# Week 5 - Microservices with Spring Boot 3 and Spring Cloud - Hands-on

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 an independent Spring RESTful Webservice Maven project with its own `pom.xml`.

These services are simple and have no backend connectivity.

------------------------------------------------------------  
Account Microservice

1. Create a folder with your employee ID in the D: drive.  
2. Inside this folder, create another folder named 'microservices'.  
3. Open https://start.spring.io/ in your browser.  
4. Enter the following:  
 - Group: `com.cognizant`  
 - Artifact: `account`  
5. Select modules:  
 - Developer Tools > Spring Boot DevTools  
 - Web > Spring Web  
6. Click Generate and download the zip file.  
7. Extract the 'account' folder and place it inside the 'microservices' folder.  
8. Open Command Prompt in the 'account' folder and run:  
 `mvn clean package`  
9. Import the project in Eclipse.  
10. Implement a controller method to get account details by account number.

- Method: `GET`  
 - Endpoint: `/accounts/{number}`  
 - Sample Response:  
 ```json  
 {  
 "number": "00987987973432",  
 "type": "savings",  
 "balance": 234343  
 }  
 ```  
11. Run the application and test the service in the browser.

------------------------------------------------------------  
Loan Microservice

1. Repeat the same steps as the account microservice.  
2. Create a project named `loan` instead of `account`.  
3. Implement the controller:

- Method: `GET`  
 - Endpoint: `/loans/{number}`  
 - Sample Response:  
 ```json  
 {  
 "number": "H00987987972342",  
 "type": "car",  
 "loan": 400000,  
 "emi": 3258,  
 "tenure": 18  
 }  
 ```

4. While running this project, if the account service is already running, you may face a port conflict.  
5. To fix it, add the following property in `application.properties`:  
 ```  
 server.port=8081  
 ```  
6. Now launch the application again.

You now have two microservices running on different ports:  
- Account service: `http://localhost:8080/accounts/{number}`  
- Loan service: `http://localhost:8081/loans/{number}`

Note: Use the Eclipse console view to switch between service logs using the monitor icon.