**Creating Microservices for Account and Loan**

In this hands-on exercise, we will create **two microservices for a bank**:

* One microservice for handling **accounts**
* One microservice for handling **loans**

Each microservice will be a **Spring Boot RESTful Web Service**, built as a **Maven project** with its own pom.xml. These services will not connect to any backend for now — just return dummy responses.

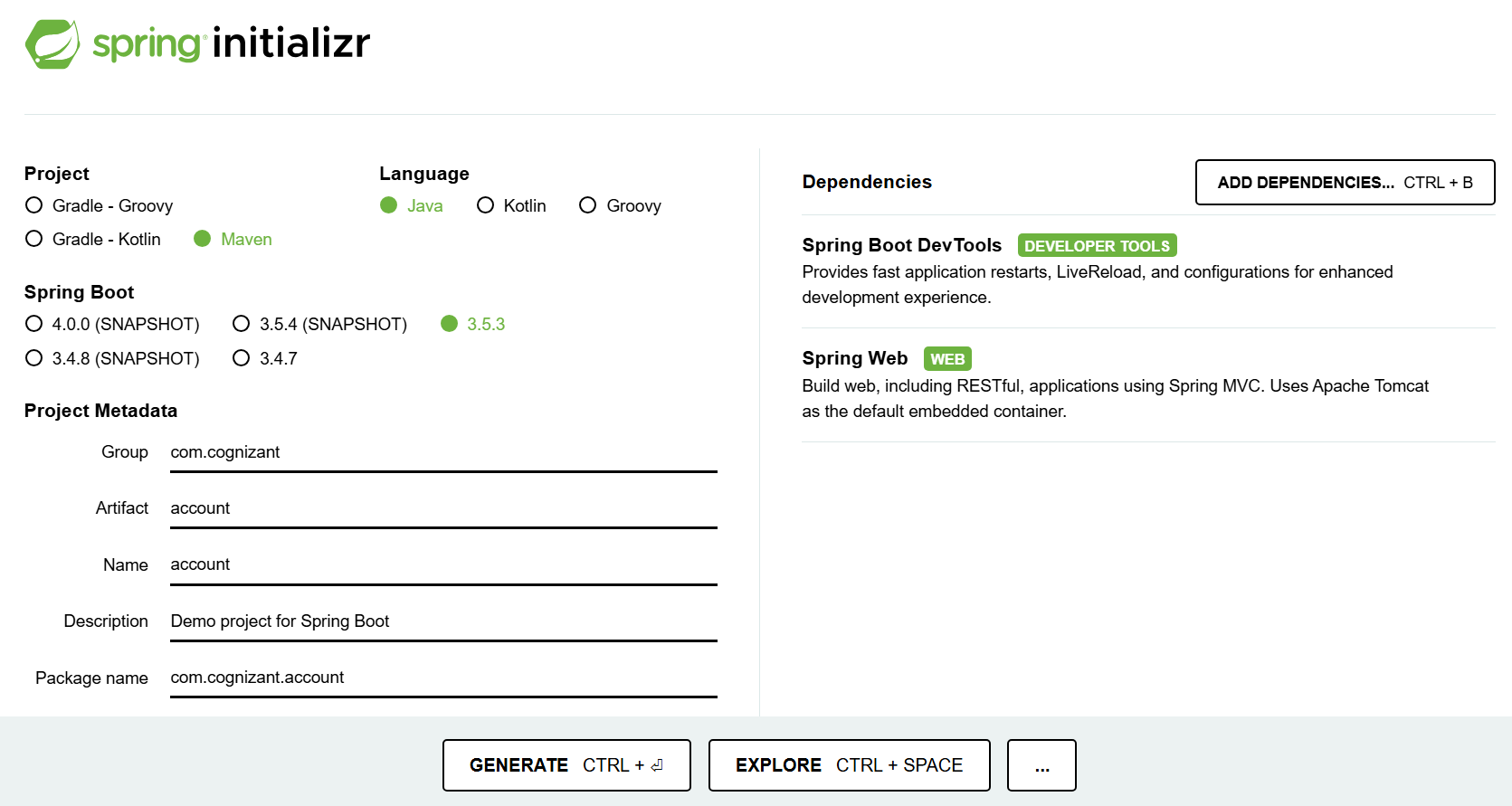
**📁 Folder Structure Setup**

1. Create a folder with your **employee ID** in the D: drive.
2. Inside that folder, create another folder named microservices.  
   This will hold all the sample projects we create.

**🔧 Account Microservice**

**Step-by-Step Instructions:**

1. Go to <https://start.spring.io>
2. Fill in the following fields:
   * **Group:** com.cognizant
   * **Artifact:** account
3. Select dependencies:
   * *Developer Tools* > Spring Boot DevTools
   * *Web* > Spring Web
4. Click **Generate** to download the zip file.
5. Extract the downloaded zip file.
6. Move the extracted account folder into your D:\<employee\_id>\microservices folder.



**Build the Project**

1. Open **Command Prompt** inside the account folder.
2. Run the following Maven command:

**Import and Implement in Eclipse**

1. Open **Eclipse** and import the account project.
2. Create a controller class to expose a REST API.

**📄 Controller Specification**

* **Method:** GET
* **Endpoint:** /accounts/{number}
* **Response:** (No backend, just a dummy string)

**Example Output:**

{ number: "00987987973432", type: "savings", balance: 234343 }

**Application Code:**

package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class AccountApplication {

public static void main(String[] args) {

SpringApplication.run(AccountApplication.class, args);

}

}

**Controller Code:**

package com.cognizant.account.controller;

import org.springframework.web.bind.annotation.\*;

@RestController

@RequestMapping("/accounts")

public class AccountController {

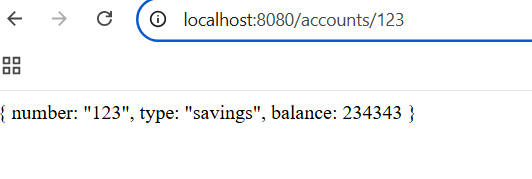
@GetMapping("/{number}")

public String getAccountDetails(@PathVariable String number) {

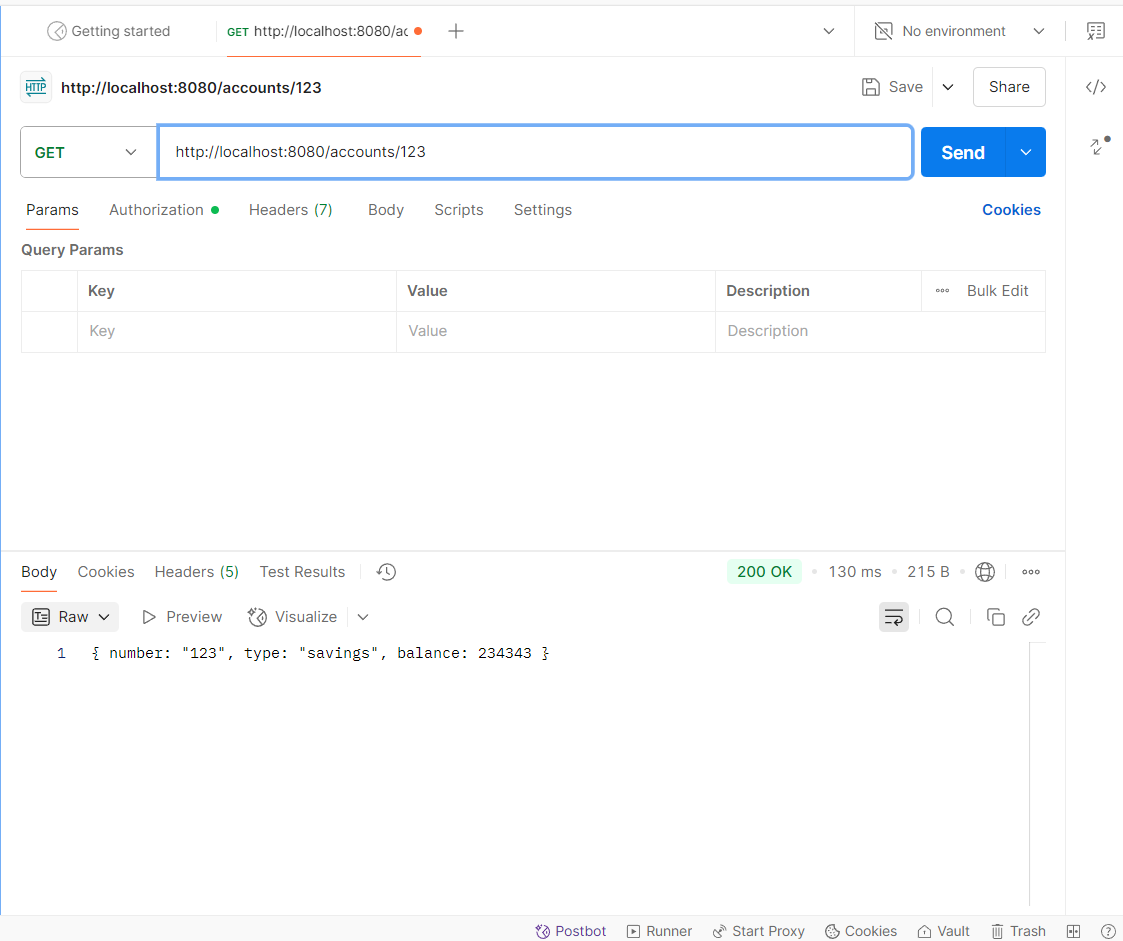
return "{ number: \"" + number + "\", type: \"savings\", balance: 234343 }";

}}

**Chrome:**

}

**Postman:**



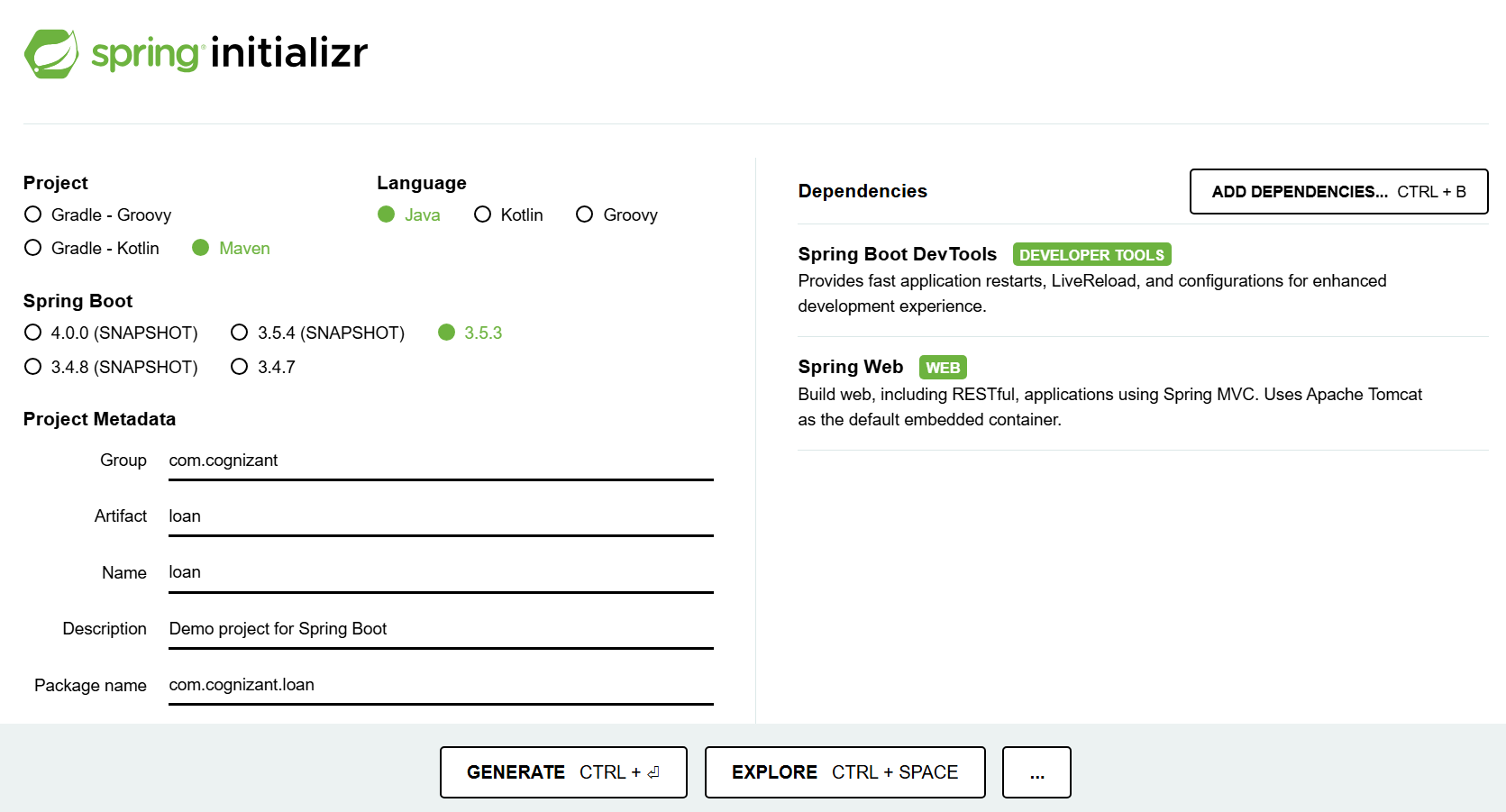
**Loan Microservice Setup**

In this exercise, you'll create a **Loan Microservice** for a bank. It will be an independent Spring Boot RESTful Web Service (just like the Account service) with a simple dummy response and **no backend connection**.

**Loan Microservice Creation**

**Step-by-Step Instructions:**

1. Go to <https://start.spring.io>
2. Fill in the following fields:
   * **Group:** com.cognizant
   * **Artifact:** loan
3. Select dependencies:
   * *Developer Tools* > Spring Boot DevTools
   * *Web* > Spring Web
4. Click **Generate** to download the zip file.
5. Extract the downloaded zip file.
6. Move the extracted loan folder into:  
   D:\<employee\_id>\microservices



**Build the Loan Project**

1. Open **Command Prompt** inside the loan folder.
2. Run the following Maven command:

**Import and Implement in Eclipse**

1. Open **Eclipse** and import the loan project.
2. Create a controller class to expose a REST API.

**Controller Specification**

* **Method:** GET
* **Endpoint:** /loans/{number}
* **Response:** (No backend, just return dummy string)

**Example Output:**

{ number: "H00987987972342", type: "car", loan: 400000, emi: 3258, tenure: 18 }

**Application Code**

package com.cognizant.loan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class LoanApplication {

public static void main(String[] args) {

SpringApplication.run(LoanApplication.class, args);

}

}

**Controller Code**

package com.cognizant.loan.controller;

import org.springframework.web.bind.annotation.\*;

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{number}")

public String getLoanDetails(@PathVariable String number) {

return "{ number: \"" + number + "\", type: \"car\", loan: 400000, emi: 3258, tenure: 18 }";

}

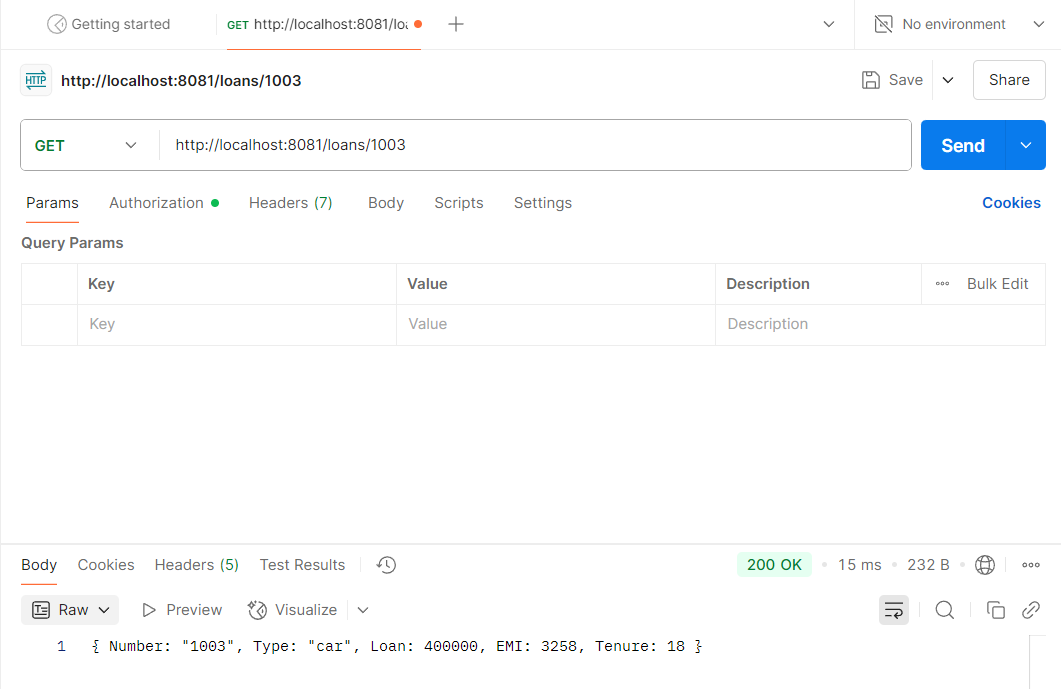
}

**Run and Test Loan Microservice**

1. Start the **Loan** service (now configured to run on 8081).
2. Test in browser/Postman:

<http://localhost:8081/loans/H00987987972342>

postman-



Chrome-