Lab No: 9 by Arkadeep Ganguly

Lab Name: Bulk Customer Generation

Overview, Objectives & Concepts (Combined)

Bulk customer generation involves creating or importing multiple customer records simultaneously, commonly used for database initialization, system testing, or onboarding large user groups.For this lab,as per the manual steps,i have implemented the classes User,UserRepository,UserService and UserController along with the exception handling class GlobalExceptionHandler to manage controller errors.Then later,I ran the application and tested it using PostMan.

Exercise / Output

(Answer the exercise below)

1. Generate Bulk customers using `/customers` API

Changes in UserService:

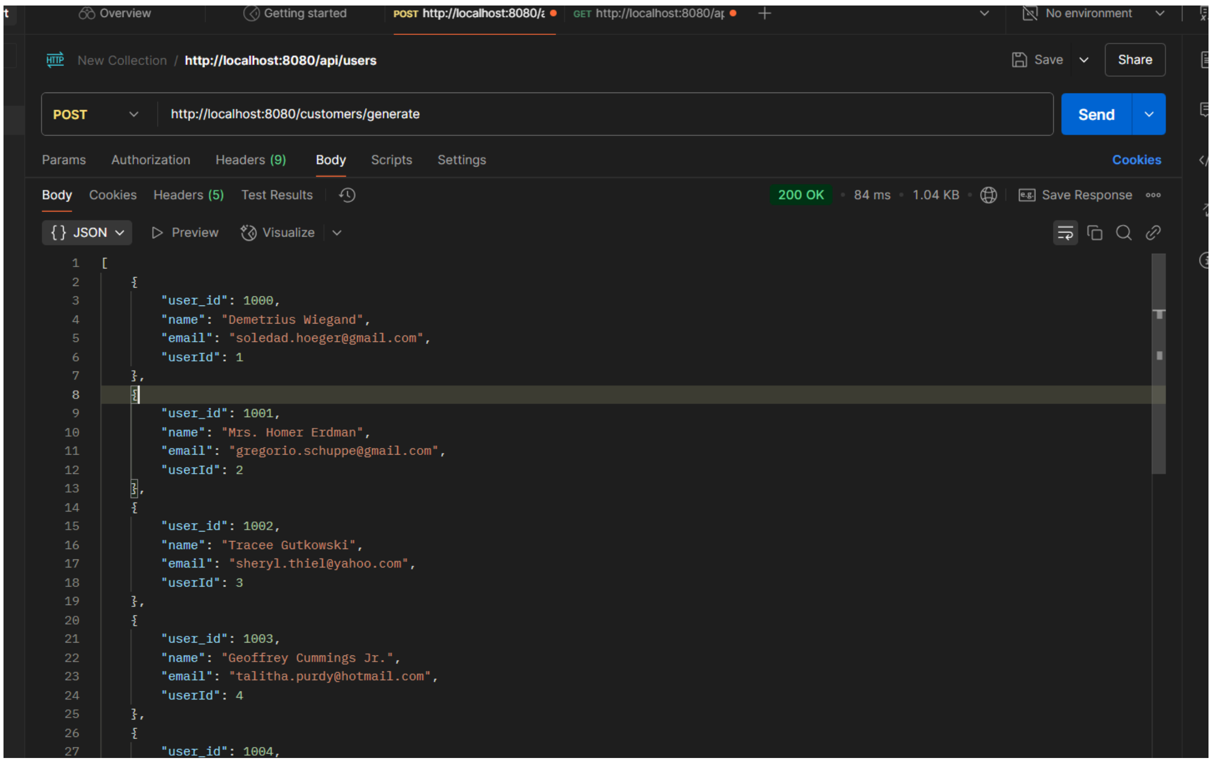
```
public List<User> generateBulkCustomers(int count) {
   List<User> users = new ArrayList<>();
   for (int i = 0; i < count; i++) {
       User user = new User();
       user.setUserId((long) (i + 1)); // Auto-increment ID (if not using DB auto-generation)
       user.setUser_id((long) (i + 1000)); // Custom user ID
       user.setName(faker.name().fullName()); // Random name
       user.setEmail(faker.internet().emailAddress()); // Random email
       users.add(user);
   }
   return userRepository.saveAll(users); // Batch insert
}
```

Changes in UserController:

```

 @PostMapping("/generate")
public ResponseEntity<List<User>> generateBulkCustomers(
       @RequestParam(defaultValue = "10") int count) {
   List<User> users = userService.generateBulkCustomers(count);
   return ResponseEntity.ok(users);
}
```

Output:

1)Postman


New Collection / http://localhost:8080/api/users

POST http://localhost:8080/customers/generate

Params Authorization Headers (9) Body Scripts Settings

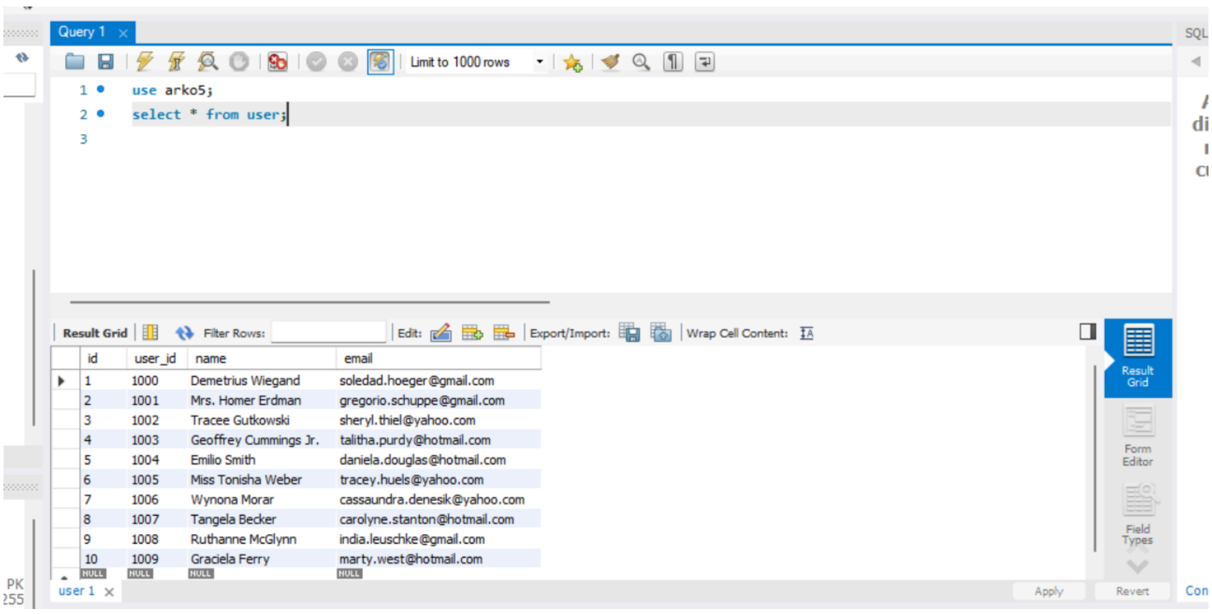
Body Cookies Headers (5) Test Results — 200 OK · 84 ms · 1.04 KB

```
[
    {
        "user_id": 1000,
        "name": "Demetrius Wiegand",
        "email": "soledad.hoeger@gmail.com",
        "userId": 1
    },
    {
        "user_id": 1001,
        "name": "Mrs. Homer Erdman",
        "email": "gregorio.schuppe@gmail.com",
        "userId": 2
    },
    {
        "user_id": 1002,
        "name": "Tracee Gutkowski",
        "email": "sheryl.thiel@yahoo.com",
        "userId": 3
    },
    {
        "user_id": 1003,
        "name": "Geoffrey Cummings Jr.",
        "email": "talitha.purdy@hotmail.com",
        "userId": 4
    },
    {
        "user_id": 1004,
```

```
[

    {

        "user_id": 1000,

        "name": "Demetrius Wiegand",

        "email": "soledad.hoeger@gmail.com",

        "userId": 1

    },

    {

        "user_id": 1001,

        "name": "Mrs. Homer Erdman",

        "email": "gregorio.schuppe@gmail.com",

        "userId": 2

    },
```

```json
    {
        "user_id": 1002,
        "name": "Tracee Gutkowski",
        "email": "sheryl.thiel@yahoo.com",
        "userId": 3
    },
    {
        "user_id": 1003,
        "name": "Geoffrey Cummings Jr.",
        "email": "talitha.purdy@hotmail.com",
        "userId": 4
    },
    {
        "user_id": 1004,
        "name": "Emilio Smith",
        "email": "daniela.douglas@hotmail.com",
        "userId": 5
    },
    {
        "user_id": 1005,
        "name": "Miss Tonisha Weber",
        "email": "tracey.huels@yahoo.com",
        "userId": 6
    },
    {
        "user_id": 1006,
        "name": "Wynona Morar",

        "email": "cassaundra.denesik@yahoo.com",

        "userId": 7

    },

    {

        "user_id": 1007,

        "name": "Tangela Becker",

        "email": "carolyne.stanton@hotmail.com",

        "userId": 8

    },

    {

        "user_id": 1008,

        "name": "Ruthanne McGlynn",

        "email": "india.leuschke@gmail.com",

        "userId": 9

    },

    {

        "user_id": 1009,

        "name": "Graciela Ferry",

        "email": "marty.west@hotmail.com",

        "userId": 10

    }

]
```

2)Changes made in Database:



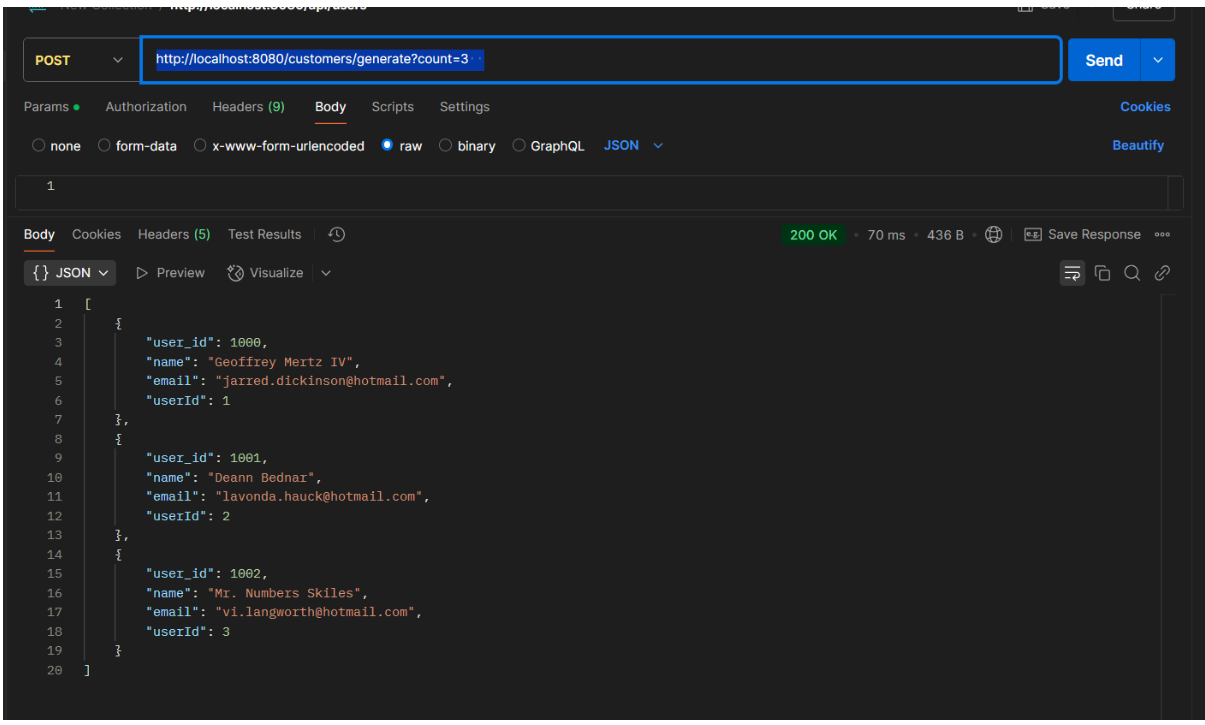
```
use arko5;
select * from user;
```

id	user_id	name	email
1	1000	Demetrius Wiegand	soledad.hoeger@gmail.com
2	1001	Mrs. Homer Erdman	gregorio.schuppe@gmail.com
3	1002	Tracee Gutkowski	sheryl.thiel@yahoo.com
4	1003	Geoffrey Cummings Jr.	talitha.purdy@hotmail.com
5	1004	Emilio Smith	daniela.douglas@hotmail.com
6	1005	Miss Tonisha Weber	tracey.huels@yahoo.com
7	1006	Wynona Morar	cassaundra.denesik@yahoo.com
8	1007	Tangela Becker	carolyne.stanton@hotmail.com
9	1008	Ruthanne McGlynn	india.leuschke@gmail.com
10	1009	Graciela Ferry	marty.west@hotmail.com
NULL	NULL	NULL	NULL

2. Test the API using POST method in Postman

Here,I have decided to go with some test cases.They are:

- http://localhost:8080/customers/generate?count=3

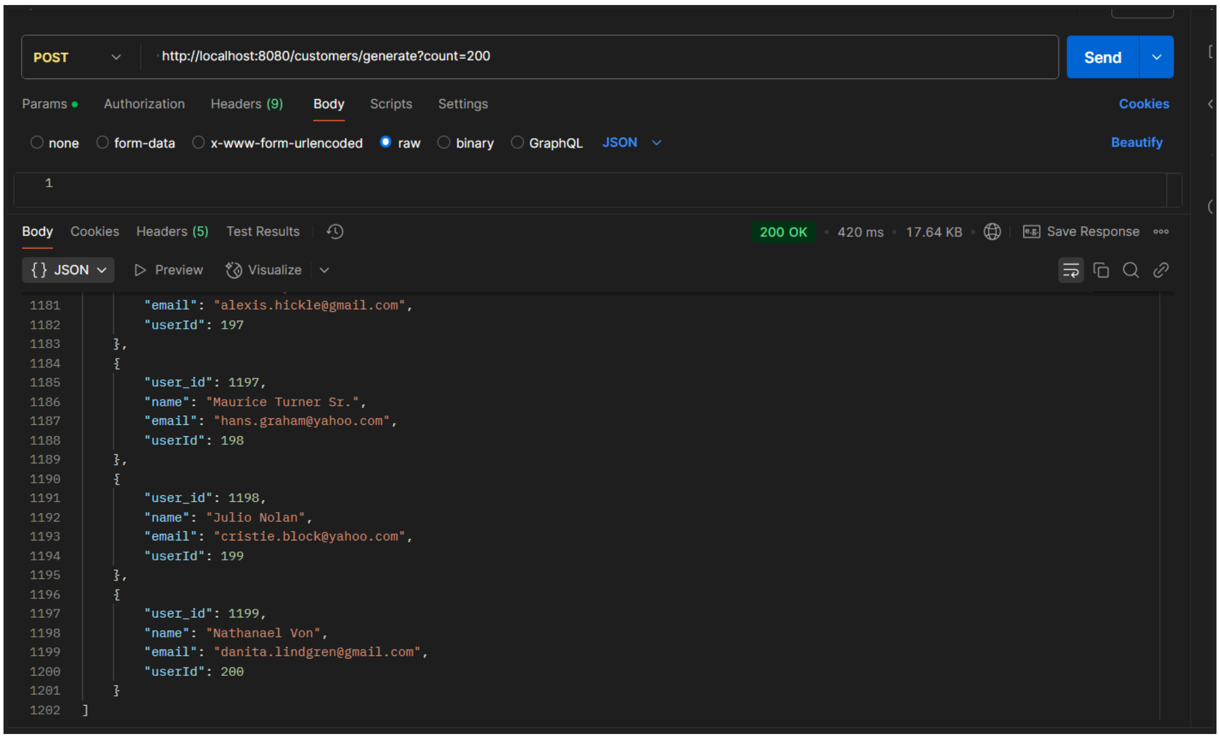


POST http://localhost:8080/customers/generate?count=3

200 OK · 70 ms · 436 B

```
[
    {
        "user_id": 1000,
        "name": "Geoffrey Mertz IV",
        "email": "jarred.dickinson@hotmail.com",
        "userId": 1
    },
    {
        "user_id": 1001,
        "name": "Deann Bednar",
        "email": "lavonda.hauck@hotmail.com",
        "userId": 2
    },
    {
        "user_id": 1002,
        "name": "Mr. Numbers Skiles",
        "email": "vi.langworth@hotmail.com",
        "userId": 3
    }
]
```

- POST http://localhost:8080/customers/generate?count=200

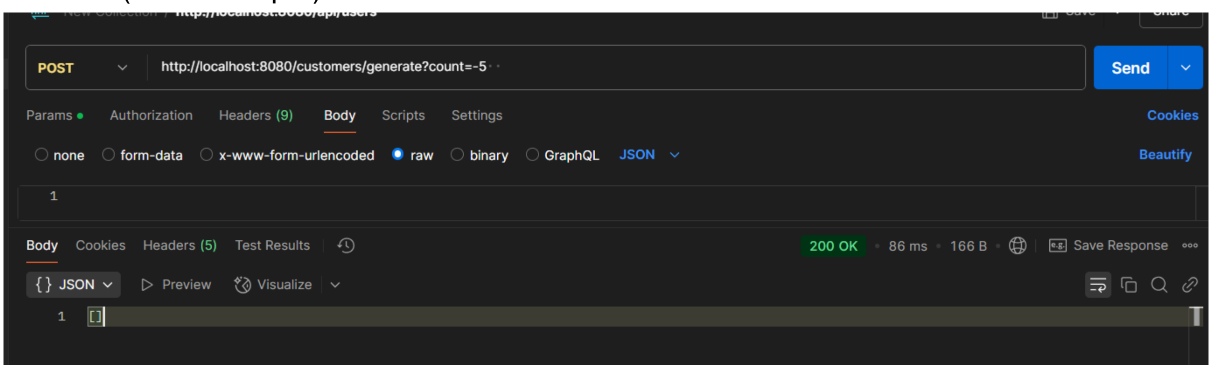


http://localhost:8080/customers/generate?count=200

200 OK • 420 ms • 17.64 KB

```
        "email": "alexis.hickle@gmail.com",
        "userId": 197
    },
    {
        "user_id": 1197,
        "name": "Maurice Turner Sr.",
        "email": "hans.graham@yahoo.com",
        "userId": 198
    },
    {
        "user_id": 1198,
        "name": "Julio Nolan",
        "email": "cristie.block@yahoo.com",
        "userId": 199
    },
    {
        "user_id": 1199,
        "name": "Nathanael Von",
        "email": "danita.lindgren@gmail.com",
        "userId": 200
    }
]
```

- Count=-5(Invalid Output)



http://localhost:8080/customers/generate?count=-5

200 OK • 86 ms • 166 B

```
[]
```