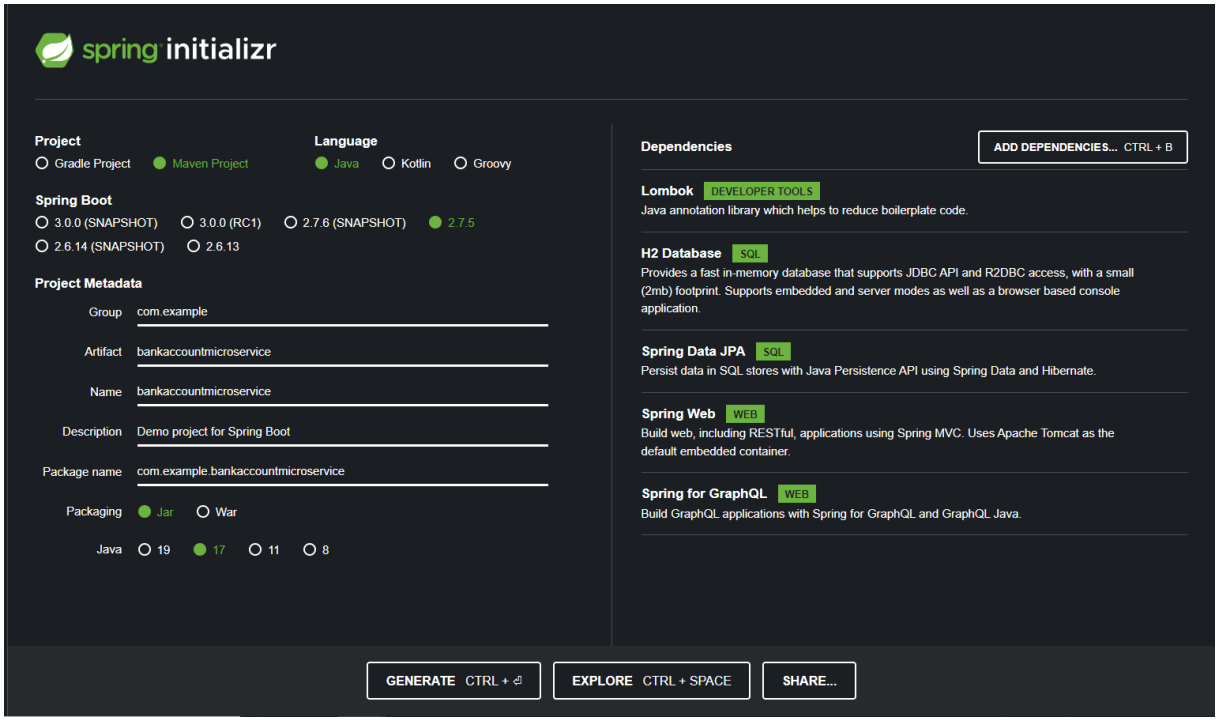
# Activité Pratique N° 2 : Développement du Premier Micro-service

***Objectif :*** Créer un micro-service qui permet de gérer des comptes bancaires en utilisant RestApi et GraphQL.

1. Créer un projet Spring Boot avec les dépendances Web, Spring Data JPA, H2, Lombok




2. Créer l'entité JPA Compte

```java
package com.example.bankaccountmicroservice.entities;

import com.example.bankaccountmicroservice.enums.AccountType;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

import javax.persistence.*;
import java.util.Date;
@Entity
@Data @AllArgsConstructor @NoArgsConstructor @Builder
public class BankAccount {
    @Id
    private String id;
    private Date createdAt;
    private Double balance;
    private String currency;
    @Enumerated(EnumType.STRING)
    private AccountType type;
    @ManyToOne
    private Customer customer;
}
```

3. Créer l'interface CompteRepository basée sur Spring Data

```java
package com.example.bankaccountmicroservice.repositories;

import com.example.bankaccountmicroservice.entities.BankAccount;
import com.example.bankaccountmicroservice.enums.AccountType;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.repository.query.Param;
import org.springframework.data.rest.core.annotation.RepositoryRestResource;
import org.springframework.data.rest.core.annotation.RestResource;

import java.util.List;

@RepositoryRestResource
public interface BankAccountRepository extends JpaRepository<BankAccount, String> {
    @RestResource(path = "/byType")
    List<BankAccount> findByType(@Param("t") AccountType type);
}
```

4. Tester la couche DAO

SELECT * FROM BANK_ACCOUNT;

| ID | BALANCE | CREATED_AT | CURRENCY | TYPE |
|---|---|---|---|---|
| c525853a-692b-40b1-945c-9e0a8d236c5d | 84439.14046409412 | 2022-10-26 19:23:00.224 | MAD | SAVING_ACCOUNT |
| 562bdc64-3c2b-4527-af5a-58d70062246e | 56507.849832254695 | 2022-10-26 19:23:00.281 | MAD | CURRENT_ACCOUNT |
| 504f5f58-2147-4b72-907c-c684bf19fd9c | 86184.41992312111 | 2022-10-26 19:23:00.282 | MAD | SAVING_ACCOUNT |
| 1532704b-e934-4664-831c-b6c9e5d12501 | 16267.71155118684 | 2022-10-26 19:23:00.283 | MAD | SAVING_ACCOUNT |
| 8c6f17fc-36a3-473f-92b5-6f1c35228269 | 79201.43503068989 | 2022-10-26 19:23:00.284 | MAD | CURRENT_ACCOUNT |
| c6180894-dac5-4663-a2f0-a9901f0847f1 | 88073.11587437837 | 2022-10-26 19:23:00.285 | MAD | SAVING_ACCOUNT |
| 58d77ddb-25e2-4435-8723-290098be5062 | 76064.2559860494 | 2022-10-26 19:23:00.285 | MAD | SAVING_ACCOUNT |
| 5320f29b-30af-46d1-8cac-987bdee68a73 | 13757.072441256523 | 2022-10-26 19:23:00.286 | MAD | SAVING_ACCOUNT |
| f7cb52ef-0a67-49a2-b2a9-93760293d4e5 | 48670.175320589624 | 2022-10-26 19:23:00.287 | MAD | CURRENT_ACCOUNT |

(9 rows, 5 ms)

5. Créer le Web service Restfull qui permet de gérer des comptes

```java
@GetMapping("/bankAccounts")
public List<BankAccount> bankAccounts() { return bankAccountRepository.findAll(); }

@GetMapping("/bankAccounts/{id}")
public BankAccount bankAccount(@PathVariable String id) {
    return bankAccountRepository.findById(id).orElseThrow(() -> new RuntimeException(String.format("Accound %s not found", id)));
}

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

@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);
}

@DeleteMapping("/bankAccounts/{id}")
public void deleteAccount(@PathVariable String id) { bankAccountRepository.deleteById(id); }
```

6. Tester le web micro-service en utilisant un client REST comme Postman

GET http://localhost:8081/bankAccounts

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON


```
[
    {
        "id": "561ad79f-35b1-4561-99fb-b5493297e862",
        "createdAt": "2022-10-27T18:34:33.594+00:00",
        "balance": 67470.7378415584,
        "currency": "MAD",
        "type": "CURRENT_ACCOUNT"
    },
    {
        "id": "35f65323-79df-4b86-92e9-2c21d1fe2c95",
        "createdAt": "2022-10-27T18:34:33.685+00:00",
        "balance": 65787.89405249021,
        "currency": "MAD",
        "type": "CURRENT_ACCOUNT"
    },
    {
        "id": "3b2adcb1-623e-4b50-96b1-189f59859118",
        "createdAt": "2022-10-27T18:34:33.686+00:00",
        "balance": 45827.970948549286,
        "currency": "MAD",
        "type": "CURRENT_ACCOUNT"
    },
```


*Figure 1 : Affichage des comptes*

GET http://localhost:8081/bankAccounts/561ad79f-35b1-4561-99fb-b5493297e862

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON


```
{
    "id": "561ad79f-35b1-4561-99fb-b5493297e862",
    "createdAt": "2022-10-27T18:34:33.594+00:00",
    "balance": 67470.7378415584,
    "currency": "MAD",
    "type": "CURRENT_ACCOUNT"
}
```


*Figure 2 : Affichage d'un compte par son identifiant*

POST http://localhost:8081/bankAccounts

Params Authorization Headers (9) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary Graph

```
1 {
2     "balance" : 8000,
3     "type" : "SAVING_ACCOUNT",
4     "currency" : "EUR"
5 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2     "id": "ca26bee4-2c34-4bb5-9033-df2d3ce2052c",
3     "createdAt": null,
4     "balance": 8000.0,
5     "currency": "EUR",
6     "type": "SAVING_ACCOUNT"
7 }
```

*Figure 3 : Ajouter un compte*

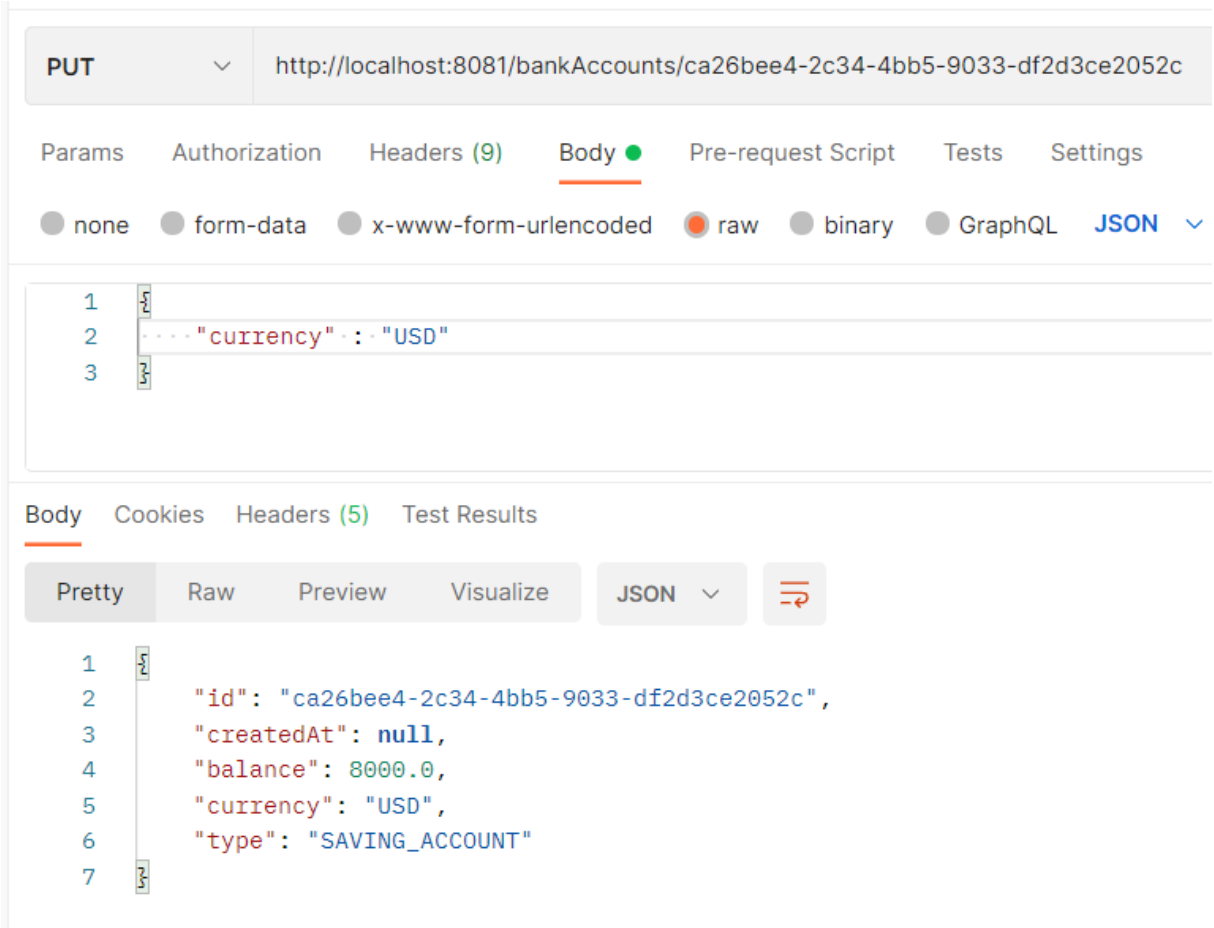



*Figure 4 : Modifier un compte*

7. Générer et tester la documentation Swagger de des API Rest du Web service

Swagger
Supported by SMARTBEAR

/v3/api-docs

# OpenAPI definition v0 OAS3

/v3/api-docs

Servers

http://localhost:8081 - Generated server url

## account-rest-controller

| | |
|---|---|
| GET | /bankAccounts/{id} |
| PUT | /bankAccounts/{id} |
| DELETE | /bankAccounts/{id} |
| GET | /bankAccounts |
| POST | /bankAccounts |

```
{
    "openapi": "3.0.1",
    "info": {
        "title": "OpenAPI definition",
        "version": "v0"
    },
    "servers": [
        {
            "url": "http://localhost:8081",
            "description": "Generated server url"
        }
    ],
    "paths": { … }, // 2 items
    "components": { … } // 1 item
}
```


8. Exposer une API Restful en utilisant Spring Data Rest en exploitant des projections

localhost:8081/bankAccounts


```
{
    "_embedded": {
        "bankAccounts": [
            {
                "createdAt": "2022-10-27T19:22:14.714+00:00",
                "balance": 69108.22192339861,
                "currency": "MAD",
                "type": "SAVING_ACCOUNT",
                "_links": {
                    "self": {
                        "href": "http://localhost:8081/bankAccounts/bd1f8756-5505-4f0b-8b4a-bbcdedcd12dc"
                    },
                    "bankAccount": {
                        "href": "http://localhost:8081/bankAccounts/bd1f8756-5505-4f0b-8b4a-bbcdedcd12dc"
                    }
                }
            },
            {
                "createdAt": "2022-10-27T19:22:15.037+00:00",
                "balance": 42626.46016165123,
                "currency": "MAD",
                "type": "SAVING_ACCOUNT",
                "_links": {
                    "self": {
                        "href": "http://localhost:8081/bankAccounts/477545cd-e0e9-46ab-8c2b-b2af00b0dfb4"
                    },
                    "bankAccount": {
                        "href": "http://localhost:8081/bankAccounts/477545cd-e0e9-46ab-8c2b-b2af00b0dfb4"
                    }
                }
            },
```


*Figure 5 : RESTFUL API / Spring Data Rest*

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


```
{
  "_embedded": {
    "bankAccounts": [
      {
        "createdAt": "2022-10-27T19:33:24.806+00:00",
        "balance": 66421.557043751,
        "currency": "MAD",
        "type": "CURRENT_ACCOUNT",
        "_links": {
          "self": {
            "href": "http://localhost:8081/bankAccounts/0102e9cd-64fd-4ac0-98f8-5e64c6c7ca8a"
          },
          "bankAccount": {
            "href": "http://localhost:8081/bankAccounts/0102e9cd-64fd-4ac0-98f8-5e64c6c7ca8a"
          }
        }
      },
      {
        "createdAt": "2022-10-27T19:33:24.952+00:00",
        "balance": 26635.407586246878,
        "currency": "MAD",
        "type": "CURRENT_ACCOUNT",
        "_links": {
          "self": {
            "href": "http://localhost:8081/bankAccounts/210b6566-ac88-4cc9-b7ac-064e5bf89e6d"
          },
          "bankAccount": {
            "href": "http://localhost:8081/bankAccounts/210b6566-ac88-4cc9-b7ac-064e5bf89e6d"
          }
        }
      },
```

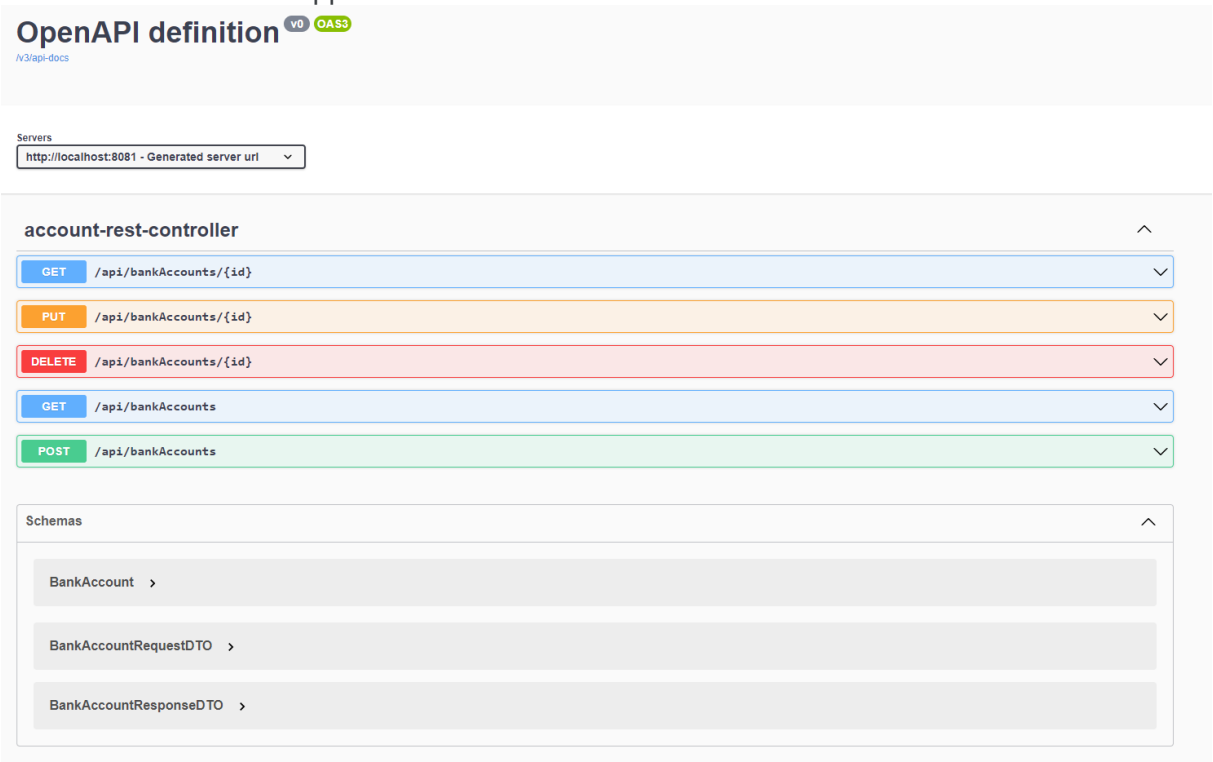

*Figure 6 : findByType method*

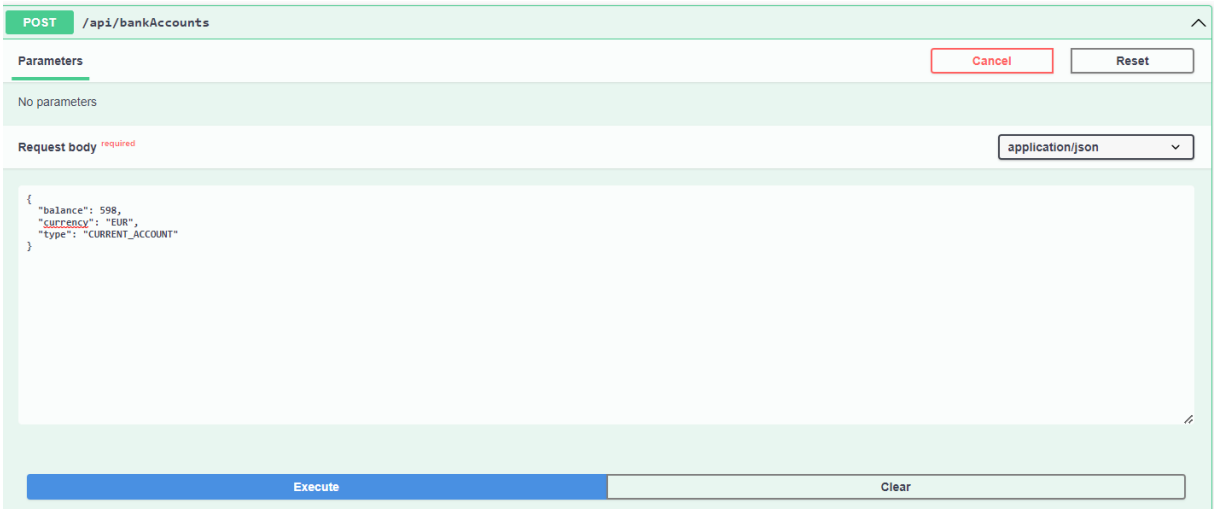
localhost:8081/bankAccounts?projection=p1

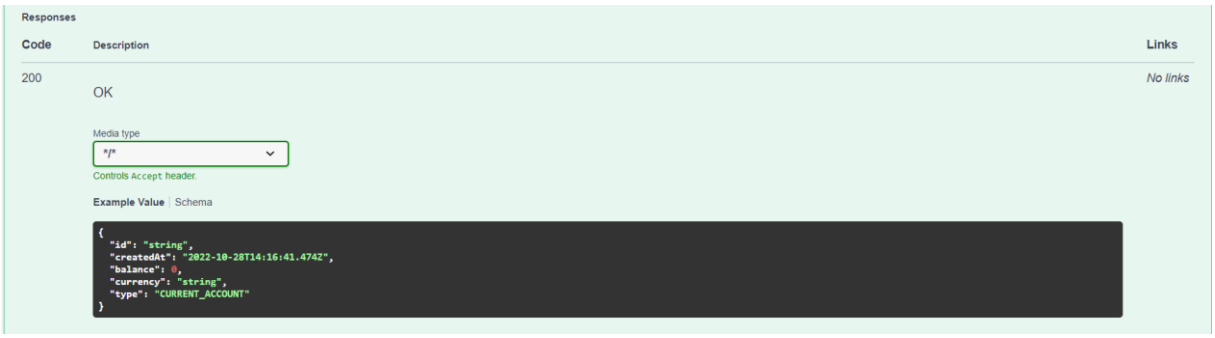

```
{
  "_embedded": {
    "bankAccounts": [
      {
        "id": "719ee37e-41aa-4162-b4ef-de4149aec142",
        "type": "SAVING_ACCOUNT",
        "_links": {
          "self": {
            "href": "http://localhost:8081/bankAccounts/719ee37e-41aa-4162-b4ef-de4149aec142"
          },
          "bankAccount": {
            "href": "http://localhost:8081/bankAccounts/719ee37e-41aa-4162-b4ef-de4149aec142{?projection}",
            "templated": true
          }
        }
      },
      {
        "id": "ffbb3652-f8b6-496a-8748-8fd157b3d256",
        "type": "CURRENT_ACCOUNT",
        "_links": {
          "self": {
            "href": "http://localhost:8081/bankAccounts/ffbb3652-f8b6-496a-8748-8fd157b3d256"
          },
          "bankAccount": {
            "href": "http://localhost:8081/bankAccounts/ffbb3652-f8b6-496a-8748-8fd157b3d256{?projection}",
            "templated": true
          }
        }
      },
```


*Figure 7 : Projection p1 avec Spring Data Rest*

9. Créer les DTOs et Mappers




10. Créer la couche Service (métier) et du micro service

```java
@Override
public BankAccountResponseDTO addAccount(BankAccountRequestDTO bankAccountDTO) {
    BankAccount bankAccount = BankAccount.builder()
            .id(UUID.randomUUID().toString())
            .createdAt(new Date())
            .balance(bankAccountDTO.getBalance())
            .type(bankAccountDTO.getType())
            .currency(bankAccountDTO.getCurrency())
            .build();
    BankAccount savedBankAccount = bankAccountRepository.save(bankAccount);
    BankAccountResponseDTO bankAccountResponseDTO = accountMapper.fromBankAccount(savedBankAccount);
    return bankAccountResponseDTO;
}

@Override
public BankAccountResponseDTO updateAccount(String id,BankAccountRequestDTO bankAccountDTO) {
    BankAccount bankAccount = BankAccount.builder()
            .id(id)
            .createdAt(new Date())
            .balance(bankAccountDTO.getBalance())
            .type(bankAccountDTO.getType())
            .currency(bankAccountDTO.getCurrency())
            .build();
    BankAccount savedBankAccount = bankAccountRepository.save(bankAccount);
    BankAccountResponseDTO bankAccountResponseDTO = accountMapper.fromBankAccount(savedBankAccount);
    return bankAccountResponseDTO;
}
```

# Partie GraphQL

1. Projection Liste de comptes avec les attributs id et balance

```
1 {
2   accountsList {
3     id,balance
4   }
5 }
```

2. Exécution

```
{
  "data": {
    "accountsList": [
      {
        "id": "d1d0fcdf-2b87-4573-9064-595bb80011cb",
        "balance": 40009.38786451647
      },
      {
        "id": "41168985-5162-4680-a054-0f86567ca25b",
        "balance": 77669.40995292782
      },
      {
        "id": "ba902a96-2e29-4d46-8e4d-9326eafebbd0",
        "balance": 41384.19231903788
      },
      {
        "id": "7cf3a6a9-bdf2-4771-8225-acf6d3da64af",
        "balance": 22874.84183473186
      },
      {
        "id": "3bba8a55-eada-4e5c-bc2d-9449b70a298d",
        "balance": 23594.660259409553
      },
      {
        "id": "323497f6-e018-47ec-b9a9-740775c080d7",
        "balance": 41982.12310279708
      },
      {
        "id": "90ec5dd9-7cc8-4c4b-a4da-43d25a3ec465",
        "balance": 53036.74599391721
      },
      {
        "id": "b4f446f8-5c7b-43e8-a897-7d41574da88b",
        "balance": 8516.319648610868
      },
      {
        "id": "042b2746-9f30-452f-aef0-4f6f0b0958d7",
        "balance": 86027.26941569851
      }
    ]
  }
}
```


3. bankAccountById query

```
query{
  bankAccountById (id : "60ed7a62-370f-4279
    id,type,balance
  }
}
```


```
{
  "data": {
    "bankAccountById": {
      "id": "60ed7a62-370f-4279-9735-d2cf5a75b91e",
      "type": "CURRENT_ACCOUNT",
      "balance": 27392.721422574727
    }
  }
}
```


4. CustomDataFetcherExceptionHandler

```java
@Component
public class CustomDataFetcherExceptionResolver extends DataFetcherExceptionResolverAdapter {
    @Override
    protected GraphQLError resolveToSingleError(Throwable ex, DataFetchingEnvironment env) {
        return new GraphQLError() {
            @Override
            public String getMessage() {
                return ex.getMessage();
            }

            @Override
            public List<SourceLocation> getLocations() { return null; }

            @Override
            public ErrorClassification getErrorType() { return null; }
        };
    }
}
```

5. Mutation Query : ajouter (1)

```
mutation{
  addAccount(bankAccount : {
    type : "CURRENT_ACCOUNT",
    balance : 2690,
    currency : "USD"
  }){
    id,balance,type
  }
}
```

```
{
  "data": {
    "addAccount": {
      "id": "98fbf661-aa44-4503-a775-8eaf866209e3",
      "balance": 2690,
      "type": "CURRENT_ACCOUNT"
    }
  }
}
```

6. Mutation Query : ajouter (2)

```
mutation($t:String,$b:Float,$c:String){
  addAccount(bankAccount : {
    type:$t,
    balance:$b,
    currency:$c
  }){
    id,balance,type,currency
  }
}
```

Variables Headers

```
{"t":"SAVING_ACCOUNT" ,"b":3542,"c":"EUR" }
```

```
{
  "data": {
    "addAccount": {
      "id": "c168e028-3432-4e1d-ad84-97c84f7747d1",
      "balance": 3542,
      "type": "SAVING_ACCOUNT",
      "currency": "EUR"
    }
  }
}
```

7. Mutation Query : mettre à jour

```
mutation($id:String,$t:String,$b:Float,$c:String){
  updateAccount(
    id:$id,
    bankAccount : {
    type:$t,
    balance:$b,
    currency:$c
  }){
    id,balance,type,currency
  }
```

Variables Headers

```
{"id": "8fb3730b-cef8-4561-8098-921447601b67",
  "t":"SAVING_ACCOUNT" ,"b":2500,"c":"EUR" }
```

```
{
  "data": {
    "updateAccount": {
      "id": "8fb3730b-cef8-4561-8098-921447601b67",
      "balance": 2500,
      "type": "SAVING_ACCOUNT",
      "currency": "EUR"
    }
  }
}
```

8. Mutation Query : supprimer

```
mutation{
  deleteAccount(id:"8fb3730b-cef8-4561-8098-921447601b67")
}
```

```
{
  "data": {
    "deleteAccount": true
  }
}
```

9. Informations de chaque compte avec le nom du client

```
query{
  accountsList {
    id,balance, customer{name}
  }
}
```

```
{
  "data": {
    "accountsList": [
      {
        "id": "4879a347-70a1-44c7-a3c3-c2aefdbc4dc7",
        "balance": 82617.39666568444,
        "customer": {
          "name": "Mohamed"
        }
      },
      {
        "id": "be8df064-e02c-44b4-b61a-bad5df52d0ae",
        "balance": 50716.54826946389,
        "customer": {
          "name": "Mohamed"
        }
      },
```

10. Informations de chaque client avec ses comptes (balance,type,currency)

```
query{
  customers{
    id,name,
    bankAccounts{balance,type,currency}
  }
}
```

```
{
  "data": {
    "customers": [
      {
        "id": "1",
        "name": "Mohamed",
        "bankAccounts": [
          {
            "balance": 22293.771904901703,
            "type": "SAVING_ACCOUNT",
            "currency": "MAD"
          },
          {
            "balance": 78667.44025354781,
            "type": "SAVING_ACCOUNT",
            "currency": "MAD"
          },
          {
            "balance": 77014.94942418295,
            "type": "SAVING_ACCOUNT",
            "currency": "MAD"
          },
          {
            "balance": 67092.26274246899,
            "type": "CURRENT_ACCOUNT",
            "currency": "MAD"
          },
```