






E E-banking-App


 Project overview


 Repository


 Issues 0


 Merge requests 0


 Requirements


 CI/CD


 Security & Compliance


 Operations

 Packages & Registries

 Analytics

 **Wiki**

 Snippets

 Members


Abdellah AHBANE > E-banking-App > Wiki > Customer Service

Last edited by  **Abdou Ahban** 11 months ago

Page history

New page

Customer Service



All what you need to run this service is to**

- `CREATE database customers ;`
- `Run the Registry service eureka & the account-service && officer-service.`

If you take a look at the main application :

```
@SpringBootApplication
@EnableFeignClients
public class CustomerServiceApplication implements CommandLineRunner {

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

    @Autowired
    AccountService accountService;

    @Autowired
    CustomerRepository clientRepository;
    @Override
    public void run(String... args) throws Exception {

        //TODO remove for mode prod !
        currentAccount c1CurrentAccount = accountService.findCurrentAccountByCustomerId ( 1L );
        savingsAccount c1savingsAccount = accountService.findSavingsAccountByCustomerId ( 1L );
        clientRepository.save ( new Customer ( null, "Abdou", "ahban", "aa@uca.ma", "Aoulouz, Targu" );

        //TODO remove for mode prod !
        currentAccount c2CurrentAccount = accountService.findCurrentAccountByCustomerId ( 2L );
        savingsAccount c2savingsAccount = accountService.findSavingsAccountByCustomerId ( 2L );
        clientRepository.save ( new Customer ( null, "Manal", "Outaleb", "Manal@uca.ma", " Somewh" );

        clientRepository.findAll ().forEach ( System.out::println );

    }
}
```

This will simply create the table in DB and insert those data into it. But if u stop running it and re-run it again, it will show exceptions. This is because there're constraints (unique) in some columns so if you insert the same data it will not accept it simply.

So you need, either

- to drop the database by `DROP database customers` Or
- to edit the main application by this :

```
@SpringBootApplication
@EnableFeignClients
public class CustomerServiceApplication {

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

}
```

I recommend the first one, by dropping just the database especially if you have IntelliJ, it's easy and quick.



- PS. You can apply this to all my codes.

APIs of this Service :

NB: In this service: `host = http://localhost:8081/api`

GET :

- `host/customers` : To get the list of all customers. (**ADMINS are the only ones allowed to call this**)
- `host/customers/{id}` : To get a customer with the given id. (**ADMINS & OFFICIERS (if they have same agencyId) & CLIENT (if he's trying to get his own infos) are the only ones allowed to call this**)
- `host/customerByEmail/{email}` : To get a customer with his email. (**ADMINS & OFFICIERS (if they have same agencyId) & CLIENT (if he's trying to get his own infos) are the only ones allowed to call this**)
- `host/customersByCin/{CIN}` : To get a Customer with his cin or list of customers which have cin contains CIN (all the cin or just part of it); Would be interesting when we search a customer in a list using his chin. (**ADMINS (all customers) & OFFICIERS (all**

 Clone repository  Edit sidebar

AA Documentation for using Gateway !

Account Service

Agency Service

Authentication Service

Customer Service

GlobalParameters Service

Officer Service

Request Service

Transaction Service

customers of his agency) & are the only ones allowed to call this)

- `host/customersByStatus/{Status}` : To get all the customers with the given Status. (`ADMINs` (all customers) & `OFFICIERS` (all customers of his agency) & are the only ones allowed to call this)
- `host/customersByContract/{Contract}` : To get all the customers with the given Contract. (`ADMINs` (all customers) & `OFFICIERS` (all customers of his agency) & are the only ones allowed to call this)
- `host/customersByAgency/{agencyId}` : To get all the customers with the given agencyId. (`ADMINs` (all agencesIds) & `OFFICIERS` (just the id of his agency) & are the only ones allowed to call this)
- `host/customersSizePerAgency` : To get numbers of customers for the agency of the officer connected. (`OFFICIERS` (all agencesIds) are the only ones allowed to call this)
- `host/getMyId` : To get the customer's Id for the one who is connected.

This one is for the backend.

- `host/CustomerIdByEmail/{email}` : To get the customersId using his email (authentication username). (`CLIENTs` are the only ones allowed to call this; there're the ones who will need this)

POST :

- `host/customers` : to create a customer, the given data should be in this format: (`ADMINs` & `OFFICIERS` (by default it will take his agencyId) & are the only ones allowed to call this)

for example :

```
{
  "firstName": "Mounib",
  "lastName": "El Boujbaoui",
  "email": "MNA@uca.ma",
  "address": "Sidi Abad, Marrakech",
  "birthDay": "1997-10-20",
  "cin": "JP8787256",
  "phoneNumber": "0973456776894567",
  "operator": "Orange",
  "gender": "Masculin",
  "status": "Actif",
  "contract": "Personnel",
  "agencyId": 1
}
```

This one is for backend.

- `host/customer/currentAccount` : This one we used it in the backend of account-service to add accounts number to the customer when we add the current-account type.

This one is for backend.

- `host/customer/savingsAccount` : same for the savingsAccount-account type.

PUT:

- `host/customers/{id}` : to update a customer with the given id, the given data should be in this format: (`ADMINs` & `OFFICIERS` (if they have same agencyId) & `CLIENT` (if he's trying to edit his own infos) are the only ones allowed to call this)

for example :

```
{
  "firstName": "Abdellah",
  "lastName": "ahban",
  "email": "abdou.ahban61@uca.ma",
  "address": "Aoulouz, Taroudant",
  "birthDay": "1997-03-16",
  "cin": "JP87256",
  "phoneNumber": "09734567894567",
  "operator": "Orange",
  "gender": "Masculin",
  "status": "Actif",
  "contract": "Personnel",
  "creationDate": "2020-05-16",
  "agencyId": 1,
  "currentAccountNum": 76528932992,
  "savingsAccountNum": 76528932992
}
```

We can use it to move the customer to another agency by editing the agencyId.. Or to edit some specific attributs.

DELETE :

- `host/customers/{id}` : To delete a customer with the given id, all his accounts we ll be deleted too ! To get all the customers with the given agencyId. (`ADMINs` (all customers) & `OFFICIERS` (just if the customer has the same agencyId as the officer) & are the only ones allowed to call this)

