

JEE 2

Premier Micro services - TP



<https://start.spring.io/>



Project

- Gradle - Groovy Gradle - Kotlin
 Maven

Language

- Java Kotlin Groovy

Spring Boot

- 3.2.0 (SNAPSHOT) 3.2.0 (M3) 3.1.5 (SNAPSHOT) 3.1.4
 3.0.12 (SNAPSHOT) 3.0.11 2.7.17 (SNAPSHOT) 2.7.16

Project Metadata

Group	org.sid
Artifact	bank-account-service
Name	bank-account-service
Description	Demo project for Spring Boot
Package name	org.sid.bank-account-service
Packaging	<input checked="" type="radio"/> Jar <input type="radio"/> War
Java	<input type="radio"/> 21 <input checked="" type="radio"/> 17 <input type="radio"/> 11 <input type="radio"/> 8

Dependencies

[ADD DEPENDENCIES... ⌘ + B](#)

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Data JPA SQL

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

H2 Database SQL

Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

Lombok DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

Spring for GraphQL WEB

Build GraphQL applications with Spring for GraphQL and GraphQL Java.



twitter.com/springboot

[GENERATE ⌘ + ↵](#)

[EXPLORE CTRL + SPACE](#)

[SHARE...](#)

Premier Micro services - TP

The screenshot shows a Java project structure in the left sidebar and the code editor for `BankAccount.java` in the center.

Project Structure:

- bank-account-service** (~/Downloads/bank-account-service)
 - > .idea
 - > .mvn
 - ✓ **src**
 - ✓ **main**
 - ✓ **java**
 - ✓ **org.sid.bankaccountservice**
 - > **entities** (highlighted)
 - > enums
 - > repositories
 - > web

Code Editor (BankAccount.java):

```
3 import lombok.AllArgsConstructor;
4 import lombok.Builder;
5 import lombok.Data;
6 import lombok.NoArgsConstructor;
7 import org.sid.bankaccountservice.enums.AccountType;
8
9 import javax.persistence.Entity;
10 import javax.persistence.Enumerated;
11 import javax.persistence.Id;
12 import java.util.Date;
13
14 @Entity
15 @Data @NoArgsConstructor @AllArgsConstructor @Builder
16 public class BankAccount {
17     @Id
18     private String id;
19     private Date createdAt;
20     private Double balance;
21     private String currency ;
22     @Enumerated(EnumType.STRING)
23     private AccountType type;
24 }
25
```

The code uses Lombok annotations (`@Entity`, `@Data`, `@AllArgsConstructor`, `@NoArgsConstructor`, `@Builder`) and Java Persistence API annotations (`@Id`, `private Date createdAt`, `private Double balance`, `private String currency`, `@Enumerated(EnumType.STRING)`, `private AccountType type`).

Premier Micro services - TP

The screenshot shows a Java project structure on the left and an IDE editor on the right.

Project Structure:

- bank-account-service (~/Downloads/bank-account-service)
 - .idea
 - .mvn
 - src
 - main
 - java
 - org.sid.bankaccountservice
 - entities
 - enums (highlighted with a blue bar)
 - repositories
 - web
 - BankAccountServiceApplication

intServiceApplication.java AccountRestController.java application.properties AccountType.java

```
1 package org.sid.bankaccountservice.enums;
2
3     5 usages
4     public enum AccountType {
5         1 usage
6             CURRENT_ACCOUNT, SAVING_ACCOUNT
7     }
```

3. Cree un package enums
 4. Cree un enumerateur
- AccountType.java

Premier Micro services - TP

The screenshot shows a Java code editor with the following details:

- Project Structure:** On the left, the project tree shows a single module named "bank-account-service" located at "/Downloads/bank-account-service". It contains ".idea", ".mvn", "src", and "main" directories. "main" has "java", "org.sid.bankaccountservice", and "BankAccountServiceApplication" files. "org.sid.bankaccountservice" contains "entities", "enums", "repositories", "web", and "BankAccount.java".
- Code Editor:** The main window displays the "BankAccountRepository.java" file. The code is as follows:

```
1 package org.sid.bankaccountservice.repositories;
2
3 import org.sid.bankaccountservice.entities.BankAccount;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface BankAccountRepository extends JpaRepository<BankAccount, String> {
7 }
```
- Toolbars and Status:** At the top, there are tabs for "BankAccount.java", "BankAccountRepository.java" (which is selected), "BankAccountServiceApplication.java", and "AccountR". A green checkmark icon is visible on the right.

5. Cree un package repositories

6.Cree une interface

BankAccount.java

Premier Micro services - TP

```
bank-account-service ~/Downloads/bank-account-service
> .idea
> .mvn
< src
  < main
    < java
      < org.sid.bankaccountservice
        entities
```

7. Dans BankAccountServiceApplication.java

The screenshot shows a Java code editor with the following details:

- File:** BankAccountServiceApplication.java
- Content:**

```
package org.sid.bankaccountservice;
import ...
@SpringBootApplication
public class BankAccountServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(BankAccountServiceApplication.class, args);
    }

    no usages
    @Bean
    CommandLineRunner start(BankAccountRepository bankAccountRepository){
        return args -> {
            for (int i = 0; i < 10; i++) {
                BankAccount bankAccount=BankAccount.builder()
                    .id(UUID.randomUUID().toString())
                    .type(Math.random()>0.5? AccountType.CURRENT_ACCOUNT:AccountType.SAVING_ACCT)
                    .balance(10000+Math.random()*90000)
                    .createdAt(new Date())
                    .currency("MAD")
                    .build();
                bankAccountRepository.save(bankAccount);
            }
        };
    }
}
```
- Toolbars and Status:** Shows tabs for other files like BankAccount.java, BankAccountRepository.java, and AccountR. A warning icon with '1' is visible.

Premier Micro services - TP

The screenshot shows the IntelliJ IDEA interface with the project tree on the left and the code editor on the right. The project tree for 'bank-account-service' includes '.idea', '.mvn', 'src' (with 'main' and 'java' subfolders), 'resources' (with 'graphql', 'static', and 'templates' subfolders), and 'application.properties'. The code editor displays the 'application.properties' file with the following content:

```
1 spring.datasource.url=jdbc:h2:mem:account-db
2 spring.h2.console.enabled=true
3 server.port:8081
4
```

A tooltip message at the top of the editor area states: "Spring Boot configuration files are supported by IntelliJ IDEA Ultimate". There are three warning icons (yellow exclamation marks) in the status bar at the bottom right of the editor.

8. Dans application.properties

Premier Micro services - TP

```
bank-account-service ~/Downloads/bank-account-service
> .idea
> .mvn
> src
> main
> java
> org.sid.bankaccountservice
> entities
> enums
> repositories
> web
> BankAccountServiceApplication
```

1. Cree un package entities
2. Cree la class BankAccount.java

5. Cree un package repositories
6. Cree une interface
BankAccount.java

```
package org.sid.bankaccountservice.repositories;

import org.sid.bankaccountservice.entities.BankAccount;
import org.springframework.data.jpa.repository.JpaRepository;

5 usages
public interface BankAccountRepository extends JpaRepository<BankAccount, String> {
}
```

3. Cree un package enums
4. Cree un enumerateur AccountType.java

```
1 package org.sid.bankaccountservice.enums;
2
3     5 usages
4     public enum AccountType {
5         1 usage
6             CURRENT_ACCOUNT, SAVING_ACCOUNT
7     }
```

7. Dans BankAccountServiceApplication.java

```
Account.java    BankAccountRepository.java    BankAccountServiceApplication.java    AccountR ...
package org.sid.bankaccountservice;
import ...

@SpringBootApplication
public class BankAccountServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(BankAccountServiceApplication.class, args);
    }

    no usages
    @Bean
    CommandLineRunner start(BankAccountRepository bankAccountRepository){
        return args -> {
            for (int i = 0; i < 10; i++) {
                BankAccount bankAccount=BankAccount.builder()
                    .id(UUID.randomUUID().toString())
                    .type(Math.random()>0.5? AccountType.CURRENT_ACCOUNT: AccountType.SAVING_ACCOUNT)
                    .balance(10000+Math.random()*90000)
                    .createdAt(new Date())
                    .currency("MAD")
                    .build();
                bankAccountRepository.save(bankAccount);
            }
        };
    }
}
```

8. Dans application.properties

```
spring.datasource.url=jdbc:h2:mem:account-db
spring.h2.console.enabled=true
server.port:8081
```

Premier Micro services - TP

← → ⌂ ⓘ localhost:8081/h2-console/login.jsp?jsessionid=795e7dcd50f49ca7985969c86d61f251

English Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded)

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:account-db

User Name: sa

Password:

jdbc:h2:mem:account-db | Auto commit | Max rows: 1000 | Run | Run Selected | Auto complete | Auto select | ?

+ jdbc:h2:mem:account-db
+ BANK_ACCOUNT
+ INFORMATION_SCHEMA
+ Users
① H2 2.1.214 (2022-06-13)

Run Run Selected Auto complete Clear SQL statement: SELECT * FROM BANK_ACCOUNT;

SELECT * FROM BANK_ACCOUNT;

ID	BALANCE	CREATED_AT	CURRENCY	TYPE
9498f090-7ce2-4bc9-beb1-1ecc79948fa4	69614.70913173445	2023-10-14 13:53:08.246	MAD	CURRENT_ACCOUNT
65a52eb0-10bb-486a-a2e7-72554f5e1d48	56639.08394987335	2023-10-14 13:53:08.295	MAD	SAVING_ACCOUNT
6abe91e7-0f44-4763-833c-f5b1486f791b	58079.394117099364	2023-10-14 13:53:08.296	MAD	CURRENT_ACCOUNT
29cc4dce-5346-424a-bdc4-76987de0cc0b	70672.4004732124	2023-10-14 13:53:08.296	MAD	CURRENT_ACCOUNT
aeb54be7-56ce-41d4-b6b6-2b37d062c45e	87569.2962628175	2023-10-14 13:53:08.297	MAD	CURRENT_ACCOUNT
b64546a7-d84e-4cdf-918a-13f2a3aac9f6	57216.202458349355	2023-10-14 13:53:08.298	MAD	SAVING_ACCOUNT
b7d90398-4673-4f57-941d-4422db5d3581	61106.9712515361	2023-10-14 13:53:08.298	MAD	CURRENT_ACCOUNT
00cbd9e4-93d6-443f-a853-bc09e5e61844	78345.74946468501	2023-10-14 13:53:08.299	MAD	CURRENT_ACCOUNT
4e098f30-c4e0-4270-b8b6-37e93d17e91f	29486.6082161305	2023-10-14 13:53:08.299	MAD	CURRENT_ACCOUNT
7d1c6f7b-3993-4be1-83a4-a618ee8c376b	36655.45350772113	2023-10-14 13:53:08.3	MAD	SAVING_ACCOUNT

(10 rows, 12 ms)

Premier Micro services - TP

9. Cree un package web

10. Cree une class AccountRestController

```
11  @RestController
12  public class AccountRestController {
13      3 usages
14      private BankAccountRepository bankAccountRepository;
15      no usages
16      public AccountRestController(BankAccountRepository bankAccountRepository){
17          this.bankAccountRepository = bankAccountRepository;
18      }
19      no usages
20      @GetMapping("/bankAccounts")
21      public List<BankAccount> bankAccounts(){
22          return bankAccountRepository.findAll();
23      }
24      no usages
25      @GetMapping("/bankAccounts/{id}")
26      public BankAccount bankAccounts(@PathVariable String id){
27          return bankAccountRepository.findById(id).orElseThrow(()->new RuntimeException(String.format("Account %s not found",id)));
28      }
29 }
```

Vérification

← → C i localhost:8081/bankAccounts

```
[{"id": "456bdedb-8b59-4b1a-8829-251d3783d9b5", "createdAt": "2023-09-29T20:01:26.080+00:00", "balance": 50074.981073606876, "currency": "MAD", "type": "CURRENT_ACCOUNT"}, {"id": "b2865324-7cc2-4a22-9f50-c5f6f532a754", "createdAt": "2023-09-29T20:01:26.118+00:00", "balance": 16687.121022361574, "currency": "MAD", "type": "SAVING_ACCOUNT"}, {"id": "6be7b9e3-2e2e-4225-b647-6528daldf8a", "createdAt": "2023-09-29T20:01:26.119+00:00", "balance": 86942.57383417788, "currency": "MAD", "type": "CURRENT_ACCOUNT"}, {"id": "1db265de-bee8-4b5c-81c9-9ec9a35ded37", "createdAt": "2023-09-29T20:01:26.120+00:00", "balance": 34248.93033510799, "currency": "MAD", "type": "SAVING_ACCOUNT"}, {"id": "e9c94a5d-f89c-4d97-81ec-37fe39096aa1", "createdAt": "2023-09-29T20:01:26.121+00:00", "balance": 70465.60790547318, "currency": "MAD", "type": "CURRENT_ACCOUNT"}, {"id": "9ee4f7b6-1134-4bc8-99aa-95eaea57564d", "createdAt": "2023-09-29T20:01:26.121+00:00", "balance": 94867.08863560727, "currency": "MAD", "type": "SAVING_ACCOUNT"}, {"id": "11c9dd02-0ddc-4b27-a811-b02ca17896bf", "createdAt": "2023-09-29T20:01:26.122+00:00", "balance": 48426.822542207155, "currency": "MAD", "type": "SAVING_ACCOUNT"}, {"id": "58f229d1-9b51-473b-908d-36b67e5af945", "createdAt": "2023-09-29T20:01:26.123+00:00", "balance": 51149.062826782356, "currency": "MAD", "type": "CURRENT_ACCOUNT"}, {"id": "4d523bab-3da3-43e1-a72b-b620c433efdd", "createdAt": "2023-09-29T20:01:26.123+00:00", "balance": 68597.91223103378, "currency": "MAD", "type": "CURRENT_ACCOUNT"}, {"id": "df5b9c6e-2772-47ce-92eb-0a0e17b4bd0f", "createdAt": "2023-09-29T20:01:26.124+00:00", "balance": 43072.11024543038, "currency": "MAD", "type": "CURRENT_ACCOUNT"}]
```

← → C i localhost:8081/bankAccounts/456bdedb-8b59-4b1a-8829-251d3783d9b5

```
{"id": "456bdedb-8b59-4b1a-8829-251d3783d9b5", "createdAt": "2023-09-29T20:01:26.080+00:00", "balance": 50074.981073606876, "currency": "MAD", "type": "CURRENT_ACCOUNT"}
```

Premier Micro services - TP

11. Ajouter , mettre a jour et supprimer un compte

```
no usages
@GetMapping("/bankAccounts/{id}")
public BankAccount bankAccounts(@PathVariable String id){
    return bankAccountRepository.findById(id).orElseThrow(()->new RuntimeException(String.format("Account %s not found",id)));
}

no usages
@PostMapping("/bankAccounts")
public BankAccount save(@RequestBody BankAccount bankAccount){

    return bankAccountRepository.save(bankAccount);
}

no usages
@PutMapping("/bankAccounts/{id} ")
public BankAccount update(@PathVariable String id, @RequestBody BankAccount bankAccount){
    BankAccount account=bankAccountRepository.findById(id).orElseThrow();
    if ( bankAccount.getBalance()!= null) account.setBalance(bankAccount.getBalance());
    if ( bankAccount.getCreatedAt()!=null)account.setCreatedAt(new Date());
    if ( bankAccount.getType()!=null)account.setType(bankAccount.getType());
    if ( bankAccount.getCurrency()!=null)account.setCurrency(bankAccount.getCurrency());
    return bankAccountRepository.save(account);
}

no usages
@DeleteMapping ("/bankAccounts/{id}")
public void deleteAccount(@PathVariable String id){
    bankAccountRepository.deleteById(id);
}
```

Premier Micro services - TP

1.Sur google chercher: spring boot openapi doc maven

2.Visiter: <https://mvnrepository.com/artifact/org.springdoc/springdoc-openapi-ui>

3.Cliquer sur une version

4.Copy the dependencies

The screenshot shows the Maven Repository website with the URL mvnrepository.com/artifact/org.springdoc/springdoc-openapi-ui/1.7.0. The page displays details for the SpringDoc OpenAPI UI version 1.7.0, including its license (Apache 2.0), tags (spring, openapi, ui, api), and various repository statistics. A prominent feature is a code block at the bottom containing the Maven dependency declaration:

```
<!-- https://mvnrepository.com/artifact/org.springdoc/springdoc-openapi-ui -->
<dependency>
    <groupId>org.springdoc</groupId>
    <artifactId>springdoc-openapi-ui</artifactId>
    <version>1.7.0</version>
</dependency>
```

5.Coller dans pom.xml

The screenshot shows an IDE interface with the file `pom.xml` open. The dependency code from the previous screenshot has been pasted into the `pom.xml` file, specifically within the `<dependencies>` section. The code is highlighted with a red border. The rest of the `pom.xml` file includes sections for `<build>`, `<plugins>`, and `<plugin>`.

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webflux</artifactId>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.springframework.graphql</groupId>
    <artifactId>spring-graphql-test</artifactId>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.springdoc</groupId>
    <artifactId>springdoc-openapi-ui</artifactId>
    <version>1.7.0</version>
</dependency>
```

Premier Micro services - TP

<http://localhost:8081/swagger-ui.html>

The screenshot shows the Swagger UI interface for a microservice. At the top, the URL is `localhost:8081/swagger-ui/index.html`. The navigation bar includes the Swagger logo, the path `/v3/api-docs`, and a green "Explore" button. Below the navigation, the title "OpenAPI definition" is displayed with a version indicator "v0 OAS3". A link to "`/v3/api-docs`" is also present. The main content area is titled "account-rest-controller". It lists several API endpoints:

- PUT** `/bankAccounts/{id}` (highlighted in orange)
- GET** `/bankAccounts` (highlighted in blue)
- POST** `/bankAccounts` (highlighted in green)
- GET** `/bankAccounts/{id}` (highlighted in light blue)
- DELETE** `/bankAccounts/{id}` (highlighted in red)

Below the endpoints, there is a section titled "Schemas" containing a link to "BankAccount >".

Premier Micro services - TP

GET /bankAccounts

Cancel

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8081/bankAccounts' \
  -H 'accept: */*' 
```

Request URL

```
http://localhost:8081/bankAccounts
```

Server response

Code Details

Code	Details						
200	<p>Response body</p> <pre>[{ "id": "35fd829d-f4f8-49b4-a86a-6724a2e616d1", "createdAt": "2023-10-08T14:21:22.625+00:00", "balance": 27892.51148573909, "currency": "MAD", "type": "CURRENT_ACCOUNT" }, { "id": "d16e80d0-aaa3-4dfd-9f91-d2eb8db18bcd", "createdAt": "2023-10-08T14:21:22.662+00:00", "balance": 79806.47014265438, "currency": "MAD", "type": "SAVING_ACCOUNT" }, { "id": "8f25b450-bd9b-4a6d-b311-35aa4963fc82", "createdAt": "2023-10-08T14:21:22.663+00:00", "balance": 27561.280272489716, "currency": "MAD", "type": "SAVING_ACCOUNT" }, { "id": "52ee2f51-c0de-4c65-a12c-89f1d6d49250", "createdAt": "2023-10-08T14:21:22.664+00:00", "balance": 27561.280272489716, "currency": "MAD", "type": "SAVING_ACCOUNT" }]  </pre> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Sun, 08 Oct 2023 14:22:25 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre> <p>Responses</p> <table border="1"><thead><tr><th>Code</th><th>Description</th><th>Links</th></tr></thead><tbody><tr><td>200</td><td>OK</td><td>No links</td></tr></tbody></table>	Code	Description	Links	200	OK	No links
Code	Description	Links					
200	OK	No links					

Premier Micro services - TP

Name Description

id required
string
(path)

35fd829d-f4f8-49b4-a86a-6724a2e616d1

Execute **Clear**

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8081/bankAccounts/35fd829d-f4f8-49b4-a86a-6724a2e616d1' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:8081/bankAccounts/35fd829d-f4f8-49b4-a86a-6724a2e616d1
```

Server response

Code	Details						
200	Response body <pre>{ "id": "35fd829d-f4f8-49b4-a86a-6724a2e616d1", "createdAt": "2023-10-08T14:21:22.625+00:00", "balance": 27892.51148573909, "currency": "MAD", "type": "CURRENT_ACCOUNT" }</pre> Response headers <pre>connection: keep-alive content-type: application/json date: Sun, 08 Oct 2023 14:22:35 GMT keep-alive: timeout=60 transfer-encoding: chunked</pre> <p>Responses</p> <table border="1"><thead><tr><th>Code</th><th>Description</th><th>Links</th></tr></thead><tbody><tr><td>200</td><td>OK</td><td>No links</td></tr></tbody></table> <p>Media type</p> <p>*/*</p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "id": "string", "createdAt": "2023-10-08T14:22:35.749Z", "balance": 0, "currency": "string", "type": "CURRENT_ACCOUNT" }</pre>	Code	Description	Links	200	OK	No links
Code	Description	Links					
200	OK	No links					

Premier Micro services - TP

POST /bankAccounts

Cancel Reset

Parameters

No parameters

Request body required

application/json

```
{  
    "id": "string",  
    "createdAt": "2023-10-08T14:36:14.686Z",  
    "balance": 100000,  
    "currency": "EUR",  
    "type": "CURRENT_ACCOUNT"  
}
```

Execute Clear

Responses

Curl

```
curl -X 'POST' \  
  'http://localhost:8081/bankAccounts' \  
  -H 'accept: */*' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "id": "string",  
    "createdAt": "2023-10-08T14:36:14.686Z",  
    "balance": 100000,  
    "currency": "EUR",  
    "type": "CURRENT_ACCOUNT"  
}'
```

Request URL

http://localhost:8081/bankAccounts

Server response

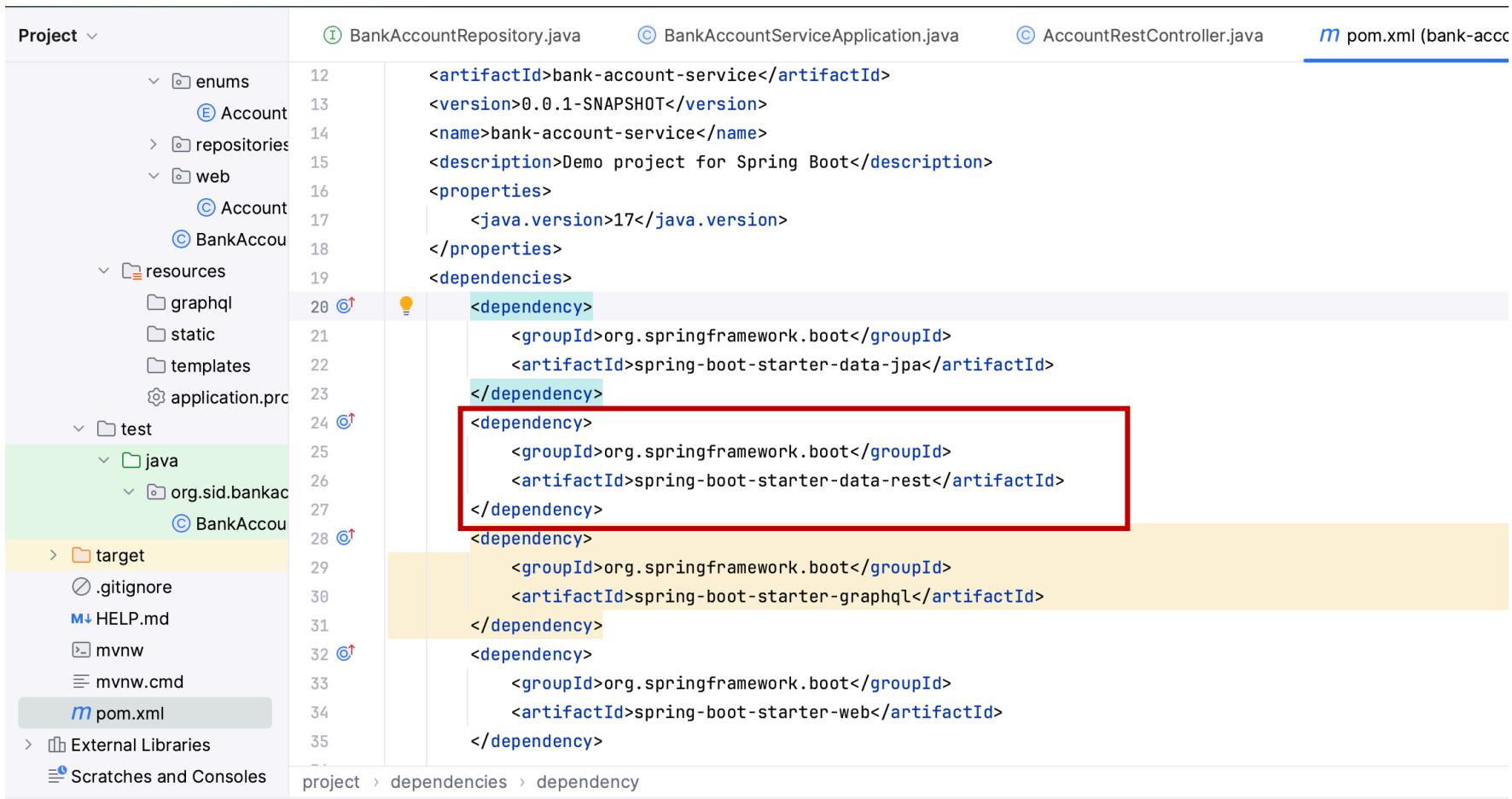
Code	Details
200	Response body

```
{  
    "id": "ac164eea-401a-4140-9d4a-834b8d69bb61",  
    "createdAt": "2023-10-08T14:36:14.686+00:00",  
    "balance": 100000,  
    "currency": "EUR",  
    "type": "CURRENT_ACCOUNT"  
}
```

Download

Premier Micro services - TP

Ajouter les dépendances de Spring Data Rest



The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays the `pom.xml` file for a Spring Boot application named "bank-account-service". The dependency section is highlighted with a red box.

```
<artifactId>bank-account-service</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>bank-account-service</name>
<description>Demo project for Spring Boot</description>
<properties>
    <java.version>17</java.version>
</properties>
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-rest</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-graphql</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

```

Premier Micro services - TP

Utiliser Spring Data REST

Dans BankAccountRepository ajouter @RepositoryRestResource

```
① BankAccountRepository.java × ② BankAccountServiceApplication.java ③ AccountRestController.java  
1 package org.sid.bankaccountservice.repositories;  
2  
3 import org.sid.bankaccountservice.entities.BankAccount;  
4 import org.springframework.data.jpa.repository.JpaRepository;  
5 import org.springframework.data.rest.core.annotation.RepositoryRestResource;  
6 import org.springframework.data.rest.webmvc.RepositoryRestController;  
7  
8 5 usages  
9 @RepositoryRestResource  
10 public interface BankAccountRepository extends JpaRepository <BankAccount, String> {  
11 }
```

AccountRestController

Ajouter @RequestMapping (""/api"") pour pouvoir tester

```
① BankAccountRepository.java ② BankAccountServiceApplication.java ③ AccountRestController.java  
1 import java.util.Date;  
2 import java.util.List;  
3 import java.util.UUID;  
4  
5 no usages  
6  
7 @RestController  
8 @RequestMapping("/api") ←  
9 public class AccountRestController {  
10     7 usages  
11     private BankAccountRepository bankAccountRepository;  
12     no usages  
13     public AccountRestController(BankAccountRepository bankAccountRepository){  
14         this.bankAccountRepository = bankAccountRepository;  
15     }  
16 }  
17 }
```

Pour acceder a ce webservice il faut faire /api/bankAccounts

Premier Micro services - TP

Utiliser Spring Data REST

```
← → ⌂ ⓘ localhost:8081/bankAccounts

{
  "_embedded" : {
    "bankAccounts" : [ {
      "createdAt" : "2023-10-08T15:25:45.609+00:00",
      "balance" : 46408.42204642685,
      "currency" : "MAD",
      "type" : "SAVING_ACCOUNT",
      "_links" : {
        "self" : {
          "href" : "http://localhost:8081/bankAccounts/498b6d6d-c7d8-4099-af9c-a8330bee72a1"
        },
        "bankAccount" : {
          "href" : "http://localhost:8081/bankAccounts/498b6d6d-c7d8-4099-af9c-a8330bee72a1"
        }
      }
    }, {
      "createdAt" : "2023-10-08T15:25:45.645+00:00",
      "balance" : 88863.51659133962,
      "currency" : "MAD",
      "type" : "CURRENT_ACCOUNT",
      "_links" : {
        "self" : {
          "href" : "http://localhost:8081/bankAccounts/0afa5b22-13fd-4c12-855c-d1c5e8004376"
        },
        "bankAccount" : {
          "href" : "http://localhost:8081/bankAccounts/0afa5b22-13fd-4c12-855c-d1c5e8004376"
        }
      }
    }, {
      "createdAt" : "2023-10-08T15:25:45.646+00:00",
      "balance" : 90816.55779492216,
      "currency" : "MAD",
      "type" : "CURRENT_ACCOUNT",
      "_links" : {
        "self" : {
          "href" : "http://localhost:8081/bankAccounts/19669250-7ad0-4dec-848d-9f0618cf5b2f"
        },
        "bankAccount" : {
          "href" : "http://localhost:8081/bankAccounts/19669250-7ad0-4dec-848d-9f0618cf5b2f"
        }
      }
    }, {
      "createdAt" : "2023-10-08T15:25:45.647+00:00",
      "balance" : 62683.74530760303,
      "currency" : "MAD",
      "type" : "SAVING_ACCOUNT",
      "_links" : {
        "self" : {
          "href" : "http://localhost:8081/bankAccounts/e3fc2ebc-6d98-4b89-b268-a872575d0e6e"
        },
        "bankAccount" : {
          "href" : "http://localhost:8081/bankAccounts/e3fc2ebc-6d98-4b89-b268-a872575d0e6e"
        }
      }
    }, {
      "createdAt" : "2023-10-08T15:25:45.647+00:00",
      "balance" : 16139.076790230076,
      "currency" : "MAD",
      "type" : "CURRENT_ACCOUNT",
      "_links" : {
        "self" : {
          "href" : "http://localhost:8081/bankAccounts/e3fc2ebc-6d98-4b89-b268-a872575d0e6e"
        }
      }
    }
  ]
}
```

Premier Micro services - TP

bankAccount.java

BankAccountRepository.java

AccountType.java

BankAccountServiceApplication.j

```
1 package org.sid.bankaccountservice.repositories;
2
3 import org.sid.bankaccountservice.entities.BankAccount;
4 import org.sid.bankaccountservice.enums.AccountType;
5 import org.springframework.data.jpa.repository.JpaRepository;
6 import org.springframework.data.rest.core.annotation.RepositoryRestResource;
7 import org.springframework.data.rest.webmvc.RepositoryRestController;
8
9 import java.util.List;
10
11 5 usages
12 @RepositoryRestResource
13 public interface BankAccountRepository extends JpaRepository<BankAccount, String> {
14     no usages
15     List<BankAccount> findByType(AccountType type);
16 }
```

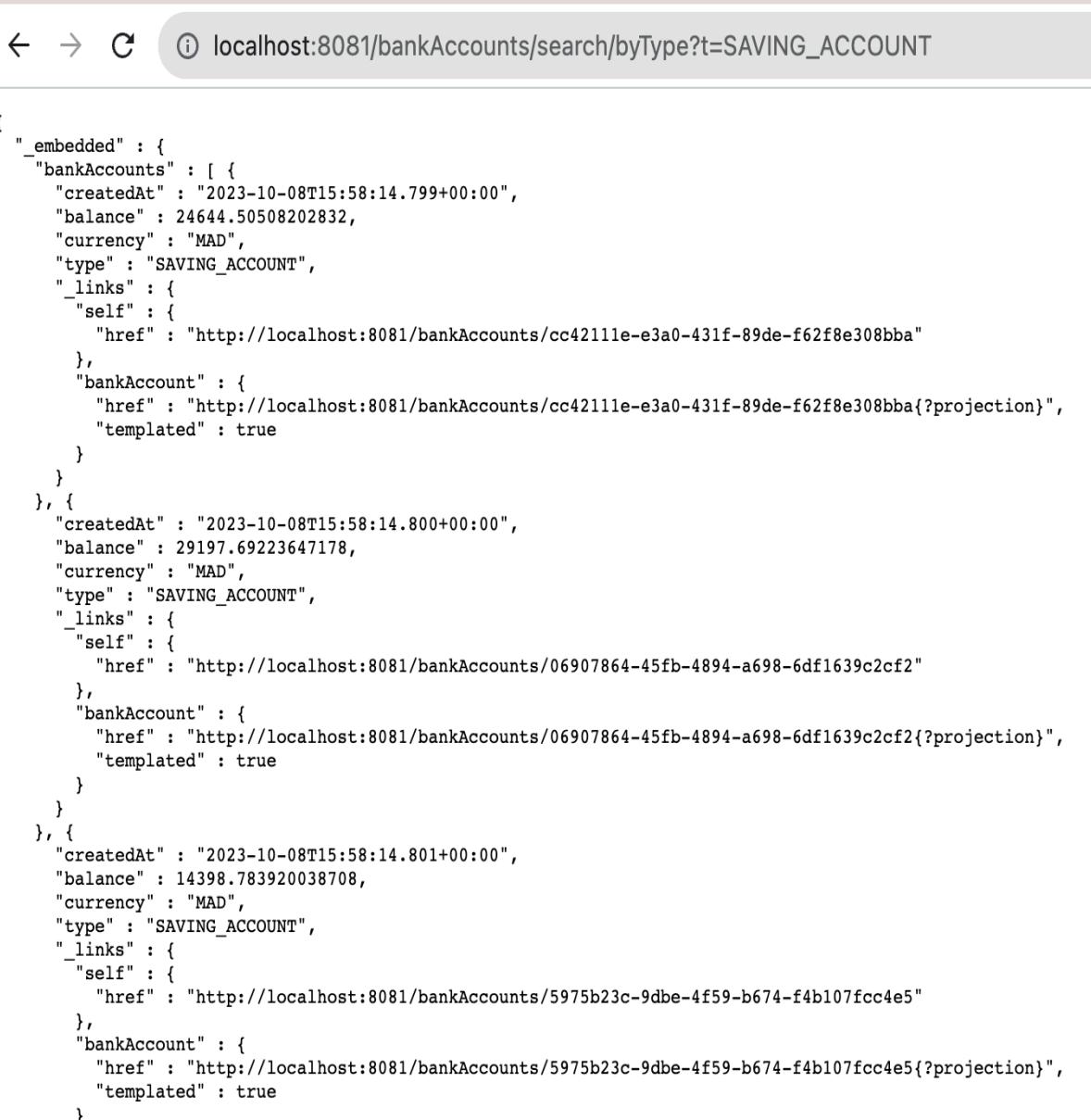
localhost:8081/bankAccounts/search/findByType?type=SAVING_ACCOUNT

```
{
    "_embedded" : {
        "bankAccounts" : [ {
            "createdAt" : "2023-10-08T15:37:02.212+00:00",
            "balance" : 61478.87686141819,
            "currency" : "MAD",
            "type" : "SAVING_ACCOUNT",
            "_links" : {
                "self" : {
                    "href" : "http://localhost:8081/bankAccounts/bf2b9228-f2c3-4b33-bef7-008f1da3b4a8"
                },
                "bankAccount" : {
                    "href" : "http://localhost:8081/bankAccounts/bf2b9228-f2c3-4b33-bef7-008f1da3b4a8"
                }
            }
        }, {
            "createdAt" : "2023-10-08T15:37:02.252+00:00",
            "balance" : 36404.06841520627,
            "currency" : "MAD",
            "type" : "SAVING_ACCOUNT",
            "_links" : {
                "self" : {
                    "href" : "http://localhost:8081/bankAccounts/4cc06cb8-9548-4a65-9448-5020beb581c6"
                },
                "bankAccount" : {
                    "href" : "http://localhost:8081/bankAccounts/4cc06cb8-9548-4a65-9448-5020beb581c6"
                }
            }
        }, {
            "createdAt" : "2023-10-08T15:37:02.254+00:00",
            "balance" : 15695.553271278859,
            "currency" : "MAD",
            "type" : "SAVING_ACCOUNT",
            "_links" : {
                "self" : {
                    "href" : "http://localhost:8081/bankAccounts/b629213e-2791-4213-ab2b-168d65f1cab1"
                },
                "bankAccount" : {
                    "href" : "http://localhost:8081/bankAccounts/b629213e-2791-4213-ab2b-168d65f1cab1"
                }
            }
        }, {
            "createdAt" : "2023-10-08T15:37:02.254+00:00",
            "balance" : 27393.75262658767,
            "currency" : "MAD",
            "type" : "SAVING_ACCOUNT",
            "_links" : {
                "self" : {
                    "href" : "http://localhost:8081/bankAccounts/aa27ccbb-7243-4a54-be87-1da8c0999d00"
                },
                "bankAccount" : {
                    "href" : "http://localhost:8081/bankAccounts/aa27ccbb-7243-4a54-be87-1da8c0999d00"
                }
            }
        }, {
            "createdAt" : "2023-10-08T15:37:02.257+00:00",
            "balance" : 77503.14522834014,
            "currency" : "MAD"
        }
    ]
}
```

Premier Micro services - TP

Normalement il ne faut pas appeler les méthodes par leur nom
-> On utilise les annotations

```
13 @RepositoryRestResource
14 public interface BankAccountRepository extends JpaRepository<BankAccou
15     no usages
16     @RestResource(path="/byType")
17     List<BankAccount> findByType(@Param("t") AccountType type);
18
19 }
```



```
{
  "_embedded" : {
    "bankAccounts" : [ {
      "createdAt" : "2023-10-08T15:58:14.799+00:00",
      "balance" : 24644.50508202832,
      "currency" : "MAD",
      "type" : "SAVING_ACCOUNT",
      "_links" : {
        "self" : {
          "href" : "http://localhost:8081/bankAccounts/cc42111e-e3a0-431f-89de-f62f8e308bba"
        },
        "bankAccount" : {
          "href" : "http://localhost:8081/bankAccounts/cc42111e-e3a0-431f-89de-f62f8e308bba{?projection}",
          "templated" : true
        }
      }
    }, {
      "createdAt" : "2023-10-08T15:58:14.800+00:00",
      "balance" : 29197.69223647178,
      "currency" : "MAD",
      "type" : "SAVING_ACCOUNT",
      "_links" : {
        "self" : {
          "href" : "http://localhost:8081/bankAccounts/06907864-45fb-4894-a698-6df1639c2cf2"
        },
        "bankAccount" : {
          "href" : "http://localhost:8081/bankAccounts/06907864-45fb-4894-a698-6df1639c2cf2{?projection}",
          "templated" : true
        }
      }
    }, {
      "createdAt" : "2023-10-08T15:58:14.801+00:00",
      "balance" : 14398.783920038708,
      "currency" : "MAD",
      "type" : "SAVING_ACCOUNT",
      "_links" : {
        "self" : {
          "href" : "http://localhost:8081/bankAccounts/5975b23c-9dbe-4f59-b674-f4b107fcc4e5"
        },
        "bankAccount" : {
          "href" : "http://localhost:8081/bankAccounts/5975b23c-9dbe-4f59-b674-f4b107fcc4e5{?projection}",
          "templated" : true
        }
      }
    } ]
}
```

Premier Micro services - TP

Cree une projection nomé p1:

Cree une interface AccountProjection dans entities

The screenshot shows the project structure in the left sidebar under 'src'. It includes 'main', 'java' (with subfolders 'org.sid.bankaccountservice' containing 'entities', 'enums', 'repositories', and 'web'), 'resources', 'test', and 'target'. In the 'BankAccount.java' code editor, the cursor is at line 24, which contains a closing brace '}' for the class definition. A tooltip 'New Java Class' is displayed over the cursor, listing options like '@Entity', '@Data', '@NoArgsConstructor', 'public class', '@Id', 'private', 'private Class', 'private Interface', 'private Record', '@Enum', 'private Annotation', and 'private'. The code itself starts with package declarations and imports for Lombok annotations, followed by the class definition.

```
1 package org.sid.bankaccountservice.entities;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Builder;
5 import lombok.Data;
6 import lombok.NoArgsConstructor;
7 import org.sid.bankaccountservice.enums.AccountType;
8
9 import javax.persistence.Entity;
10 import javax.persistence.Enumerated;
11 import javax.persistence.Id;
12 import java.util.Date;
13
14 usages
15 @Entity
16 @Data @NoArgsConstructor
17 public class BankAccount {
18     @Id
19     private Long id;
20     private Class<BankAccount> bankAccount;
21     private Interface<BankAccount> interfaceBankAccount;
22     private Record<BankAccount> recordBankAccount;
23     private Enum<AccountType> enumBankAccount;
24 }
```

```
6 @Projection(types = BankAccount.class, name="p1")
7 public interface AccountProjection {
8     no usages
9     public String getID();
10    no usages
11    public AccountType getType();
12 }
```

The screenshot shows a browser window with the URL 'localhost:8081/bankAccounts?projection=p1' highlighted with a red box. The response is a JSON array of three bank account objects, each with an '_embedded' key pointing to another JSON object containing the account details and a 'bankAccount' key with a templated href.

```
{
  "_embedded": [
    "bankAccounts": [
      {
        "id": "efc23674-b953-4ad3-9d48-48df1dd3dccb",
        "type": "CURRENT_ACCOUNT",
        "links": {
          "self": {
            "href": "http://localhost:8081/bankAccounts/efc23674-b953-4ad3-9d48-48df1dd3dccb"
          },
          "bankAccount": {
            "href": "http://localhost:8081/bankAccounts/efc23674-b953-4ad3-9d48-48df1dd3dccb{?projection}",
            "templated": true
          }
        }
      },
      {
        "id": "1e41f776-d820-498d-8504-5af61cb82fb0",
        "type": "SAVING_ACCOUNT",
        "links": {
          "self": {
            "href": "http://localhost:8081/bankAccounts/1e41f776-d820-498d-8504-5af61cb82fb0"
          },
          "bankAccount": {
            "href": "http://localhost:8081/bankAccounts/1e41f776-d820-498d-8504-5af61cb82fb0{?projection}",
            "templated": true
          }
        }
      },
      {
        "id": "156875e2-a30c-49d0-8bc0-5ef75209d619",
        "type": "SAVING_ACCOUNT",
        "links": {
          "self": {
            "href": "http://localhost:8081/bankAccounts/156875e2-a30c-49d0-8bc0-5ef75209d619"
          },
          "bankAccount": {
            "href": "http://localhost:8081/bankAccounts/156875e2-a30c-49d0-8bc0-5ef75209d619{?projection}",
            "templated": true
          }
        }
      }
    ]
  ]
}
```

Premier Micro services - TP

1.Cree la couche service: package service

2.Cree interface AccountService

```
public interface AccountService {  
    1 usage 1 implementation  
    BankAccountResponseDTO addAccount(BankAccountRequestDTO bankAccountDTO);  
}
```

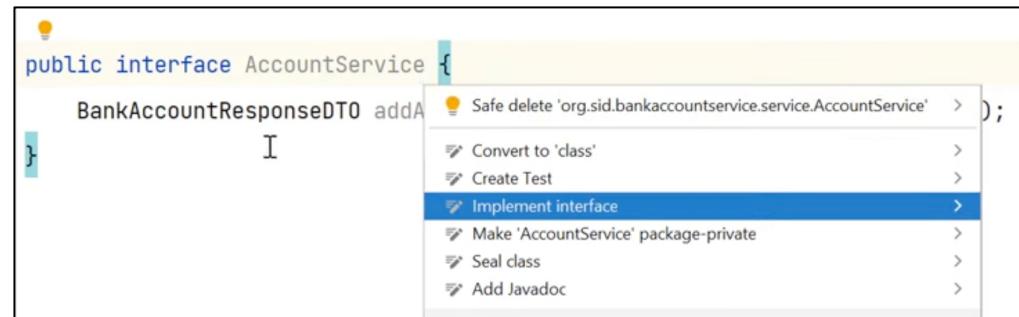
3.Cree un package DTO

4. Cree les classe BankAccountResponseDTO et BankAccountResquestDTO

```
@Data @AllArgsConstructor @NoArgsConstructor @Builder  
public class BankAccountRequestDTO {  
    private Double balance;  
    private String currency ;  
    private AccountType type;  
}
```

```
@Data @NoArgsConstructor @AllArgsConstructor @Builder  
public class BankAccountResponseDTO {  
    private String id;  
    private Date createdAt;  
    private Double balance;  
    private String currency ;  
    private AccountType type;  
}
```

5.Cree une implementation pour l'interface AccountService



```
@Service  
@Transactional  
public class AccountServiceImpl implements AccountService {  
    1 usage  
    @Autowired  
    private BankAccountRepository bankAccountRepository;  
    1 usage  
    @Autowired  
    private AccountMapper accountMapper;  
    1 usage  
    @Override  
    public BankAccountResponseDTO addAccount(BankAccountRequestDTO bankAccountDTO) {  
        //mapping  
        BankAccount bankAccount = BankAccount.builder()  
            .id(UUID.randomUUID().toString())  
            .createdAt(new Date())  
            .balance(bankAccountDTO.getBalance())  
            .type(bankAccountDTO.getType())  
            .currency(bankAccountDTO.getCurrency())  
            .build();  
        //code metier -> une ligne  
        BankAccount savedBankAccount = bankAccountRepository.save(bankAccount);  
        BankAccountResponseDTO bankAccountResponseDTO = accountMapper.fromBankAccount(savedBankAccount);  
        return bankAccountResponseDTO;  
    }  
}
```

Premier Micro services - TP

6.Modifier AccountRestController

Apres

```
@PostMapping("/bankAccounts")
public BankAccountResponseDTO save(@RequestBody BankAccountRequestDTO requestDTO ){
    return accountService.addAccount(requestDTO);
}
```

Avant

```
@PostMapping("/bankAccounts")
public BankAccount save(@RequestBody BankAccount bankAccount){
    if ( bankAccount.getId() != null) bankAccount.setId(UUID.randomUUID().toString());
    return bankAccountRepository.save(bankAccount);
}
```

N'oublier pas d'ajouter *private AccountService accountService; et de faire l'injection (@Autowired)*

7.Cree un package mappers

8.Cree la class AccountMapper

```
@Component
public class AccountMapper {
    1 usage
    public BankAccountResponseDTO fromBankAccount(BankAccount bankAccount){
        BankAccountResponseDTO bankAccountResponseDTO = new BankAccountResponseDTO();
        BeanUtils.copyProperties(bankAccount, bankAccountResponseDTO);
        return bankAccountResponseDTO;
    }
}
```

N'oublier pas d'ajouter ->
Au fichier:
AccountServiceImpl.java
AccountRestController.java

```
@Autowired
private AccountMapper accountMapper;
```

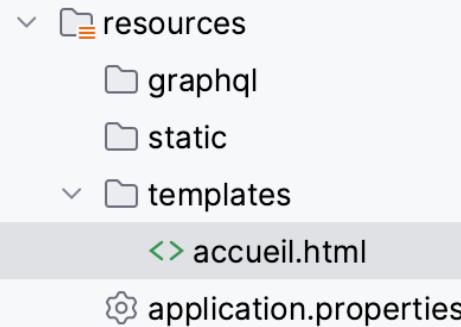
Premier Micro services - TP

Gestion des composants de la Vue

4.accueil.html

1.Ajouter la dependance dans pom.xml

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
```

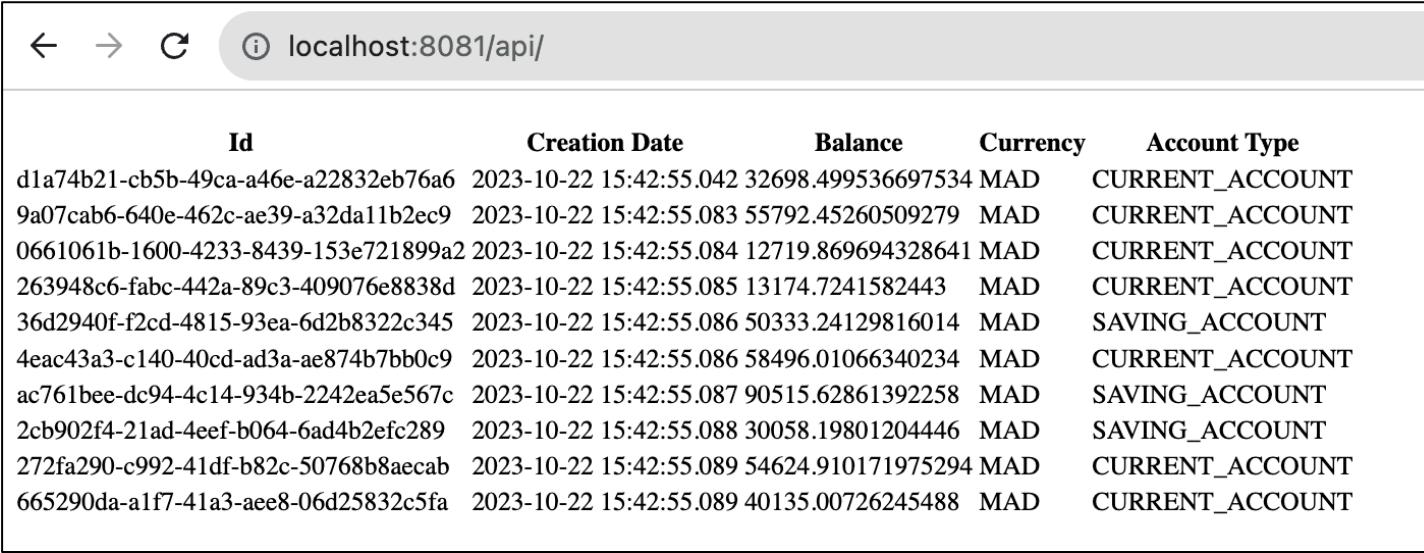


2.Ajouter accueil.html comme indique

3.Ajouter ce code dans AccountRestController.java

```
no usages
@GetMapping("/")
public ModelAndView index (Model model) {
    ModelAndView modelAndView = new ModelAndView();
    Iterable<BankAccount> listAccounts = bankAccountRepository.findAll();
    model.addAttribute( attributeName: "BankAccount", listAccounts);
    modelAndView.setViewName("accueil.html");
    return modelAndView;
}
```

```
<!DOCTYPE html>
<html lang="eng" xmlns:th="http://www.w3.org/1999/xhtml"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<head>
    <meta charset="UTF-8">
    <title>Liste des Comtes</title>
</head>
<body>
    <h2 class="h2">Liste des Comtes</h2>
    <table>
        <thead>
            <tr>
                <th>Id</th>
                <th>Creation Date</th>
                <th>Balance</th>
                <th>Currency</th>
                <th>Account Type</th>
            </tr>
        </thead>
        <tbody>
            <tr th:each="BankAccount: ${BankAccount}">
                <td><span th:text="${BankAccount.id}"> Id </span></td>
                <td><span th:text="${BankAccount.createdAt}"> Creation Date </span></td>
                <td><span th:text="${BankAccount.balance}"> Balance </span></td>
                <td><span th:text="${BankAccount.currency}"> Currency </span></td>
                <td><span th:text="${BankAccount.type}"> Account Type</span></td>
            </tr>
        </tbody>
    </table>
</body>
</html>
```



A screenshot of a web browser window displaying a table of account data. The URL in the address bar is "localhost:8081/api/". The table has columns for Id, Creation Date, Balance, Currency, and Account Type. The data shows ten entries of account information.

Id	Creation Date	Balance	Currency	Account Type
d1a74b21-cb5b-49ca-a46e-a22832eb76a6	2023-10-22 15:42:55.042	32698.499536697534	MAD	CURRENT_ACCOUNT
9a07cab6-640e-462c-ae39-a32da11b2ec9	2023-10-22 15:42:55.083	55792.45260509279	MAD	CURRENT_ACCOUNT
0661061b-1600-4233-8439-153e721899a2	2023-10-22 15:42:55.084	12719.869694328641	MAD	CURRENT_ACCOUNT
263948c6-fabc-442a-89c3-409076e8838d	2023-10-22 15:42:55.085	13174.7241582443	MAD	CURRENT_ACCOUNT
36d2940f-f2cd-4815-93ea-6d2b8322c345	2023-10-22 15:42:55.086	50333.24129816014	MAD	SAVING_ACCOUNT
4eac43a3-c140-40cd-ad3a-ae874b7bb0c9	2023-10-22 15:42:55.086	58496.01066340234	MAD	CURRENT_ACCOUNT
ac761bee-dc94-4c14-934b-2242ea5e567c	2023-10-22 15:42:55.087	90515.62861392258	MAD	SAVING_ACCOUNT
2cb902f4-21ad-4eef-b064-6ad4b2efc289	2023-10-22 15:42:55.088	30058.19801204446	MAD	SAVING_ACCOUNT
272fa290-c992-41df-b82c-50768b8aecab	2023-10-22 15:42:55.089	54624.910171975294	MAD	CURRENT_ACCOUNT
665290da-a1f7-41a3-aee8-06d25832c5fa	2023-10-22 15:42:55.089	40135.00726245488	MAD	CURRENT_ACCOUNT

- Les instructions **th:if** et **th:each** qui permettent respectivement d'implémenter une condition et une itération, fonctionnalités non présentes en HTML. Thymeleaf me fournit donc la capacité d'écrire **un code HTML dynamique**.
- La syntaxe **\${nom de l'attribut}** permet d'accéder à un objet placé comme attribut dans le **Model**.
- Notons également que Thymeleaf comprend la programmation objet, et que la syntaxe **\${\$objet.attribut}** fonctionne.