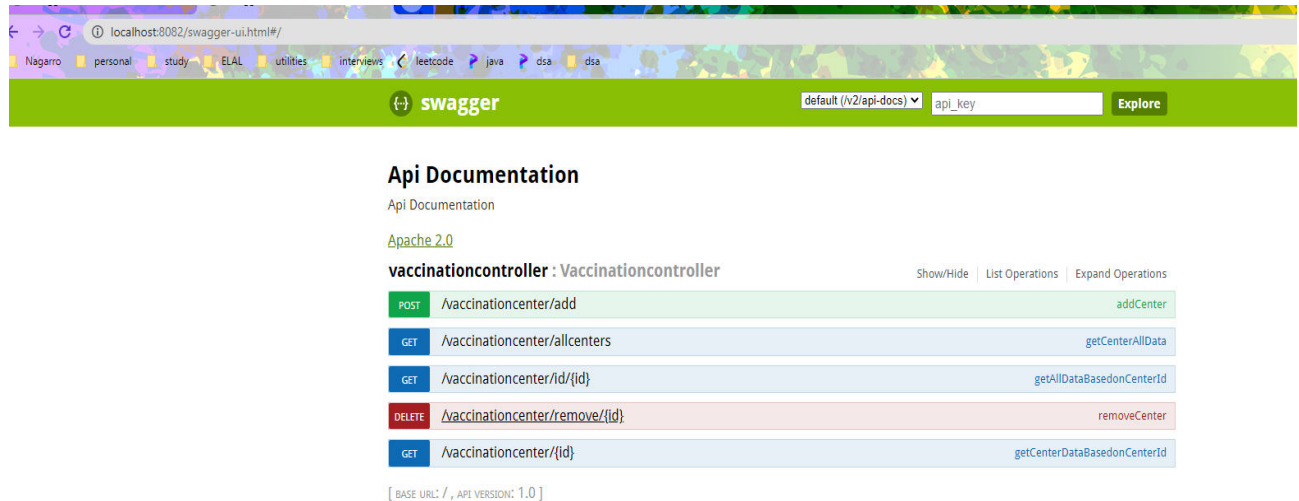


Vaccinationcenter micro service



The image shows the Swagger UI interface for the Vaccinationcenter micro service. The browser address bar shows `localhost:8082/swagger-ui.html/#/`. The Swagger logo is in the top left, and the top right has a dropdown menu set to `default (/v2/api-docs)` and an `api_key` input field with an `Explore` button.

Api Documentation

Api Documentation

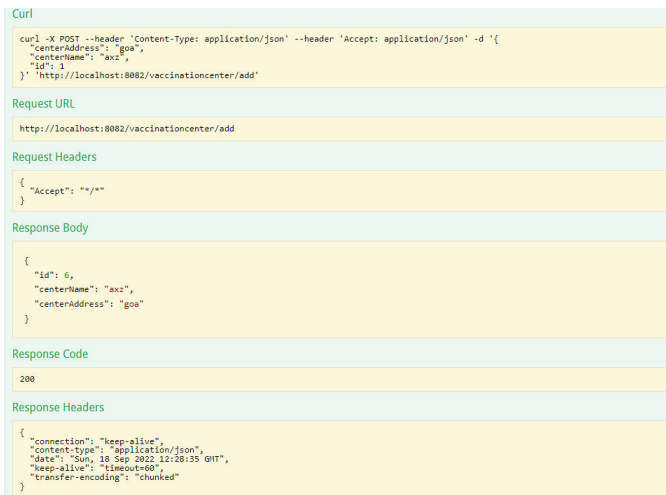
[Apache 2.0](#)

vaccinationcontroller : Vaccinationcontroller [Show/Hide](#) [List Operations](#) [Expand Operations](#)

POST	<code>/vaccinationcenter/add</code>	<code>addCenter</code>
GET	<code>/vaccinationcenter/allcenters</code>	<code>getCenterAllData</code>
GET	<code>/vaccinationcenter/id/{id}</code>	<code>getAllDataBasedonCenterId</code>
DELETE	<code>/vaccinationcenter/remove/{id}</code>	<code>removeCenter</code>
GET	<code>/vaccinationcenter/{id}</code>	<code>getCenterDataBasedonCenterId</code>

[BASE URL: / , API VERSION: 1.0]

Add operation



Curl

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{
  "centerAddress": "goa",
  "centerName": "axi",
  "id": 1
}' http://localhost:8082/vaccinationcenter/add
```

Request URL

```
http://localhost:8082/vaccinationcenter/add
```

Request Headers

```
{
  "Accept": "*/*/*"
}
```

Response Body

```
{
  "id": 6,
  "centerName": "axi",
  "centerAddress": "goa"
}
```

Response Code

```
200
```

Response Headers

```
{
  "connection": "keep-alive",
  "content-type": "application/json",
  "date": "Sun, 18 Sep 2022 12:28:25 GMT",
  "keep-alive": "timeout=60",
  "transfer-encoding": "chunked"
}
```

Get all centers operation

The screenshot displays a REST client interface with three main sections: Request URL, Request Headers, and Response Body. The Request URL is `http://localhost:8082/vaccinationcenter/allcenters`. The Request Headers section contains a single header: `"Accept": "*//*"`. The Response Body shows a JSON array of four center objects. The first two centers are in Hyderabad, and the next two are in Bangalore. The JSON is formatted with syntax highlighting.

Request URL

```
http://localhost:8082/vaccinationcenter/allcenters
```

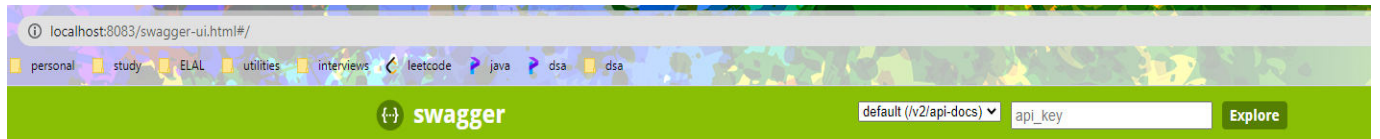
Request Headers

```
{
  "Accept": "*//*"
}
```

Response Body

```
[
  {
    "id": 2,
    "centerName": "abc",
    "centerAddress": "hyderabad"
  },
  {
    "id": 3,
    "centerName": "abcd",
    "centerAddress": "hyderabad"
  },
  {
    "id": 4,
    "centerName": "bvc",
    "centerAddress": "bangloore"
  },
  {
    "id": 6,
    "centerName": "axz",
    "centerAddress": "-----"
  }
]
```

Citizen micro service



Api Documentation

Api Documentation

[Apache 2.0](#)

citizen-controller : Citizen Controller

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

POST	/citizen/add	addCitizen
GET	/citizen/allcitizens	allCitizens
GET	/citizen/id/{id}	getById
DELETE	/citizen/remove/{id}	removeCitizen
GET	/citizen/test	test

[BASE URL: / , API VERSION: 1.0]

[Try it out!](#) [Hide Response](#)

Curl

```
curl -X GET --header 'Accept: application/json' 'http://localhost:8083/citizen/allcitizens'
```

Request URL

```
http://localhost:8083/citizen/allcitizens
```

Request Headers

```
{  
  "Accept": "*/*"  
}
```

Response Body

```
[  
  {  
    "id": 1,  
    "name": "harry",  
    "vaccinationCenterId": 1  
  },  
  {  
    "id": 2,  
    "name": "harry2",  
    "vaccinationCenterId": 2  
  }  
]
```

Response Code

```
200
```

Request URL

```
http://localhost:8083/citizen/add
```

Request Headers

```
{  
  "Accept": "*/*"  
}
```

Response Body

```
{  
  "id": 4,  
  "name": "sam",  
  "vaccinationCenterId": 5  
}
```

Response Code

```
200
```

Response Headers

```
{  
  "connection": "keep-alive",  
  "content-type": "application/json",  
  "date": "Sun, 18 Sep 2022 12:36:39 GMT",  
  "keep-alive": "timeout=60",  
  "transfer-encoding": "chunked"  
}
```

Micro service communication

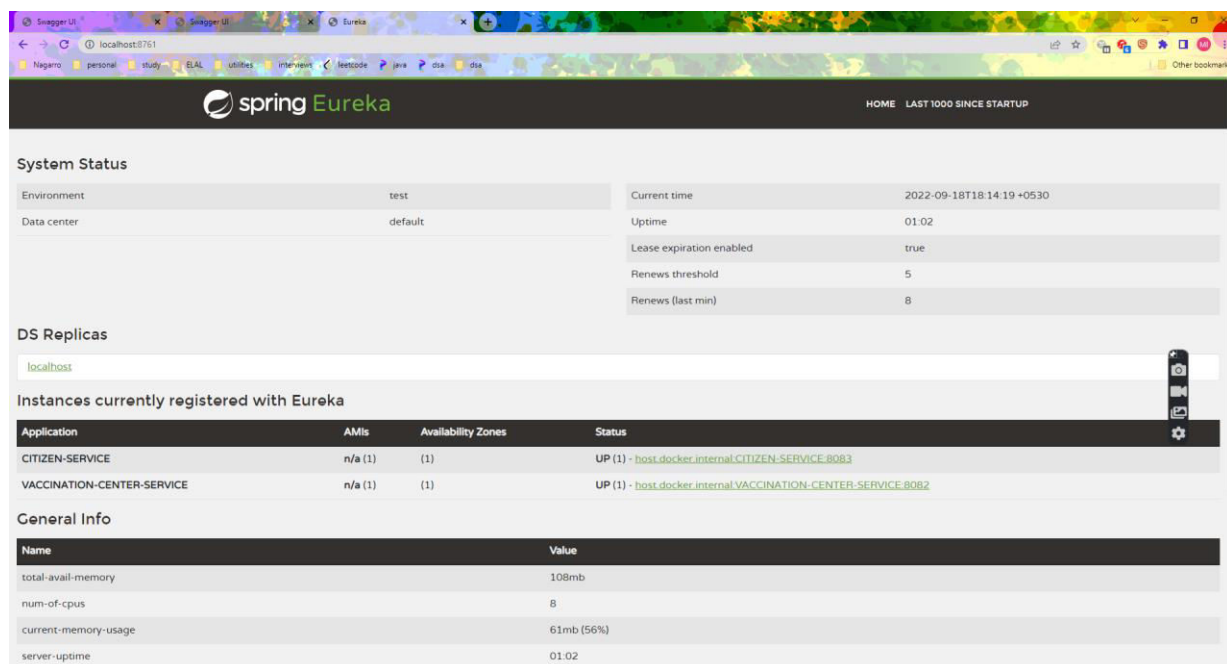
Request URL
<code>http://localhost:8082/vaccinationcenter/id/2</code>
Request Headers
<code>{ "Accept": "*/*" }</code>
Response Body
<code>{ "center": { "id": 2, "centerName": "abc", "centerAddress": "hyderabad" }, "citizens": [{ "id": 2, "name": "harry2", "vaccinationCenterId": 2 }] }</code>
Response Code
<code>200</code>

1. I used rest template to call citizen api from vaccinationcenter controller

```
@GetMapping(path = "/id/{id}")  
//@HystrixCommand(fallbackMethod="handleCitizenDowntime")  
public ResponseEntity<RequiredResponse> getAllDataBasedonCenterId(@PathVariable Integer id){  
    RequiredResponse requiredResponse = new RequiredResponse();  
    VaccinationCenter center = centerRepo.findById(id).get();  
    requiredResponse.setCenter(center);  
  
    List<Citizen> listOfCitizens = restTemplate.getForObject("http://CITIZEN-SERVICE/citizen/id/"+id, List.class);  
    requiredResponse.setCitizens(listOfCitizens);  
    return new ResponseEntity<RequiredResponse>(requiredResponse, HttpStatus.OK);  
}
```

Eureka server

I registered both my vaccinationcenter and citizen micro services In eureka server.



The screenshot displays the Spring Eureka server web interface. The top navigation bar includes the 'spring Eureka' logo and links for 'HOME' and 'LAST 1000 SINCE STARTUP'. The main content area is divided into several sections:

- System Status:** A table showing environment details.

Environment	test
Data center	default
- Current time:** 2022-09-18T18:14:19 +0530
- Uptime:** 01:02
- Lease expiration enabled:** true
- Renews threshold:** 5
- Renews (last min):** 8

DS Replicas: A dropdown menu showing 'localhost'.

Instances currently registered with Eureka: A table listing registered services.

Application	AMIs	Availability Zones	Status
CITIZEN-SERVICE	n/a (1)	(1)	UP (1) - host.docker.internal/CITIZEN-SERVICE.8083
VACCINATION-CENTER-SERVICE	n/a (1)	(1)	UP (1) - host.docker.internal/VACCINATION-CENTER-SERVICE.8082

General Info: A table showing system metrics.

Name	Value
total-avail-memory	108mb
num-of-cpus	8
current-memory-usage	61mb (56%)
server-uptime	01:02

API gateway

previously I used 8082 port for vaccination center
and 8083 port for citizen service.

Not I can access common port 8084 .

Every api call first go to the api gateway.

Api gateway direct based on predicate.

Vaccinationcontrol... CitizenController.j... CitienSwaggerConfi...

```
1 server:
2   port: 8084
3
4 spring:
5   application:
6     name: API_GATEWAY
7
8
9 cloud:
10  gateway:
11    routes:
12      - id: CITIZEN-SERVICE
13        uri:
14          lb://CITIZEN-SERVICE
15        predicates:
16          - Path=/citizen/**
17
18      - id: VACCINATION-CENTER-SERVICE
19        uri:
20          lb://VACCINATION-CENTER-SERVICE
21        predicates:
22          - Path=/vaccinationcenter/**
23
```


GET ▼ http://localhost:8084/vaccinationcenter/allcenters

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

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ 

```
1  [
2    {
3      "id": 2,
4      "centerName": "abc",
5      "centerAddress": "hyderabad"
6    },
7    {
8      "id": 3,
9      "centerName": "abcd",
10     "centerAddress": "hyderabad"
11   },
12   {
13     "id": 4,
14     "centerName": "bvc",
15     "centerAddress": "bangloore"
16   },
17   {
18     "id": 6,
19     "centerName": "axz",
20     "centerAddress": "goa"
21   }
22 ]
```

GET

http://localhost:8084/citizen/allcitizens

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE
Key	Value

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1  [
2    {
3      "id": 1,
4      "name": "harry",
5      "vaccinationCenterId": 1
6    },
7    {
8      "id": 2,
9      "name": "harry2",
10     "vaccinationCenterId": 2
11   },
12   {
13     "id": 4,
14     "name": "sam",
15     "vaccinationCenterId": 5
16   }
17 ]
```

Fault tolerance.

If citizen micro service is down.

Then `getAllDataBasedonCenterId` method throws exception .

So fallback method (`handleCitizenDowntime`) atleast return the vaccination data.

```
@GetMapping(path = "/id/{id}")
//@HystrixCommand(fallbackMethod="handleCitizenDowntime")
public ResponseEntity<RequiredResponse> getAllDataBasedonCenterId(@PathVariable Integer id){
    RequiredResponse requiredResponse = new RequiredResponse();
    VaccinationCenter center = centerRepo.findById(id).get();
    requiredResponse.setCenter(center);

    List<Citizen> listOfCitizens = restTemplate.getForObject("http://CITIZEN-SERVICE/citizen/id/"+id, List.class);
    requiredResponse.setCitizens(listOfCitizens);
    return new ResponseEntity<RequiredResponse>(requiredResponse, HttpStatus.OK);
}

//if citizen service is down ,then above method throws exception ,this method atleast return the vaccination data.
public ResponseEntity<RequiredResponse> handleCitizenDowntime(@PathVariable Integer id){
    RequiredResponse requiredResponse = new RequiredResponse();
    VaccinationCenter center = centerRepo.findById(id).get();
    requiredResponse.setCenter(center);
    return new ResponseEntity<RequiredResponse>(requiredResponse, HttpStatus.OK);
}
```