

# Tarik FERTAHI

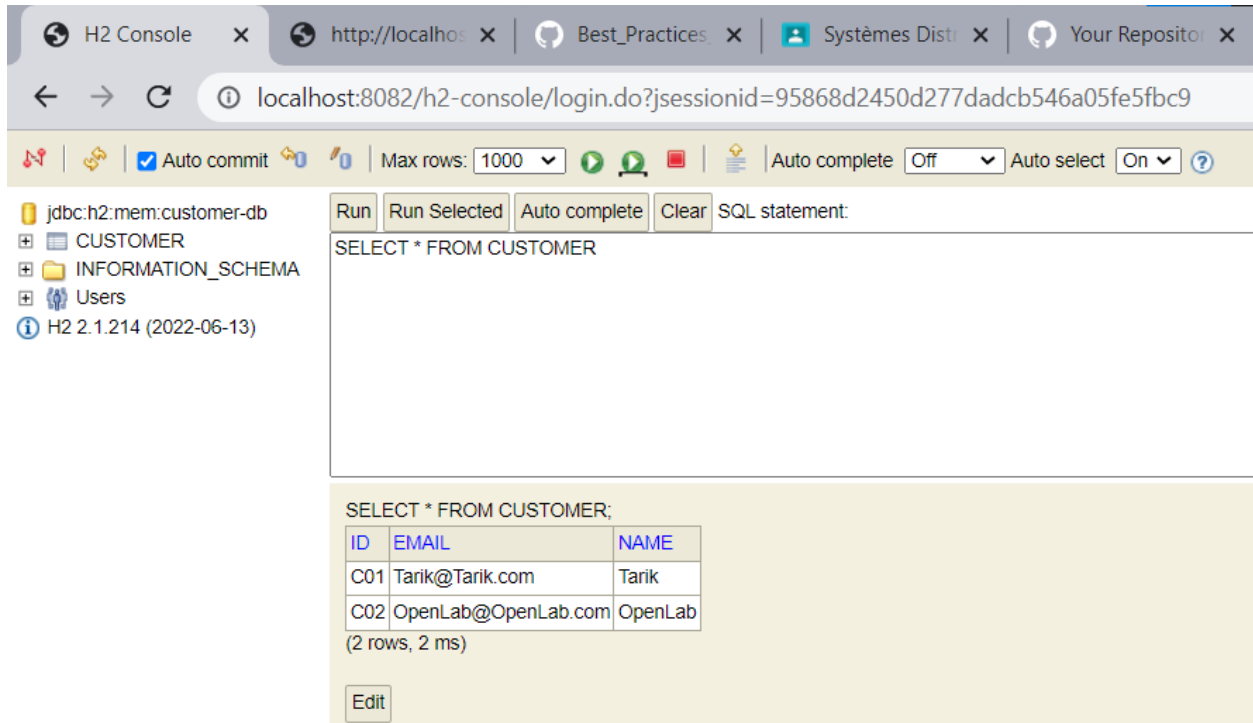
## II-BDCC

Année universitaire : 2022/2023

## La création de deux Clients

```
@Bean
CommandLineRunner start(CustomerService customerService){
    return args -> {
        customerService.save(new CustomerRequestDTO( id: "C01", name: "Tarik", email: "Tarik@Tarik.com"));
        customerService.save(new CustomerRequestDTO( id: "C02", name: "OpenLab", email: "OpenLab@OpenLab.com"));
    };
}
```

## Vérification des clients au niveau de la base de données



The screenshot shows the H2 Console web interface in a browser. The address bar shows the URL: `localhost:8082/h2-console/login.do?jsessionid=95868d2450d277dadcb546a05fe5fbc9`. The interface includes a sidebar with the database structure: `jdbc:h2:mem:customer-db`, `CUSTOMER`, `INFORMATION_SCHEMA`, and `Users`. The main area displays the SQL statement `SELECT * FROM CUSTOMER` and its results in a table format.

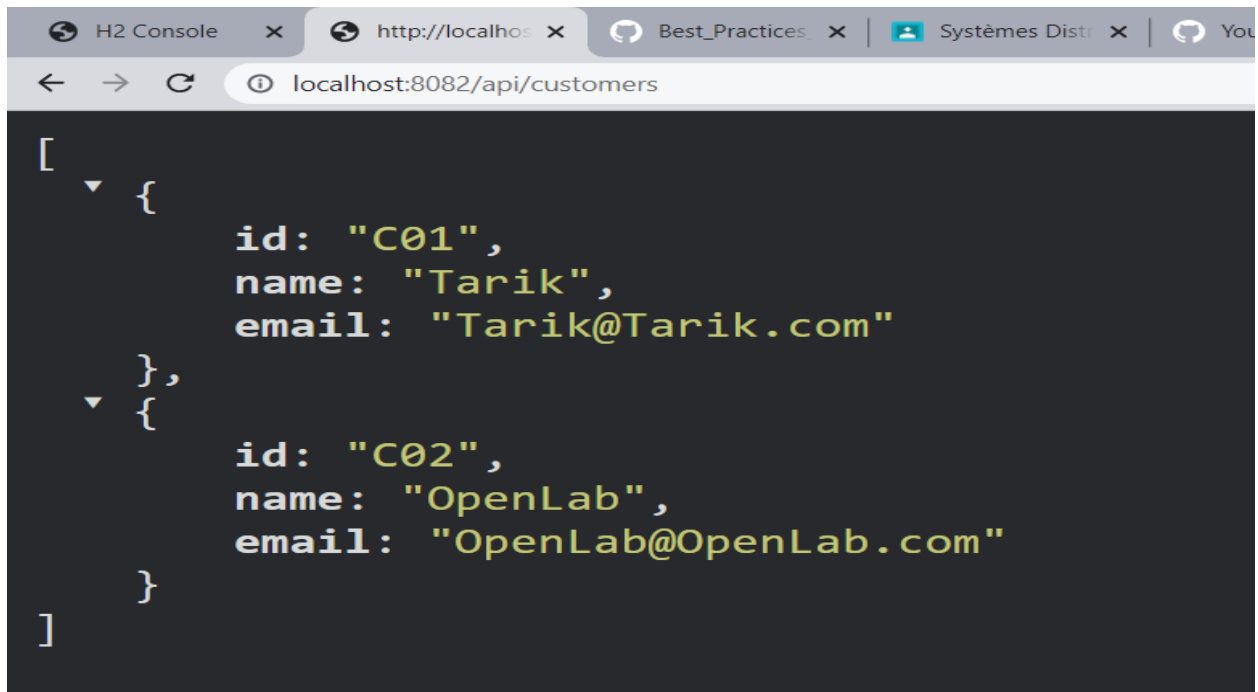
SQL statement: `SELECT * FROM CUSTOMER`

ID	EMAIL	NAME
C01	Tarik@Tarik.com	Tarik
C02	OpenLab@OpenLab.com	OpenLab

(2 rows, 2 ms)

Edit

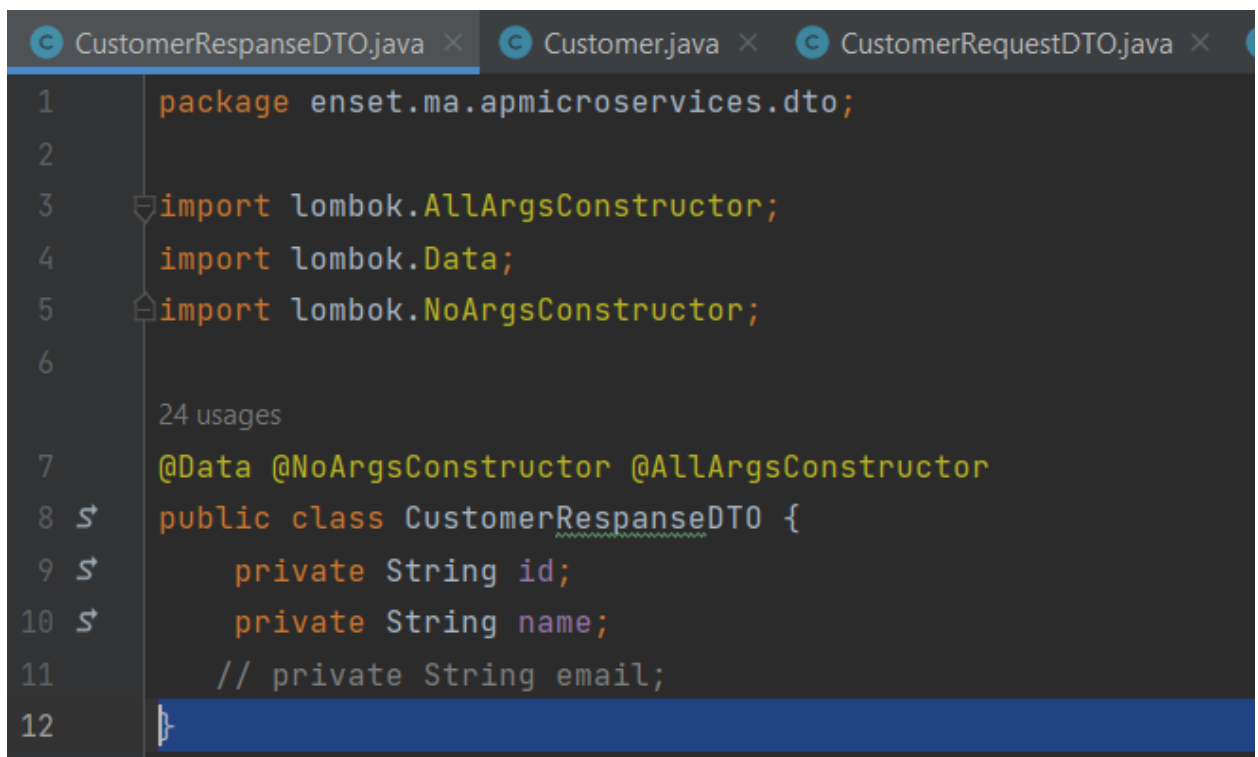
## Affichage des clients



A screenshot of a web browser window. The address bar shows 'localhost:8082/api/customers'. The page content displays a JSON array of two customer objects. The first object has id 'C01', name 'Tarik', and email 'Tarik@Tarik.com'. The second object has id 'C02', name 'OpenLab', and email 'OpenLab@OpenLab.com'.

```
[
  {
    id: "C01",
    name: "Tarik",
    email: "Tarik@Tarik.com"
  },
  {
    id: "C02",
    name: "OpenLab",
    email: "OpenLab@OpenLab.com"
  }
]
```

Grace au DTOs on peut ignorer l'attribut email dans la partie UI



A screenshot of an IDE showing the 'CustomerResponseDTO.java' file. The code defines a package, imports Lombok annotations, and defines a class with private fields for id and name, and a commented-out email field.

```
1 package enset.ma.apmicroservices.dto;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Data;
5 import lombok.NoArgsConstructor;
6
7 24 usages
8 @Data @NoArgsConstructor @AllArgsConstructor
9 public class CustomerResponseDTO {
10     private String id;
11     private String name;
12     // private String email;
13 }
```



A screenshot of a web browser window. The top bar shows three tabs: 'H2 Console', 'http://localhost', and 'Best\_Practices'. The address bar displays 'localhost:8082/api/customers'. The main content area shows a JSON array of two customer objects. The first object has 'id: "C01"' and 'name: "Tarik"'. The second object has 'id: "C02"' and 'name: "OpenLab"'. The JSON is formatted with syntax highlighting and collapsible nodes indicated by small triangles.

```
[
  {
    id: "C01",
    name: "Tarik"
  },
  {
    id: "C02",
    name: "OpenLab"
  }
]
```

Consulter un client par ID



A screenshot of a web browser window. The top bar shows three tabs: 'H2 Console', 'http://localhost', and 'Best\_Practices'. The address bar displays 'localhost:8082/api/customers/C01'. The main content area shows a single JSON object representing the customer with 'id: "C01"' and 'name: "Tarik"'. The JSON is formatted with syntax highlighting.

```
{
  id: "C01",
  name: "Tarik"
}
```

## Test de la méthode Save

Overview **POST http://localhost:8082/** GET http://localhost:8082/a PUT http://localhost:8081/b + ...


http://localhost:8082/api/customers


**POST** http://localhost:8082/api/customers

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

none form-data x-www-form-urlencoded **raw** binary GraphQL **JSON** v

```
1 {
2   "name": "Tarik 2",
3   "email": "Tarik2@Tarik2.com"
4 }
```

Body Cookies Headers (5) Test Results  Status: 200 OK

Pretty Raw Preview Visualize **JSON** v 

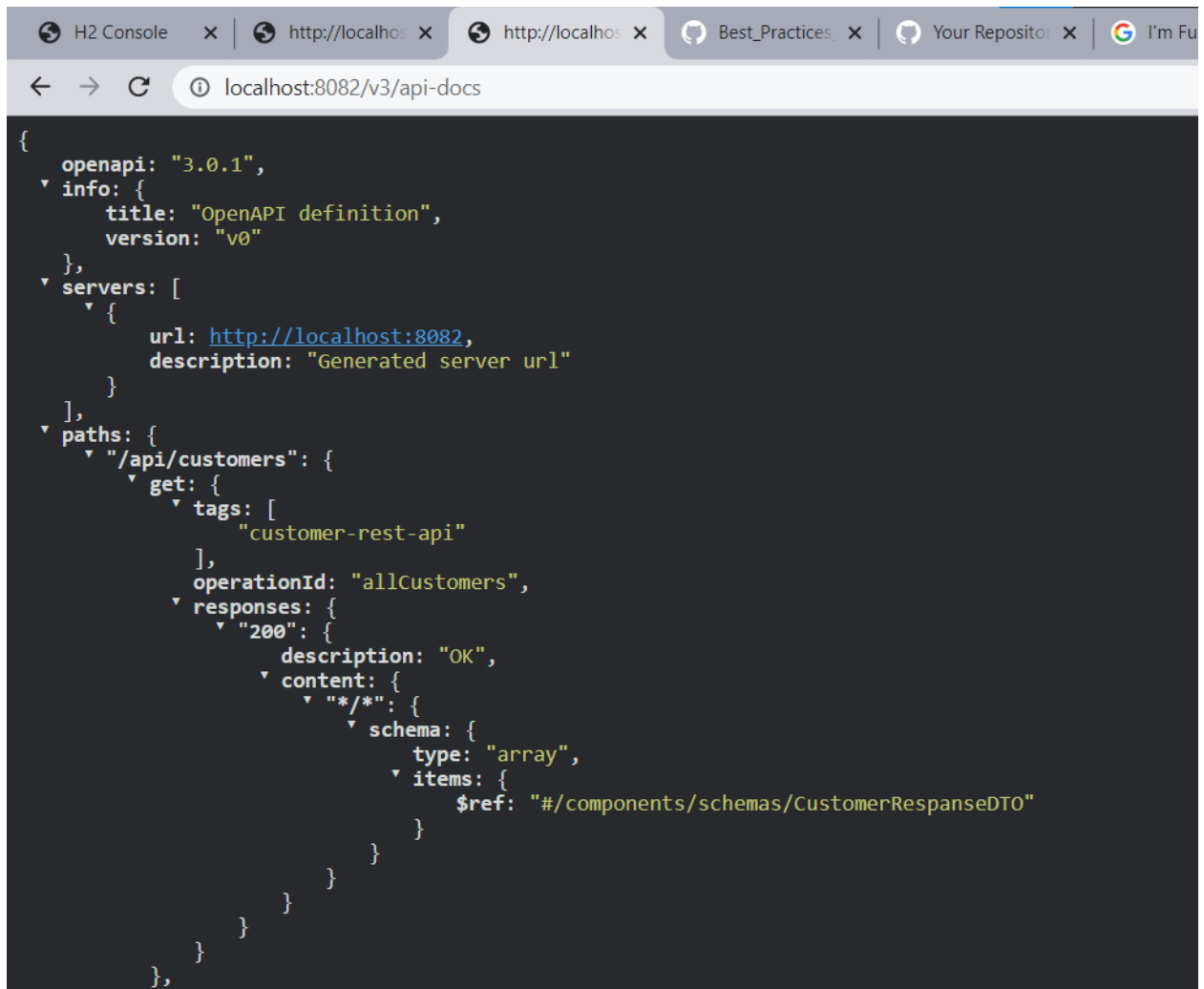
```
1 {
2   "id": "8516fc68-4289-4a62-8b15-ae464ab43200",
3   "name": "Tarik 2",
4   "email": "Tarik2@Tarik2.com"
5 }
```

H2 Console x http://localhost:8082/api/customers x Best\_Practices\_Mi x Your Repositories x

localhost:8082/api/customers

```
[
  {
    id: "C01",
    name: "Tarik",
    email: "Tarik@Tarik.com"
  },
  {
    id: "C02",
    name: "OpenLab",
    email: "OpenLab@OpenLab.com"
  },
  {
    id: "8516fc68-4289-4a62-8b15-ae464ab43200",
    name: "Tarik 2",
    email: "Tarik2@Tarik2.com"
  }
]
```

## Documentation Api



The screenshot shows a web browser window with the address bar displaying `localhost:8082/v3/api-docs`. The browser tabs include "H2 Console", "http://localhost:", "http://localhost:", "Best\_Practices", "Your Repositor", and "I'm Fu". The main content area displays a JSON document representing an OpenAPI definition. The JSON is partially expanded, showing the `info` and `paths` sections. The `info` section includes `openapi: "3.0.1"`, `title: "OpenAPI definition"`, and `version: "v0"`. The `paths` section includes a `get` method for `/api/customers` with a `tags` array containing `"customer-rest-api"`, an `operationId` of `"allCustomers"`, and a `responses` object. The `responses` object has a `200` response with a `description` of `"OK"` and a `content` object. The `content` object has a `*/*` media type with a `schema` object. The `schema` object has a `type` of `"array"` and an `items` object with a `$ref` of `"#/components/schemas/CustomerRespansedDTO"`.

```
{
  openapi: "3.0.1",
  info: {
    title: "OpenAPI definition",
    version: "v0"
  },
  servers: [
    {
      url: http://localhost:8082,
      description: "Generated server url"
    }
  ],
  paths: {
    "/api/customers": {
      get: {
        tags: [
          "customer-rest-api"
        ],
        operationId: "allCustomers",
        responses: {
          "200": {
            description: "OK",
            content: {
              "*/*": {
                schema: {
                  type: "array",
                  items: {
                    $ref: "#/components/schemas/CustomerRespansedDTO"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

OpenAPI definition

OpenAPI definition

draft

OpenAPI definition

api/customers

GET all Customers

POST save

GET get Customer

OpenAPI definition / api/customers / get Customer

GET

{{baseUrl}}/api/customers/C01

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Query Params

KEY	VALUE
Key	Value

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": "C01",
3   "name": "Tarik",
4   "email": "Tarik@Tarik.com"
5 }
```

H2 Console x

http://locall x


http://locall x

Swagger UI x

Best\_Practic x

← → ↻ ⓘ

localhost:8082/swagger-ui/index.html

 **Swagger**  
Supported by SMARTBEAR

/v3/api-docs

# OpenAPI definition v0 OAS3

/v3/api-docs

## Servers

http://localhost:8082 - Generated server url

## customer-rest-api

**GET** /api/customers

**POST** /api/customers

**GET** /api/customers/{id}