer

Description automatically generated

After we run this we can see that we get a 200 (OK) that mean the user have been created into system and is in database as well.

A screenshot of a computer

Description automatically generated

(I will create more users to run the get all method and get by ID)

For get all method we will use the same endpoint but just GET instead of POST (this method can be call as well from the browser)

After we call this function we will get a response in JSON format a list of employee

A screenshot of a computer

Description automatically generated

If we want to search only by ID we need to add “/idnumber” after (e.x: “http://localhost:8080/api/v1/employees/1”)

A screenshot of a computer

Description automatically generated with medium confidence

In this case will only return the details of one user

For update we will use PUT and using the same endpoint as get user by ID, using PUT method we will need to provide a body as well for request.

Graphical user interface, application

Description automatically generated

In my case I’ve update last name from Test to Adrian, if we run get user by ID we will see that the modification have been set

And for delete we will use the same endpoint just using DELETE and we will not need any request body but we can see that after we execute the request we will get 200 (OK).

A screenshot of a computer

Description automatically generated with medium confidence

FOR those who still love to use the default JDBC:

[sdj3-a21/cars-spring-rest/src/main/java/dk/via at master · olehougaard/sdj3-a21 (github.com)](https://github.com/olehougaard/sdj3-a21/tree/master/cars-spring-rest/src/main/java/dk/via)

This one is an exercise with REST API in springboot and using JDBC driver to database made by Ole Hougaard.

My Github exercise: [GoForMusic/springbootRESTsdj3: Springboot application using REST-test with postman and H2 for db (github.com)](https://github.com/GoForMusic/springbootRESTsdj3/tree/master)