

RLL Project

Insurance Policy Management

Group Members

1. Swetha M K
2. T R Akshay (Team Lead)
3. Tamalika Dey
4. Veeravalli Deepika
5. Abesh Das
6. Abishek Prabhu Kannan
7. Aishwarya P R
8. B Mounika
9. Dasari Manjula
10. Gangavarapu Akhila

List of Microservices

Microservices with Databases

1. Insurance-Address
2. Insurance-Admin
3. Insurance-Contact-Form
4. Insurance-Customer
5. Insurance-Customer-Queries
6. Insurance-Policy
7. Insurance-Policy-Categories

Central Feign Microservice

1. Insurance-Feign-Client

Microservices for servers

1. Insurance-Config-Server
2. Insurance-Eureka-Server

Tables Present in the Microservices (Created by ORM)

1. Insurance-Address

Database Name: insurance_address_db

Table: address

Attributes: Created by ORM

```
address (
    address_id bigint not null,
    city varchar(255),
    country varchar(255),
    postal_code varchar(255),
    state varchar(255),
    street_address varchar(255),
    primary key (address_id)
)
```

2. Insurance-Admin

Database Name: insurance_admin_db

Table: admin

Attributes: Created by ORM

```
admin_user (
    admin_id bigint not null auto_increment,
    admin_email varchar(255) not null,
    admin_name varchar(255),
    admin_password varchar(255),
    primary key (admin_id)
)
```

3. Insurance-Contact-Form

Database Name: insurance_contact_form_db

Table: contact_form

Attributes: Created by ORM

```
contact_form (
    form_id bigint not null auto_increment,
    email varchar(255),
    full_name varchar(255),
    message varchar(255),
    primary key (form_id)
)
```

4. Insurance-Customer

Database Name: insurance_customer_db

Table: customer

Attributes: Created by ORM

```
customer (
    customer_id bigint not null auto_increment,
    customer_age integer not null,
    customer_email varchar(255) not null,
    customer_gender varchar(255),
    customer_image_url varchar(255),
    customer_mobile varchar(255),
    customer_name varchar(255),
    customer_password varchar(255),
    primary key (customer_id)
)
```

5. Insurance-Customer-Queries

Database Name: insurance_question_db

Table: question

Attributes: Created by ORM

```
question (
    question_id bigint not null auto_increment,
    answer varchar(255),
    customer_id bigint,
    question varchar(255),
    primary key (question_id)
)
```

6. Insurance-Policy

Database Name: insurance_policy_db

Tables: policy, issue_policy

Attributes: Created by ORM

```
policy (
```

```

    policy_id bigint not null auto_increment,
    policy_coverage_amount decimal(19,2),
    policy_coverage_limit decimal(19,2),
    policy_coverage_period_months integer,
    policy_name varchar(255),
    policy_premium decimal(19,2),
    primary key (policy_id)
)

issue_policy (
    issue_policy_id bigint not null auto_increment,
    customer_id bigint,
    policy_date_time datetime,
    policy_id bigint,
    policy_status integer not null,
    primary key (issue_policy_id)
)

```

7. Insurance-Policy-Category

Database Name: insurance_policy_category_db

Tables: policy_category, policy_category_policy_ids (Created by
@ElementCollection annotation, dynamically)

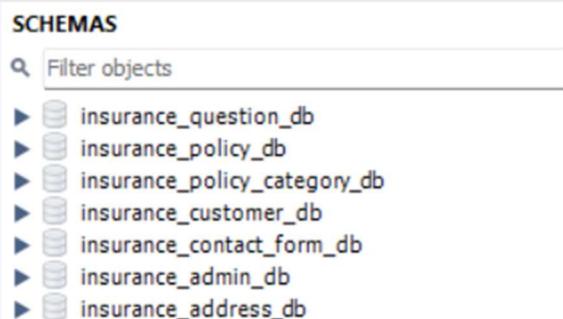
Attributes: Created by ORM

```

policy_category (
    category_id bigint not null auto_increment,
    category_name varchar(255),
    primary key (category_id)
)

policy_category_policy_ids (
    policy_category_category_id bigint not null,
    policy_ids bigint
)

```



Starting all the microservices with 2 instances of customer-service and 1 instance of each of all

The screenshot shows the Spring Eureka dashboard at localhost:8761. The top navigation bar includes links for HOME, LAST 1000 SINCE STARTUP, and other system status metrics like Current time, Uptime, Lease expiration enabled, Renews threshold, and Renews (last min). Below this is a section titled "DS Replicas" which lists various microservices and their current status (UP or DOWN) along with their respective URLs.

Application	AMIs	Availability Zones	Status
ADDRESS-SERVICE	n/a (1)	(1)	UP (1) - address-service.eb4418caf0072eb77293787f01265e20
ADMIN-SERVICE	n/a (1)	(1)	UP (1) - admin-service.9bd13b9c7ed08838102975a9e7835998
CONTACT-FORM-SERVICE	n/a (1)	(1)	UP (1) - contact-form-service.9174029ec6a78e6ff9796fe5ec4b6119
CUSTOMER-SERVICE	n/a (2)	(2)	UP (2) - customer-service.9b7dc85e0c8f07d5b82684f46e19d623 , customer-service.918f43d84875a86cbd8589cf526b2f4e
INSURANCE-APPLICATION-FEIGN	n/a (1)	(1)	UP (1) - host.docker.internal.insurance-application-feign.9000
POLICY-CATEGORY-SERVICE	n/a (1)	(1)	UP (1) - policy-category-service.188f14e80efb296d1ba8d6e37b8f757f
POLICY-SERVICE	n/a (1)	(1)	UP (1) - policy-service.4908126f38eb8dc7b6bb12d2eefd55e4
QUESTION-SERVICE	n/a (1)	(1)	UP (1) - question-service.a1bf9713cfebfc4c50fc67ee41f0bc99a

Customer Register

<http://localhost:9000/user-feign/customer/register>

The screenshot shows a POST request to <http://localhost:9000/user-feign/customer/register>. The request body is defined in JSON format, containing fields for customerEmail, customerName, customerPassword, customerAge, customerGender, customerMobile, and customerImageUrl.

```
POST http://localhost:9000/user-feign/customer/register
JSON
{
  "customerEmail": "customer2@example.com",
  "customerName": "Suresh Reddy",
  "customerPassword": "customerPass123",
  "customerAge": 35,
  "customerGender": "Male",
  "customerMobile": "+91-9123456780",
  "customerImageUrl": "https://example.com/image2.jpg"
}
```

POST <http://localhost:9000/user-feign/customer/register>

JSON	HEADERS	QUERY STRING	AUTH	ASSERTION	BODY	HEADERS	SEND ►
------	---------	--------------	------	-----------	------	---------	--------

```

1 - {
2   "customerEmail": "customer2@example.com",
3   "customerName": "Suresh Reddy",
4   "customerPassword": "customerPass123",
5   "customerAge": 35,
6   "customerGender": "Male",
7   "customerMobile": "+91-9123456780",
8   "customerImageUrl": "https://example.com/image2.jpg"
9 }
10 }
```

201 SUCCESS

Customer Login

POST <http://localhost:9000/user-feign/customer/login>

JSON	HEADERS	QUERY STRING	AUTH	ASSERTION
------	---------	--------------	------	-----------

```

1 - {
2   "customerEmail": "customer5@example.com",
3   "customerPassword": "mypwd123"
4
5 }
```

POST <http://localhost:9000/user-feign/customer/login>

JSON	HEADERS	QUERY STRING	AUTH	ASSERTION	BODY	HEADERS	SEND ►
------	---------	--------------	------	-----------	------	---------	--------

```

1 - {
2   "customerEmail": "customer5@example.com",
3   "customerPassword": "mypwd123"
4
5 }
```

```

1 - {
2   "customerId": 3,
3   "customerEmail": "customer5@example.com",
4   "customerName": "Meera Mehta",
5   "customerPassword": "mypwd123",
6   "customerAge": 54,
7   "customerGender": "Female",
8   "customerMobile": "+91-6789123450",
9   "customerImageUrl": "https://example.com/image5.jpg"
10 }
```

200 SUCCESS

Add Address

<http://localhost:9000/user-feign/customer/add-address/3>

POST <http://localhost:9000/user-feign/customer/add-address/3>

JSON	HEADERS	QUERY STRING	AUTH	ASSERTION
------	---------	--------------	------	-----------

```

1 - {
2   "streetAddress": "456, Park Avenue",
3   "city": "Delhi",
4   "state": "Delhi",
5   "postalCode": "110001",
6   "country": "India"
7
8 }
```

The screenshot shows a Postman interface with a successful response (201 SUCCESS). The request URL is `http://localhost:9000/user-feign/customer/3`. The request body contains the following JSON:

```

1- {
2-   "streetAddress": "456, Park Avenue",
3-   "city": "Delhi",
4-   "state": "Delhi",
5-   "postalCode": "110001",
6-   "country": "India"
7- }
8-
1- {
2-   "customerId": 3,
3-   "customerEmail": "customers@example.com",
4-   "customerName": "Neera Mehta",
5-   "customerPassword": "mypwd123",
6-   "customerAge": 54,
7-   "customerGender": "Female",
8-   "customerMobile": "+91-6789123450",
9-   "customerImageUrl": "https://example.com/image5.jpg",
10-
11-   "customerAddress": {
12-     "addressId": 3,
13-     "streetAddress": "456, Park Avenue",
14-     "city": "Delhi",
15-     "state": "Delhi",
16-     "postalCode": "110001",
17-     "country": "India"
18-   }

```

Update Customer Details with address

<http://localhost:9000/user-feign/customer/3>

The screenshot shows a Postman interface with a successful response (200 SUCCESS). The request URL is `http://localhost:9000/user-feign/customer/3`. The request body contains the following JSON:

```

1- {
2-   "customerId": 3,
3-   "customerEmail": "customer5@example.com",
4-   "customerName": "Meera Mehta",
5-   "customerPassword": "mypwd123",
6-   "customerAge": 57,
7-   "customerGender": "Female",
8-   "customerMobile": "+91-9967891234",
9-   "customerImageUrl": "https://example1234.com/image5.jpg",
10-
11-   "customerAddress": {
12-     "addressId": 3,
13-     "streetAddress": "11456, Park Avenue",
14-     "city": "Delhi",
15-     "state": "Delhi",
16-     "postalCode": "110001",
17-     "country": "India"
18-   }

```

The screenshot shows a Postman interface with a successful response (200 SUCCESS). The request URL is `http://localhost:9000/user-feign/customer/3`. The request body contains the following JSON:

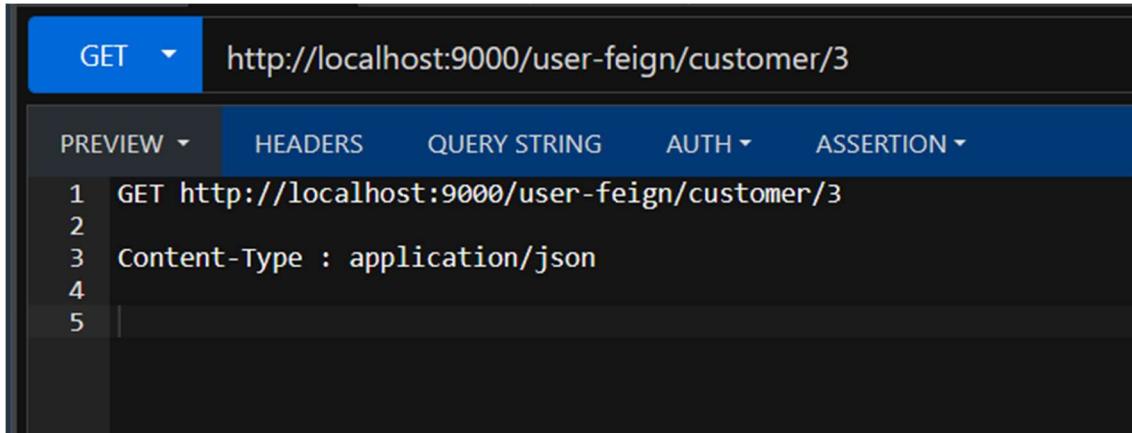
```

1- {
2-   "customerId": 3,
3-   "customerEmail": "customers@example.com",
4-   "customerName": "Neera Mehta",
5-   "customerPassword": "mypwd123",
6-   "customerAge": 57,
7-   "customerGender": "Female",
8-   "customerMobile": "+91-9967891234",
9-   "customerImageUrl": "https://example1234.com/image5.jpg",
10-
11-   "customerAddress": {
12-     "addressId": 3,
13-     "streetAddress": "11456, Park Avenue",
14-     "city": "Delhi",
15-     "state": "Delhi",
16-     "postalCode": "110001",
17-     "country": "India"
18-   }

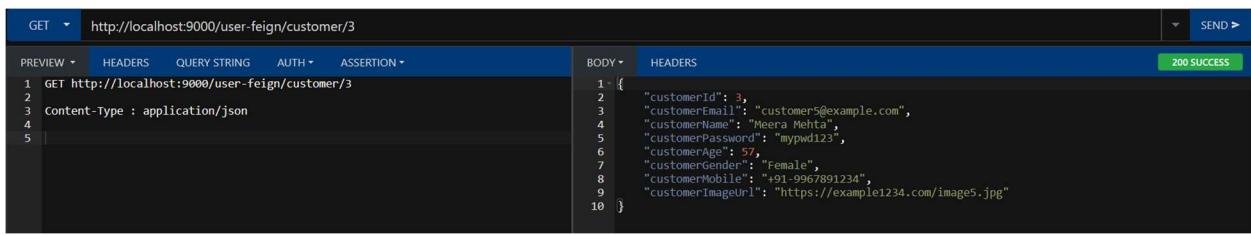
```

Get Basic Customer Details

<http://localhost:9000/user-feign/customer/3>



```
1 GET http://localhost:9000/user-feign/customer/3
2
3 Content-Type : application/json
4
5
```

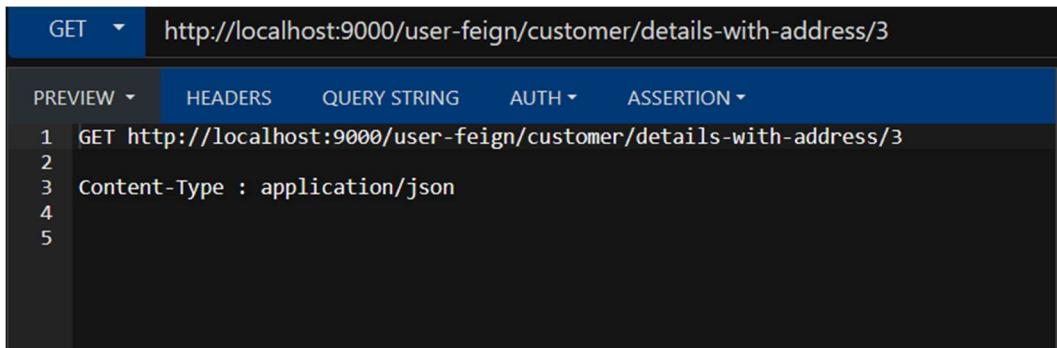


200 SUCCESS

```
1 - {
2     "customerId": 3,
3     "customerEmail": "customer5@example.com",
4     "customerName": "Meera Mehta",
5     "customerPassword": "mypwd123",
6     "customerAge": 57,
7     "customerGender": "Female",
8     "customerMobile": "+91-9967891234",
9     "customerImageURL": "https://example1234.com/image5.jpg"
10 }
```

Get Details along with Address

<http://localhost:9000/user-feign/customer/details-with-address/3>



```
1 GET http://localhost:9000/user-feign/customer/details-with-address/3
2
3 Content-Type : application/json
4
5
```

GET ▾ http://localhost:9000/user-feign/customer/details-with-address/3

PREVIEW	HEADERS	QUERY STRING	AUTH	ASSERTION	
<pre>1 GET http://localhost:9000/user-feign/customer/details-with-address/3 2 3 Content-Type : application/json 4 5</pre>					200 SUCCESS
	BODY ▾	HEADERS			
	<pre>1 { 2 "customerId": 3, 3 "customerEmail": "customers@example.com", 4 "customerName": "Meena Mehta", 5 "customerPassword": "mypwd123", 6 "customerAge": 57, 7 "customerGender": "Female", 8 "customerMobile": "+91-9967891234", 9 "customerImageUrl": "https://example1234.com/images5.jpg", 10 "customerAddress": { 11 "addressId": 3, 12 "streetAddress": "11456, Park Avenue", 13 "city": "Delhi", 14 "state": "Delhi", 15 "postalCode": "110001", 16 "country": "India" 17 } 18 }</pre>				

Admin Register

<http://localhost:9000/admin-feign/admin/register>

POST ▾ http://localhost:9000/admin-feign/admin/register

JSON	HEADERS	QUERY STRING	AUTH	ASSERTION	
<pre>1 { 2 "adminEmail": "admin1@example.com", 3 "adminName": "Ravi Kumar", 4 "adminPassword": "admin@123" 5 } 6</pre>					

POST ▾ http://localhost:9000/admin-feign/admin/register

JSON	HEADERS	QUERY STRING	AUTH	ASSERTION	
<pre>1 { 2 "adminEmail": "admin1@example.com", 3 "adminName": "Ravi Kumar", 4 "adminPassword": "admin@123" 5 }</pre>	BODY ▾	HEADERS			201 SUCCESS
	<pre>1 { 2 "adminId": 1, 3 "adminEmail": "admin1@example.com", 4 "adminName": "Ravi Kumar", 5 "adminPassword": "admin@123" 6 }</pre>				

Admin Login

<http://localhost:9000/admin-feign/admin/login>

POST ▾ http://localhost:9000/admin-feign/admin/login

JSON	HEADERS	QUERY STRING	AUTH	ASSERTION	
<pre>1 { 2 "adminEmail": "admin1@example.com", 3 "adminPassword": "admin@123" 4 } 5</pre>					

POST http://localhost:9000/admin-feign/admin/login

JSON

```
1 { "adminEmail": "admin1@example.com",  
2 "adminPassword": "admin@123"  
3 }  
4  
5
```

BODY

```
1 {  
2   "adminId": 1,  
3   "adminEmail": "admin1@example.com",  
4   "adminName": "Ramesh Kumar",  
5   "adminPassword": "updatedadmin@123"  
6 }
```

HEADERS

200 SUCCESS

Update Admin

<http://localhost:9000/admin-feign/admin/update/1>

PUT http://localhost:9000/admin-feign/admin/update/1

JSON

```
1 {  
2   "adminId": 1,  
3   "adminEmail": "admin1@example.com",  
4   "adminName": "Ramesh Kumar",  
5   "adminPassword": "updatedadmin@123"  
6 }
```

HEADERS

QUERY STRING

AUTH

ASSERTION

PUT http://localhost:9000/admin-feign/admin/update/1

JSON

```
1 {  
2   "adminId": 1,  
3   "adminEmail": "admin1@example.com",  
4   "adminName": "Ramesh Kumar",  
5   "adminPassword": "updatedadmin@123"  
6 }
```

BODY

```
1 {  
2   "adminId": 1,  
3   "adminEmail": "admin1@example.com",  
4   "adminName": "Ramesh Kumar",  
5   "adminPassword": "updatedadmin@123"  
6 }
```

HEADERS

200 SUCCESS

Get All Registered Admins

<http://localhost:9000/admin-feign/admin>

GET http://localhost:9000/admin-feign/admin

PREVIEW

```
1 GET http://localhost:9000/admin-feign/admin  
2  
3 Content-Type : application/json  
4  
5
```

HEADERS

QUERY STRING

AUTH

ASSERTION

GET http://localhost:9000/admin-feign/admin

PREVIEW	HEADERS	QUERY STRING	AUTH	ASSERTION
1 GET http://localhost:9000/admin-feign/admin 2 3 Content-Type : application/json 4 5	1 [2 { 3 "adminId": 1, 4 "adminEmail": "admin1@example.com", 5 "adminName": "Ramesh Kumar", 6 "adminPassword": "updatedadmin@123" 7 }, 8 { 9 "adminId": 2, 10 "adminEmail": "admin2@example.com", 11 "adminName": "Priya Sharma", 12 "adminPassword": "secureAdmin1" 13 }, 14 { 15 "adminId": 3, 16 "adminEmail": "admin3@example.com", 17 "adminName": "Ankit Patel", 18 "adminPassword": "adminPass123" 19 } 20]	1 [2 { 3 "adminId": 1, 4 "adminEmail": "admin1@example.com", 5 "adminName": "Ramesh Kumar", 6 "adminPassword": "updatedadmin@123" 7 }, 8 { 9 "adminId": 2, 10 "adminEmail": "admin2@example.com", 11 "adminName": "Priya Sharma", 12 "adminPassword": "secureAdmin1" 13 }, 14 { 15 "adminId": 3, 16 "adminEmail": "admin3@example.com", 17 "adminName": "Ankit Patel", 18 "adminPassword": "adminPass123" 19 } 20]	1 [2 { 3 "adminId": 1, 4 "adminEmail": "admin1@example.com", 5 "adminName": "Ramesh Kumar", 6 "adminPassword": "updatedadmin@123" 7 }, 8 { 9 "adminId": 2, 10 "adminEmail": "admin2@example.com", 11 "adminName": "Priya Sharma", 12 "adminPassword": "secureAdmin1" 13 }, 14 { 15 "adminId": 3, 16 "adminEmail": "admin3@example.com", 17 "adminName": "Ankit Patel", 18 "adminPassword": "adminPass123" 19 } 20]	200 SUCCESS

Getting an Admin by ID

<http://localhost:9000/admin-feign/admin/2>

GET http://localhost:9000/admin-feign/admin/2

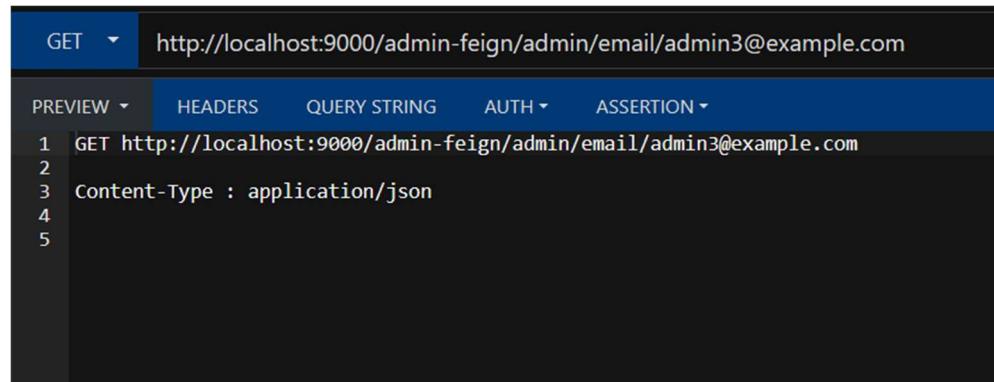
PREVIEW	HEADERS	QUERY STRING	AUTH	ASSERTION
1 GET http://localhost:9000/admin-feign/admin/2 2 3 Content-Type : application/json 4 5	1 [2 { 3 "adminId": 2, 4 "adminEmail": "admin2@example.com", 5 "adminName": "Priya Sharma", 6 "adminPassword": "secureAdmin1" 7 }]	1 [2 { 3 "adminId": 2, 4 "adminEmail": "admin2@example.com", 5 "adminName": "Priya Sharma", 6 "adminPassword": "secureAdmin1" 7 }]	1 [2 { 3 "adminId": 2, 4 "adminEmail": "admin2@example.com", 5 "adminName": "Priya Sharma", 6 "adminPassword": "secureAdmin1" 7 }]	200 SUCCESS

GET http://localhost:9000/admin-feign/admin/2

PREVIEW	HEADERS	QUERY STRING	AUTH	ASSERTION
1 GET http://localhost:9000/admin-feign/admin/2 2 3 Content-Type : application/json 4 5	1 [2 { 3 "adminId": 2, 4 "adminEmail": "admin2@example.com", 5 "adminName": "Priya Sharma", 6 "adminPassword": "secureAdmin1" 7 }]	1 [2 { 3 "adminId": 2, 4 "adminEmail": "admin2@example.com", 5 "adminName": "Priya Sharma", 6 "adminPassword": "secureAdmin1" 7 }]	1 [2 { 3 "adminId": 2, 4 "adminEmail": "admin2@example.com", 5 "adminName": "Priya Sharma", 6 "adminPassword": "secureAdmin1" 7 }]	200 SUCCESS

Getting an Admin by email

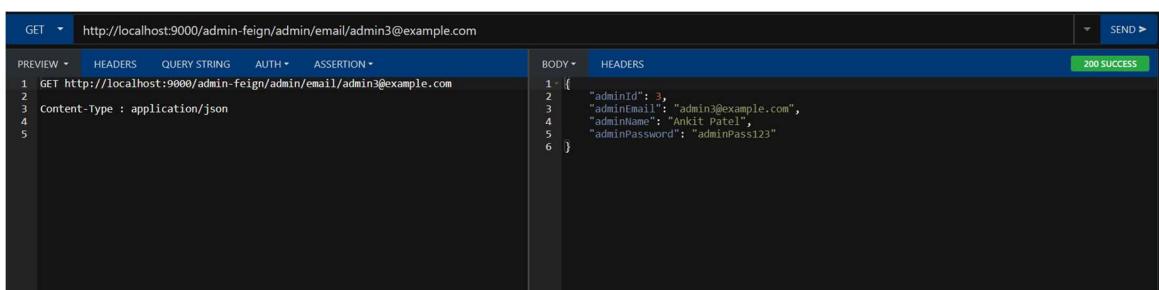
<http://localhost:9000/admin-feign/admin/email/admin3@example.com>



```
GET http://localhost:9000/admin-feign/admin/email/admin3@example.com
```

PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

```
1 GET http://localhost:9000/admin-feign/admin/email/admin3@example.com
2
3 Content-Type : application/json
4
5
```



GET http://localhost:9000/admin-feign/admin/email/admin3@example.com

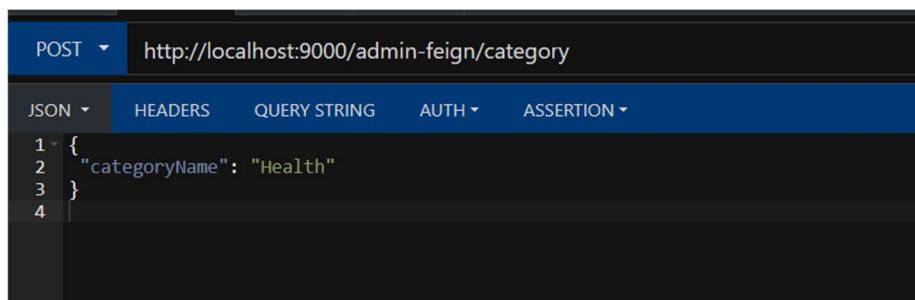
PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾ BODY ▾ HEADERS 200 SUCCESS

```
1 GET http://localhost:9000/admin-feign/admin/email/admin3@example.com
2
3 Content-Type : application/json
4
5
6 }
```

```
1 {
2   "adminId": 3,
3   "adminEmail": "admin@example.com",
4   "adminName": "Ankit Patel",
5   "adminPassword": "adminPass123"
6 }
```

Adding a new Policy Category by Admin

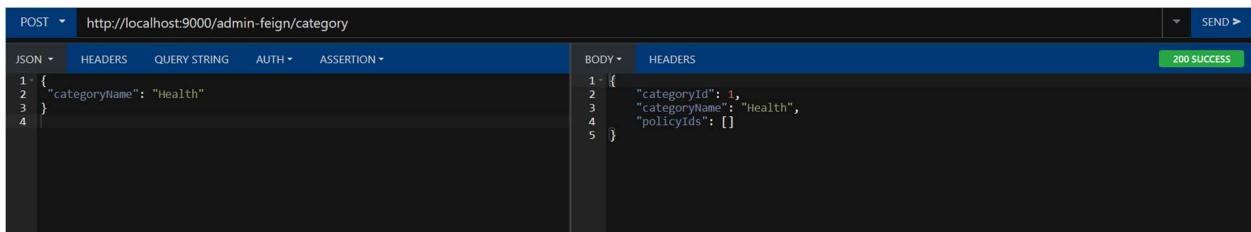
<http://localhost:9000/admin-feign/category>



```
POST http://localhost:9000/admin-feign/category
```

JSON ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

```
1 {
2   "categoryName": "Health"
3 }
4
```



POST http://localhost:9000/admin-feign/category

JSON ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾ BODY ▾ HEADERS 200 SUCCESS

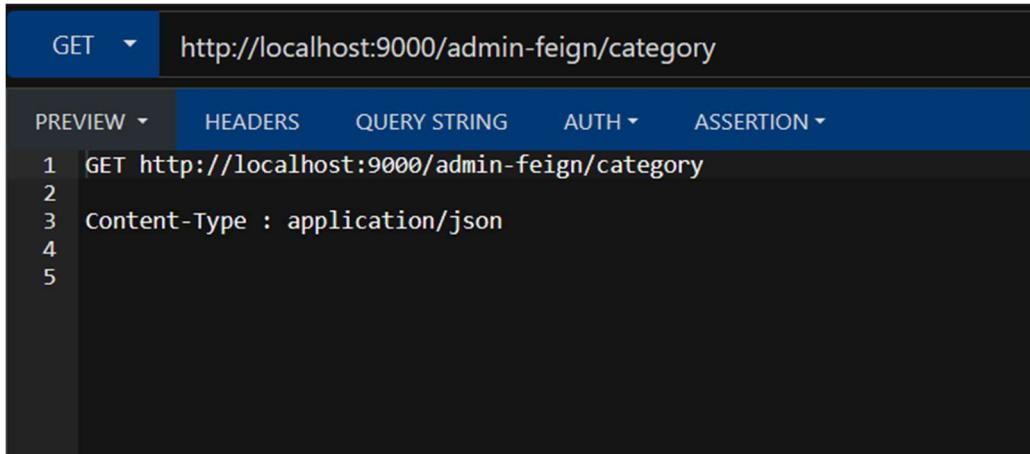
```
1 {
2   "categoryName": "Health"
3 }
4
```

```
1 {
2   "categoryId": 1,
3   "categoryName": "Health",
4   "policyIds": []
5 }
```

Initially, no policies are registered under a category

Get all Categories

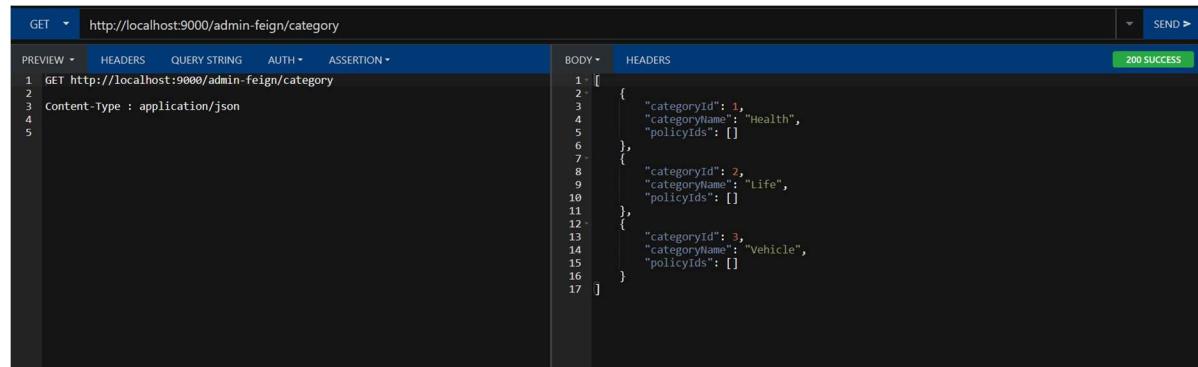
<http://localhost:9000/admin-feign/category>



GET ▾ http://localhost:9000/admin-feign/category

PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

```
1 GET http://localhost:9000/admin-feign/category
2
3 Content-Type : application/json
4
5
```



GET ▾ http://localhost:9000/admin-feign/category

PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

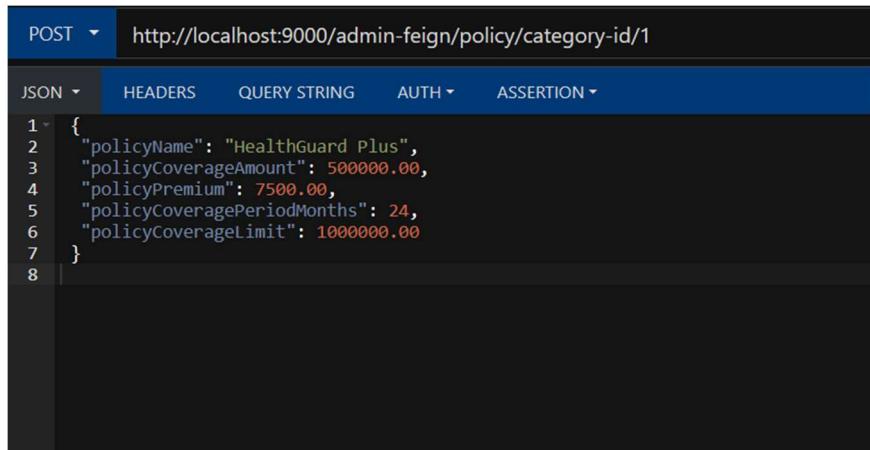
```
1 GET http://localhost:9000/admin-feign/category
2
3 Content-Type : application/json
4
5
```

BODY ▾ HEADERS 200 SUCCESS

```
1 [ {
2   "categoryId": 1,
3   "categoryName": "Health",
4   "policyIds": []
5 },
6   {
7     "categoryId": 2,
8     "categoryName": "Life",
9     "policyIds": []
10 },
11   {
12     "categoryId": 3,
13     "categoryName": "Vehicle",
14     "policyIds": []
15 }
16 ]
```

Adding Policy to a category

<http://localhost:9000/admin-feign/policy/category-id/1>



POST ▾ http://localhost:9000/admin-feign/policy/category-id/1

JSON ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

```
1 {
2   "policyName": "HealthGuard Plus",
3   "policyCoverageAmount": 500000.00,
4   "policyPremium": 7500.00,
5   "policyCoveragePeriodMonths": 24,
6   "policyCoverageLimit": 1000000.00
7 }
```

POST ▾ http://localhost:9000/admin-feign/policy/category-id/1

JSON ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

```

1 - {
2   "policyName": "HealthGuard Plus",
3   "policyCoverageAmount": 500000.00,
4   "policyPremium": 7500.00,
5   "policyCoveragePeriodMonths": 24,
6   "policyCoverageLimit": 1000000.00
7 }
8

```

BODY ▾ HEADERS

```

1 - {
2   "policyId": 1,
3   "policyName": "HealthGuard Plus",
4   "policyCoverageAmount": 500000,
5   "policyPremium": 7500,
6   "policyCoveragePeriodMonths": 24,
7   "policyCoverageLimit": 1000000
8 }

```

SEND ▶ 201 SUCCESS

Get all Policies

<http://localhost:9000/admin-feign/policy>

GET ▾ http://localhost:9000/admin-feign/policy

PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

```

1 GET http://localhost:9000/admin-feign/policy
2
3 Content-Type : application/json
4
5

```

GET ▾ http://localhost:9000/admin-feign/policy

PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

```

1 GET http://localhost:9000/admin-feign/policy
2
3 Content-Type : application/json
4
5

```

BODY ▾ HEADERS

```

1 - [
2   {
3     "policyId": 1,
4     "policyName": "HealthGuard Plus",
5     "policyCoverageAmount": 500000,
6     "policyPremium": 7500,
7     "policyCoveragePeriodMonths": 24,
8     "policyCoverageLimit": 1000000
9   },
10  {
11    "policyId": 2,
12    "policyName": "Wellness Assurance",
13    "policyCoverageAmount": 300000,
14    "policyPremium": 6000,
15    "policyCoveragePeriodMonths": 12,
16    "policyCoverageLimit": 500000
17  },
18  {
19    "policyId": 3,
20    "policyName": "Family Wellness Plan",
21    "policyCoverageAmount": 800000,
22    "policyPremium": 12000,
23    "policyCoveragePeriodMonths": 36,
24    "policyCoverageLimit": 1500000
25  },
26  {
27    "policyId": 4,
28    "policyName": "LifeSecure Plus",
29    "policyCoverageAmount": 1000000,
30    "policyPremium": 8500,
31    "policyCoveragePeriodMonths": 240,
32    "policyCoverageLimit": 2000000
33  },
34  {
35    "policyId": 5,
36    "policyName": "Education Savings Plan",
37    "policyCoverageAmount": 500000,
38    "policyPremium": 6000,
39    "policyCoveragePeriodMonths": 120,
40    "policyCoverageLimit": 1000000
41  },
42  {
43    "policyId": 6,
44    "policyName": "AutoShield Plus",
45    "policyCoverageAmount": 300000,
46    "policyPremium": 4500,
47    "policyCoveragePeriodMonths": 36
48  }
49 ]

```

SEND ▶ 200 SUCCESS

After adding policies, checking the categories for policy IDs

<http://localhost:9000/admin-feign/category>

The screenshot shows a REST client interface with a blue header bar. The URL 'http://localhost:9000/admin-feign/category' is entered in the address bar. Below the address bar, there are tabs for 'PREVIEW', 'HEADERS', 'QUERY STRING', 'AUTH', and 'ASSERTION'. The 'PREVIEW' tab is selected, displaying the following code:

```
1 GET http://localhost:9000/admin-feign/category
2
3 Content-Type : application/json
4
5
```

The screenshot shows a REST client interface with a blue header bar. The URL 'http://localhost:9000/admin-feign/category' is entered in the address bar. Below the address bar, there are tabs for 'PREVIEW', 'HEADERS', 'QUERY STRING', 'AUTH', and 'ASSERTION'. The 'PREVIEW' tab is selected, displaying the following code:

```
1 GET http://localhost:9000/admin-feign/category
2
3 Content-Type : application/json
4
5
```

On the right side, under 'BODY', the response content is shown as JSON:

```
1 [ [
2   {
3     "categoryId": 1,
4     "categoryName": "Health",
5     "policyIds": [
6       2,
7       1,
8       3
9     ]
10   },
11   {
12     "categoryId": 2,
13     "categoryName": "Life",
14     "policyIds": [
15       4,
16       5
17     ]
18   },
19   {
20     "categoryId": 3,
21     "categoryName": "Vehicle",
22     "policyIds": [
23       8,
24       6,
25       7,
26       9
27     ]
28   }
29 ]
```

A green button at the top right indicates a '200 SUCCESS' response.

Update a Policy's details by ID

<http://localhost:9000/admin-feign/policy/4>

The screenshot shows a REST client interface with a blue header bar. The URL 'http://localhost:9000/admin-feign/policy/4' is entered in the address bar. Below the address bar, there are tabs for 'PUT', 'HEADERS', 'QUERY STRING', 'AUTH', and 'ASSERTION'. The 'PUT' tab is selected, and the 'JSON' tab is also selected. The body of the request is a JSON object:

```
1 {
2   "policyName": "LifeSecure Plus NEW",
3   "policyCoverageAmount": 1500000,
4   "policyPremium": 9500,
5   "policyCoveragePeriodMonths": 300,
6   "policyCoverageLimit": 2500000
7 }
```

PUT http://localhost:9000/admin-feign/policy/4

JSON HEADERS QUERY STRING AUTH ASSERTION

```
1 [{}]
2   "policyName": "LifeSecure Plus NEW",
3   "policyCoverageAmount": 1500000,
4   "policyPremium": 9500,
5   "policyCoveragePeriodMonths": 300,
6   "policyCoverageLimit": 2500000
7 ]
```

BODY HEADERS 200 SUCCESS

```
1 Policy with ID 4 updated successfully
```

Get Policy by ID

<http://localhost:9000/admin-feign/policy/5>

GET http://localhost:9000/admin-feign/policy/5

PREVIEW HEADERS QUERY STRING AUTH ASSERTION

```
1 GET http://localhost:9000/admin-feign/policy/5
2
3 Content-Type : application/json
4
5
```

GET http://localhost:9000/admin-feign/policy/5

PREVIEW HEADERS QUERY STRING AUTH ASSERTION

```
1 GET http://localhost:9000/admin-feign/policy/5
2
3 Content-Type : application/json
4
5
```

BODY HEADERS 200 SUCCESS

```
1 [{}]
2   "policyId": 5,
3   "policyName": "Education Savings Plan",
4   "policyCoverageAmount": 500000,
5   "policyPremium": 6000,
6   "policyCoveragePeriodMonths": 120,
7   "policyCoverageLimit": 1000000
8 ]
```

Get Policy By Name Phrase match

<http://localhost:9000/admin-feign/policy/name-like/plus>

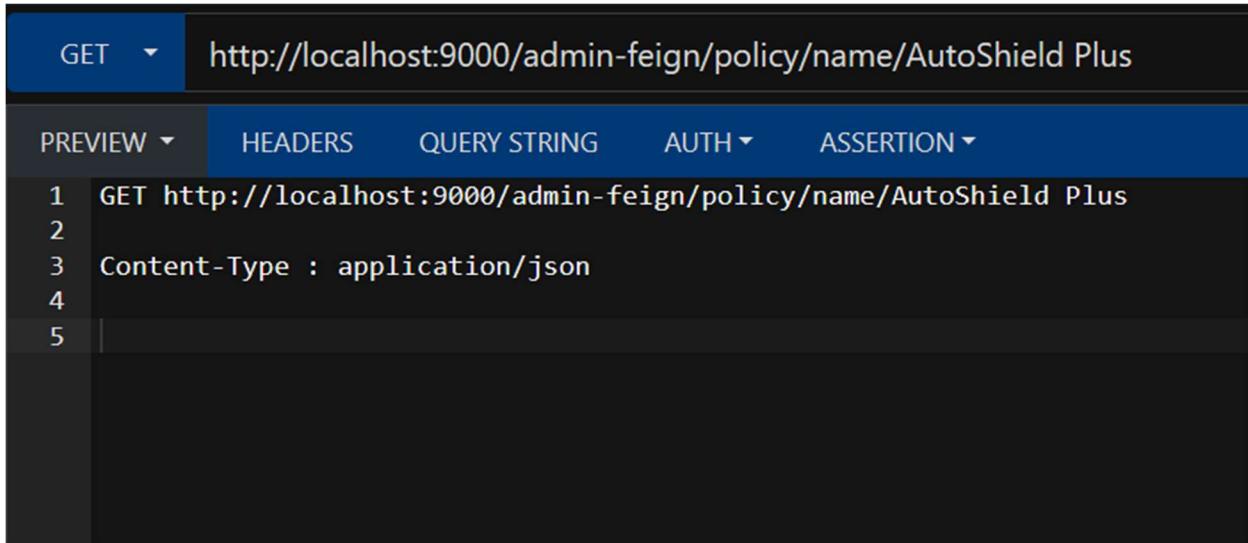
```
GET ▾ http://localhost:9000/admin-feign/policy/name-like/plus
PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾
1 GET http://localhost:9000/admin-feign/policy/name-like/plus
2
3 Content-Type : application/json
4
5
```

```
GET ▾ http://localhost:9000/admin-feign/policy/name-like/plus ▾ SEND ▾ 200 SUCCESS
PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾
1 GET http://localhost:9000/admin-feign/policy/name-like/plus
2
3 Content-Type : application/json
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
```

BODY ▾	HEADERS
[{"policyId": 1, "policyName": "HealthGuard Plus", "policyCoverageAmount": 500000, "policyPremium": 7500, "policyCoveragePeriodMonths": 24, "policyCoverageLimit": 1000000}, {"policyId": 4, "policyName": "LifeSecure Plus NEW", "policyCoverageAmount": 1500000, "policyPremium": 9500, "policyCoveragePeriodMonths": 300, "policyCoverageLimit": 2500000}, {"policyId": 6, "policyName": "AutoShield Plus", "policyCoverageAmount": 300000, "policyPremium": 4500, "policyCoveragePeriodMonths": 36, "policyCoverageLimit": 500000}], []	

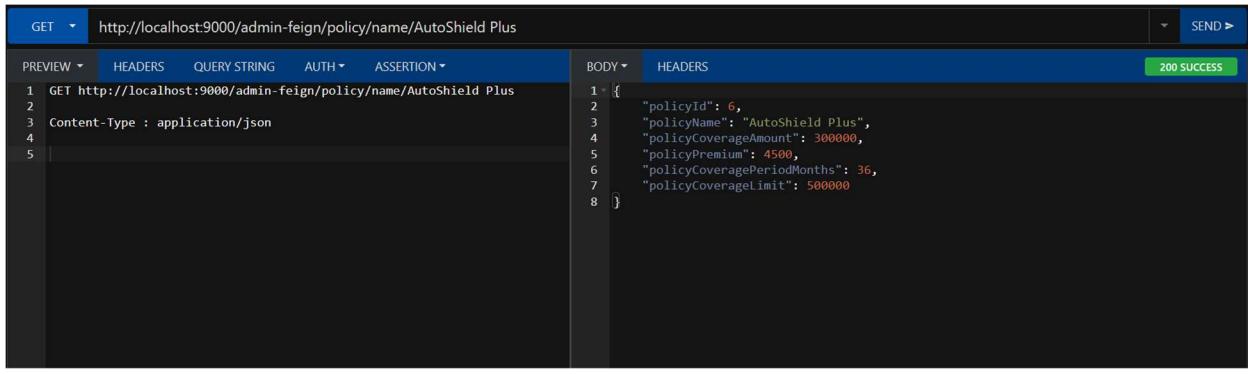
Get Policy By Full Policy Name

<http://localhost:9000/admin-feign/policy/name/AutoShield Plus>



The screenshot shows the Postman interface with a blue header bar. On the left, it says "GET" and "http://localhost:9000/admin-feign/policy/name/AutoShield Plus". Below the header are tabs: "PREVIEW", "HEADERS", "QUERY STRING", "AUTH", and "ASSERTION". The "PREVIEW" tab is selected. In the main body area, there is a code editor with the following content:

```
1 GET http://localhost:9000/admin-feign/policy/name/AutoShield Plus
2
3 Content-Type : application/json
4
5
```



The screenshot shows the Postman interface with a blue header bar. On the left, it says "GET" and "http://localhost:9000/admin-feign/policy/name/AutoShield Plus". Below the header are tabs: "PREVIEW", "HEADERS", "QUERY STRING", "AUTH", and "ASSERTION". The "PREVIEW" tab is selected. In the main body area, there is a code editor with the following content:

```
1 GET http://localhost:9000/admin-feign/policy/name/AutoShield Plus
2
3 Content-Type : application/json
4
5
```

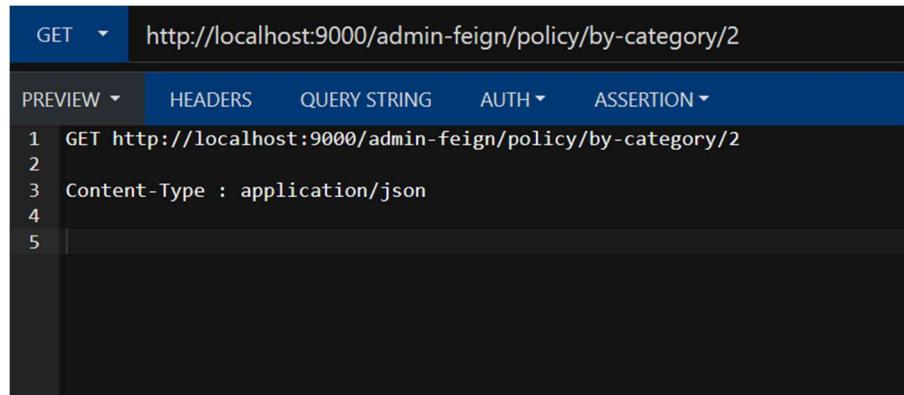
On the right side, there is a "BODY" tab with a "JSON" sub-tab. It shows the response body as a JSON object:

```
1 {
2   "policyId": 6,
3   "policyName": "AutoShield Plus",
4   "policyCoverageAmount": 300000,
5   "policyPremium": 4500,
6   "policyCoveragePeriodMonths": 36,
7   "policyCoverageLimit": 500000
8 }
```

A green button at the top right of the preview panel says "200 SUCCESS".

Get Policy by Category ID

<http://localhost:9000/admin-feign/policy/by-category/2>



The screenshot shows the Postman interface with a blue header bar. On the left, it says "GET" and "http://localhost:9000/admin-feign/policy/by-category/2". Below the header are tabs: "PREVIEW", "HEADERS", "QUERY STRING", "AUTH", and "ASSERTION". The "PREVIEW" tab is selected. In the main body area, there is a code editor with the following content:

```
1 GET http://localhost:9000/admin-feign/policy/by-category/2
2
3 Content-Type : application/json
4
5
```

GET http://localhost:9000/admin-feign/policy/by-category/2

PREVIEW	HEADERS	QUERY STRING	AUTH	ASSERTION
1 GET http://localhost:9000/admin-feign/policy/by-category/2 2 3 Content-Type : application/json 4 5	BODY	HEADERS	200 SUCCESS	
	[2 { 3 "policyId": 4, 4 "policyName": "LifeSecure Plus NEW", 5 "policyCoverageAmount": 1500000, 6 "policyPremium": 9500, 7 "policyCoveragePeriodMonths": 300, 8 "policyCoverageLimit": 2500000 9 }, 10 { 11 "policyId": 5, 12 "policyName": "Education Savings Plan", 13 "policyCoverageAmount": 500000, 14 "policyPremium": 6000, 15 "policyCoveragePeriodMonths": 120, 16 "policyCoverageLimit": 1000000 17 } 18]			

Delete a Policy by ID

<http://localhost:9000/admin-feign/policy/4>

DELETE http://localhost:9000/admin-feign/policy/4

PREVIEW	HEADERS	QUERY STRING	AUTH	ASSERTION
1 DELETE http://localhost:9000/admin-feign/policy/4 2 3 Content-Type : application/json 4 5				

DELETE http://localhost:9000/admin-feign/policy/4

PREVIEW	HEADERS	QUERY STRING	AUTH	ASSERTION
1 DELETE http://localhost:9000/admin-feign/policy/4 2 3 Content-Type : application/json 4 5	BODY	HEADERS	200 SUCCESS	
	[1]			

Checking list of policies and categories for policy of ID 4 getting removed

GET http://localhost:9000/admin-feign/policy

PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

```
1 GET http://localhost:9000/admin-feign/policy
2
3 Content-Type : application/json
4
5
```

BODY ▾ HEADERS

```
2   "policyCoverageAmount": 500000,
3   "policyPremium": 7500,
4   "policyCoveragePeriodMonths": 24,
5   "policyCoverageLimit": 1000000
6 },
7 {
8   "policyId": 2,
9   "policyName": "Wellness Assurance",
10  "policyCoverageAmount": 300000,
11  "policyPremium": 6000,
12  "policyCoveragePeriodMonths": 12,
13  "policyCoverageLimit": 500000
14 },
15 {
16   "policyId": 3,
17   "policyName": "Family Wellness Plan",
18   "policyCoverageAmount": 800000,
19   "policyPremium": 9000,
20   "policyCoveragePeriodMonths": 36,
21   "policyCoverageLimit": 1500000
22 },
23 {
24   "policyId": 5,
25   "policyName": "Education Savings Plan",
26   "policyCoverageAmount": 500000,
27   "policyPremium": 6000,
28   "policyCoveragePeriodMonths": 120,
29   "policyCoverageLimit": 1000000
30 },
31 {
32   "policyId": 6,
33   "policyName": "AutoShield Plus",
34   "policyCoverageAmount": 300000,
```

200 SUCCESS

GET http://localhost:9000/admin-feign/category

PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

```
1 GET http://localhost:9000/admin-feign/category
2
3 Content-Type : application/json
4
5
```

BODY ▾ HEADERS

```
1 [
2   {
3     "categoryId": 1,
4     "categoryName": "Health",
5     "policyIds": [
6       2,
7       1,
8       3
9     ]
10   },
11   {
12     "categoryId": 2,
13     "categoryName": "Life",
14     "policyIds": [
15       5
16     ]
17   },
18   {
19     "categoryId": 3,
20     "categoryName": "Vehicle",
21     "policyIds": [
22       8,
23       6,
24       7,
25       9
26     ]
27 }
```

200 SUCCESS

Deleting a category by ID

<http://localhost:9000/admin-feign/category/1>

The screenshot shows a Postman request configuration. The method is set to 'DELETE' and the URL is 'http://localhost:9000/admin-feign/category/1'. The 'PREVIEW' tab is selected, showing the raw request code:

```
1 DELETE http://localhost:9000/admin-feign/category/1
2
3 Content-Type : application/json
4
5
```

The screenshot shows the response from the previous request. The status bar indicates '200 SUCCESS'. The 'BODY' tab is selected, showing the JSON response:

```
1
```

Checking list of categories and Policies

The screenshot shows a Postman request configuration. The method is set to 'GET' and the URL is 'http://localhost:9000/admin-feign/category'. The 'PREVIEW' tab is selected, showing the raw request code:

```
1 GET http://localhost:9000/admin-feign/category
2
3 Content-Type : application/json
4
5
```

The 'BODY' tab is selected, showing the JSON response:

```
1 - [
2   {
3     "categoryId": 2,
4     "categoryName": "Life",
5     "policyIds": [
6       5
7     ]
8   },
9   {
10    "categoryId": 3,
11    "categoryName": "Vehicle",
12    "policyIds": [
13      8,
14      6,
15      7,
16      9
17    ]
18  }
19]
```

GET http://localhost:9000/admin-feign/policy

PREVIEW	HEADERS	QUERY STRING	AUTH	ASSERTION	BODY	HEADERS	SEND ▶
1 GET http://localhost:9000/admin-feign/policy					1 [2 { 3 "policyId": 5, 4 "policyName": "Education Savings Plan", 5 "policyCoverageAmount": 500000, 6 "policyPremium": 6000, 7 "policyCoveragePeriodMonths": 120, 8 "policyCoverageLimit": 1000000 9 }, 10 { 11 "policyId": 6, 12 "policyName": "AutoShield Plus", 13 "policyCoverageAmount": 300000, 14 "policyPremium": 4500, 15 "policyCoveragePeriodMonths": 36, 16 "policyCoverageLimit": 500000 17 }, 18 { 19 "policyId": 7, 20 "policyName": "Comprehensive Vehicle Insurance", 21 "policyCoverageAmount": 600000, 22 "policyPremium": 7500, 23 "policyCoveragePeriodMonths": 60, 24 "policyCoverageLimit": 1000000 25 }, 26 { 27 "policyId": 8, 28 "policyName": "Roadside Assistance", 29 "policyCoverageAmount": 300000, 30 "policyPremium": 2500, 31 "policyCoveragePeriodMonths": 36, 32 "policyCoverageLimit": 400000 33 }, 34 { 35 "policyId": 9, 36 "policyName": "Collision Coverage", 37 "policyCoverageAmount": 1000000, 38 "policyPremium": 7000, 39 "policyCoveragePeriodMonths": 24, 40 "policyCoverageLimit": 1500000 41 }] 42]		200 SUCCESS

(Policies corresponding to Category ID 1 got deleted)

Updating a category name

<http://localhost:9000/admin-feign/category/update-name/3/Automobile>

PUT http://localhost:9000/admin-feign/category/update-name/3/Automobile

PREVIEW	HEADERS	QUERY STRING	AUTH	ASSERTION
1 PUT http://localhost:9000/admin-feign/category/update-name/3/Automobile				1 2 3 Content-Type : application/json

PUT http://localhost:9000/admin-feign/category/update-name/3/Automobile

PREVIEW HEADERS QUERY STRING AUTH ASSERTION

```
1 PUT http://localhost:9000/admin-feign/category/update-name/3/Automobile
2
3 Content-Type : application/json
4
5
```

BODY HEADERS

```
1 {
2     "categoryId": 3,
3     "categoryName": "Automobile",
4     "policyIds": [
5         8,
6         6,
7         7,
8         9
9     ]
10 }
```

200 SUCCESS

Get Category By ID

<http://localhost:9000/admin-feign/category/2>

GET http://localhost:9000/admin-feign/category/2

PREVIEW HEADERS QUERY STRING AUTH ASSERTION

```
1 GET http://localhost:9000/admin-feign/category/2
2
3 Content-Type : application/json
4
5
```

GET http://localhost:9000/admin-feign/category/2

PREVIEW HEADERS QUERY STRING AUTH ASSERTION

```
1 GET http://localhost:9000/admin-feign/category/2
2
3 Content-Type : application/json
4
5
```

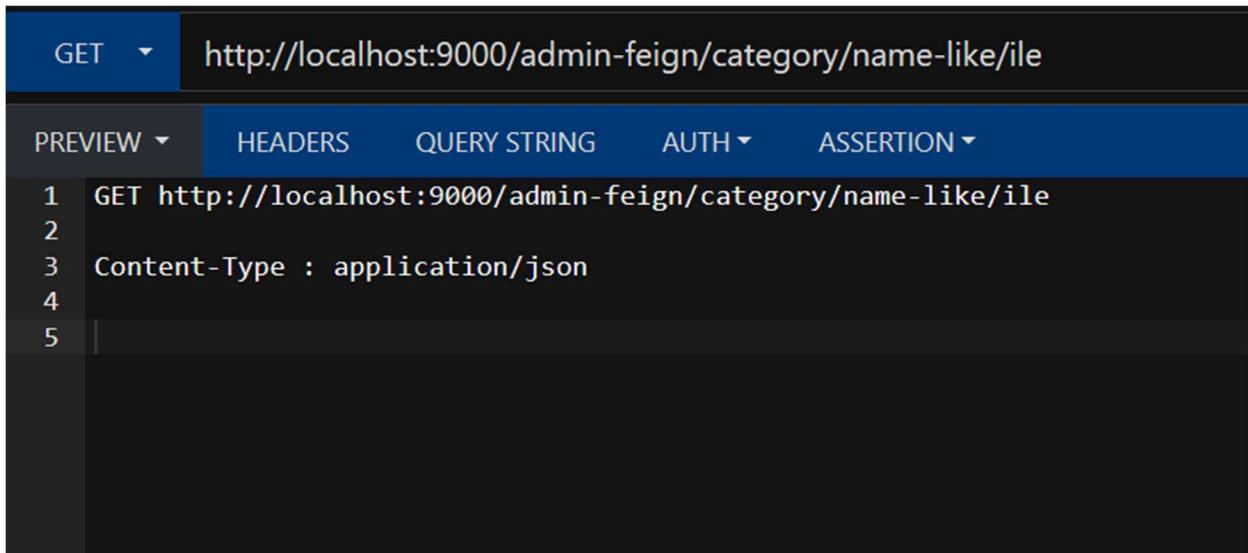
BODY HEADERS

```
1 [
2     {
3         "categoryId": 2,
4         "categoryName": "Life",
5         "policyIds": [
6             5
7         ]
8     }
9 ]
```

200 SUCCESS

Get Category By Name Phrase Match

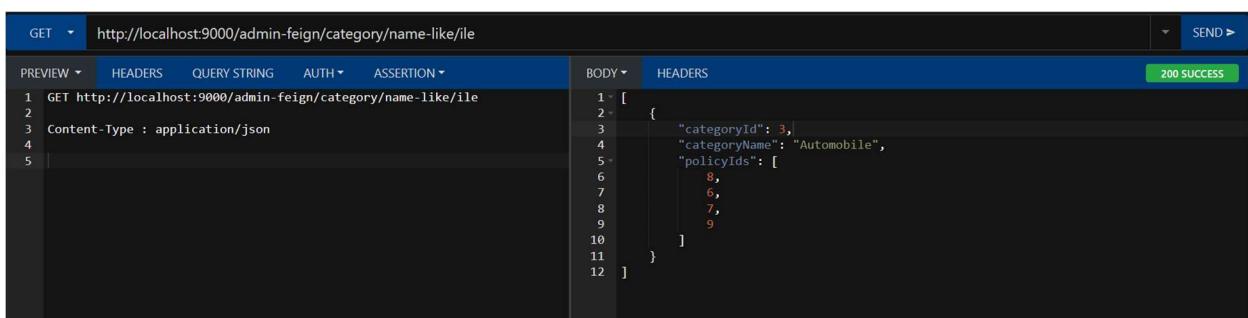
<http://localhost:9000/admin-feign/category/name-like/ile>



GET http://localhost:9000/admin-feign/category/name-like/ile

PREVIEW HEADERS QUERY STRING AUTH ASSERTION

```
1 GET http://localhost:9000/admin-feign/category/name-like/ile
2
3 Content-Type : application/json
4
5
```



GET http://localhost:9000/admin-feign/category/name-like/ile

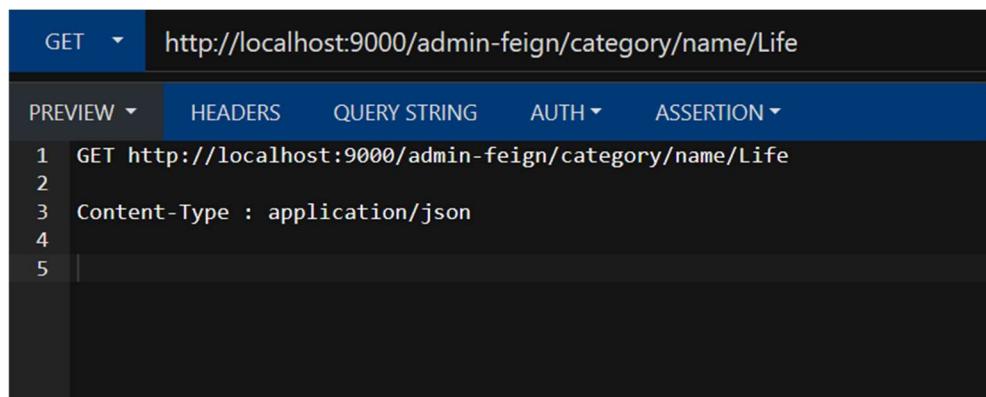
PREVIEW HEADERS QUERY STRING AUTH ASSERTION BODY HEADERS 200 SUCCESS

```
1 GET http://localhost:9000/admin-feign/category/name-like/ile
2
3 Content-Type : application/json
4
5
```

BODY	HEADERS
[{"categoryId": 3, "categoryName": "Automobile", "policyIds": [8, 6, 7, 9]}]	Content-Type: application/json Content-Length: 143

Get Category By Full Name

<http://localhost:9000/admin-feign/category/name/Life>



GET http://localhost:9000/admin-feign/category/name/Life

PREVIEW HEADERS QUERY STRING AUTH ASSERTION

```
1 GET http://localhost:9000/admin-feign/category/name/Life
2
3 Content-Type : application/json
4
5
```

GET http://localhost:9000/admin-feign/category/name/Life

Content-Type : application/json

```
1 { "categoryId": 2, "categoryName": "Life", "policyIds": [ 5 ] }
```

200 SUCCESS

User viewing available categories

All Categories

<http://localhost:9000/user-feign/category>

GET http://localhost:9000/user-feign/category

Content-Type : application/json

```
1 [ 2 { 3 "categoryId": 2, 4 "categoryName": "Life", 5 "policyIds": [ 6 5 ] 7 }, 8 { 9 "categoryId": 3, 10 "categoryName": "Automobile", 11 "policyIds": [ 12 8, 13 6, 14 7, 15 9 ] 16 } 17 ], 18 ] 19 ]
```

200 SUCCESS

Category By ID

<http://localhost:9000/user-feign/category/3>

GET http://localhost:9000/user-feign/category/3

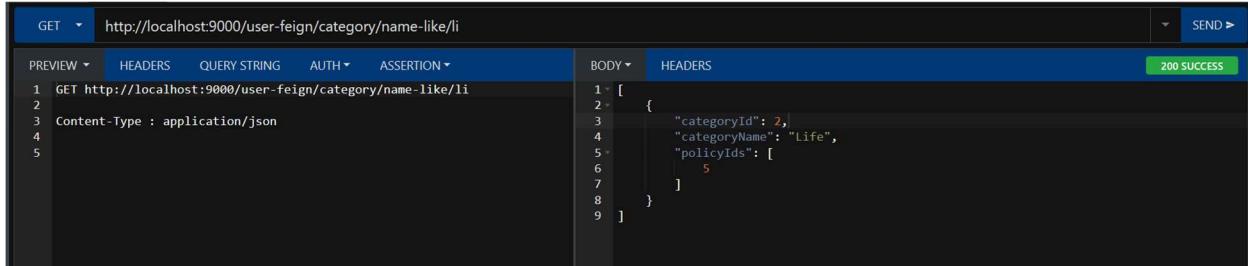
Content-Type : application/json

```
1 { 2 "categoryId": 3, 3 "categoryName": "Automobile", 4 "policyIds": [ 5 8, 6 6, 7 7, 8 9 ] 9 } 10 }
```

200 SUCCESS

Category By Name Phrase Match

<http://localhost:9000/user-feign/category/name-like/li>

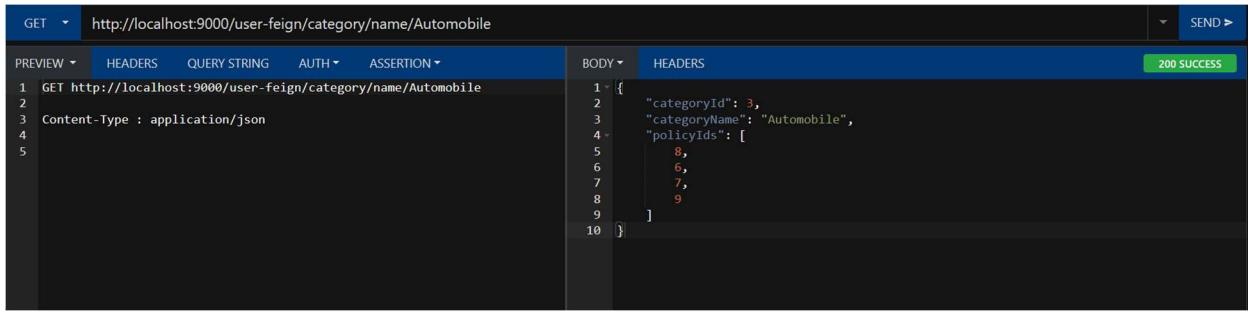


```
GET http://localhost:9000/user-feign/category/name-like/li
Content-Type : application/json
```

```
1 [ 2 { 3 "categoryId": 2, 4 "categoryName": "Life", 5 "policyIds": [ 6 5 7 ] 8 } 9 ]
```

Category By Full Name

<http://localhost:9000/user-feign/category/name/Automobile>



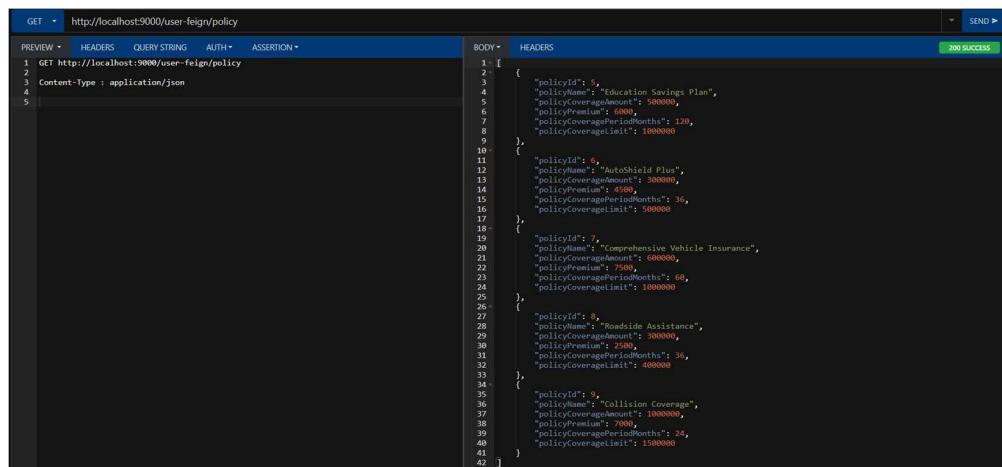
```
GET http://localhost:9000/user-feign/category/name/Automobile
Content-Type : application/json
```

```
1 [ 2 { 3 "categoryId": 3, 4 "categoryName": "Automobile", 5 "policyIds": [ 6 8, 7, 9 8 ] 9 } 10 ]
```

User viewing available policies

All Policies

<http://localhost:9000/user-feign/policy>

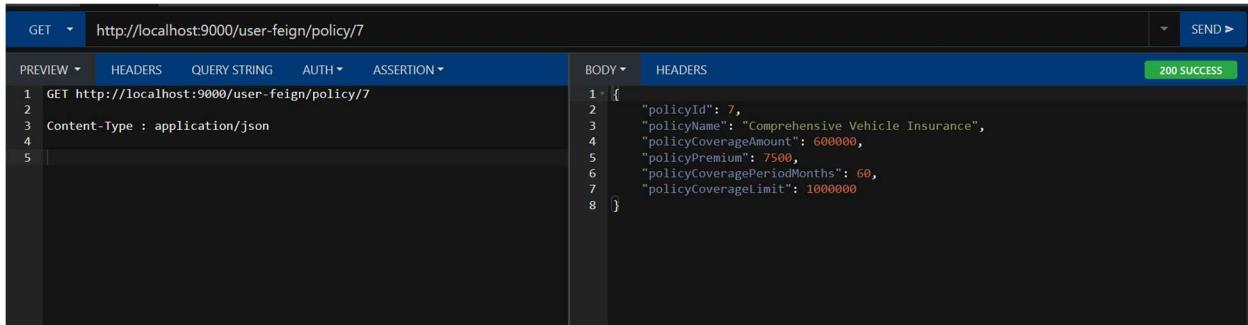


```
GET http://localhost:9000/user-feign/policy
Content-Type : application/json
```

```
1 [ 2 { 3 "policyId": 1, 4 "policyName": "Education Savings Plan", 5 "policyCoverageAmount": 500000, 6 "policyPremium": 6000, 7 "policyCoveragePeriodMonths": 120, 8 "policyCoverageLimit": 1000000 9 }, 10 { 11 "policyId": 2, 12 "policyName": "AutoShield Plus", 13 "policyCoverageAmount": 300000, 14 "policyPremium": 4500, 15 "policyCoveragePeriodMonths": 60, 16 "policyCoverageLimit": 500000 17 }, 18 { 19 "policyId": 3, 20 "policyName": "Comprehensive Vehicle Insurance", 21 "policyCoverageAmount": 600000, 22 "policyPremium": 7500, 23 "policyCoveragePeriodMonths": 60, 24 "policyCoverageLimit": 1000000 25 }, 26 { 27 "policyId": 4, 28 "policyName": "Roadside Assistance", 29 "policyCoverageAmount": 300000, 30 "policyPremium": 2500, 31 "policyCoveragePeriodMonths": 36, 32 "policyCoverageLimit": 400000 33 }, 34 { 35 "policyId": 5, 36 "policyName": "Collision Coverage", 37 "policyCoverageAmount": 1000000, 38 "policyPremium": 7000, 39 "policyCoveragePeriodMonths": 24, 40 "policyCoverageLimit": 1500000 41 } 42 ]
```

Policy By ID

<http://localhost:9000/user-feign/policy/7>

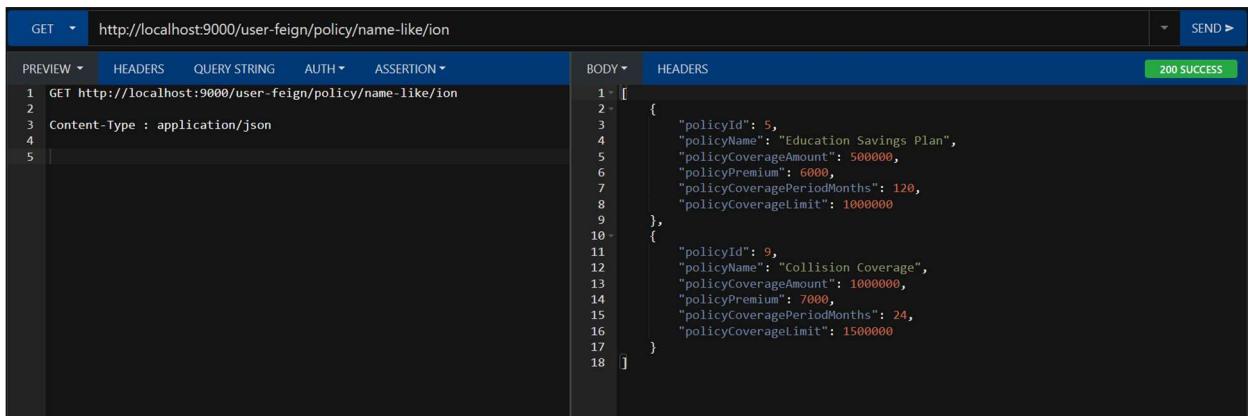


```
GET http://localhost:9000/user-feign/policy/7
Content-Type : application/json
```

PREVIEW	HEADERS	QUERY STRING	AUTH	ASSERTION	BODY	HEADERS	200 SUCCESS
1 GET http://localhost:9000/user-feign/policy/7					1 { 2 "policyId": 7, 3 "policyName": "Comprehensive Vehicle Insurance", 4 "policyCoverageAmount": 600000, 5 "policyPremium": 7500, 6 "policyCoveragePeriodMonths": 60, 7 "policyCoverageLimit": 1000000 8 }		

Policies with name phrase match

<http://localhost:9000/user-feign/policy/name-like/ion>

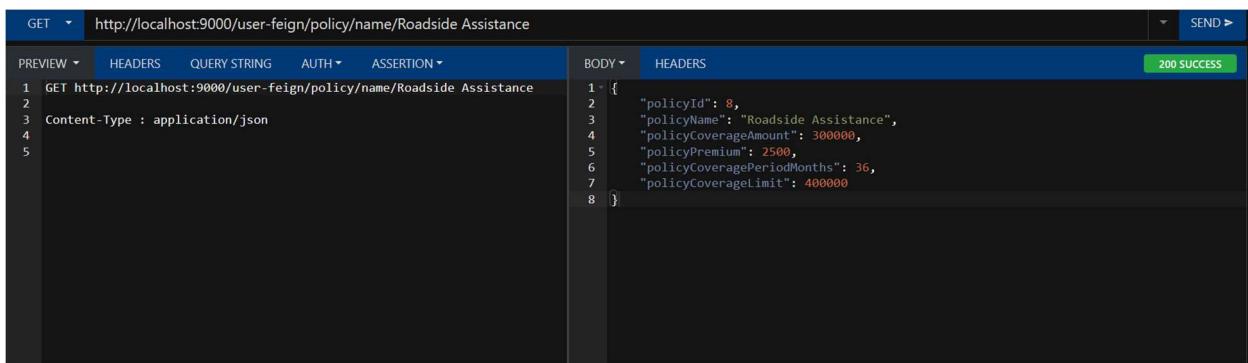


```
GET http://localhost:9000/user-feign/policy/name-like/ion
Content-Type : application/json
```

PREVIEW	HEADERS	QUERY STRING	AUTH	ASSERTION	BODY	HEADERS	200 SUCCESS
1 GET http://localhost:9000/user-feign/policy/name-like/ion					1 [2 { 3 "policyId": 5, 4 "policyName": "Education Savings Plan", 5 "policyCoverageAmount": 500000, 6 "policyPremium": 6000, 7 "policyCoveragePeriodMonths": 120, 8 "policyCoverageLimit": 1000000 9 }, 10 { 11 "policyId": 9, 12 "policyName": "Collision Coverage", 13 "policyCoverageAmount": 1000000, 14 "policyPremium": 7000, 15 "policyCoveragePeriodMonths": 24, 16 "policyCoverageLimit": 1500000 17 } 18]		

Policy By Full Name

<http://localhost:9000/user-feign/policy/name/Roadside Assistance>

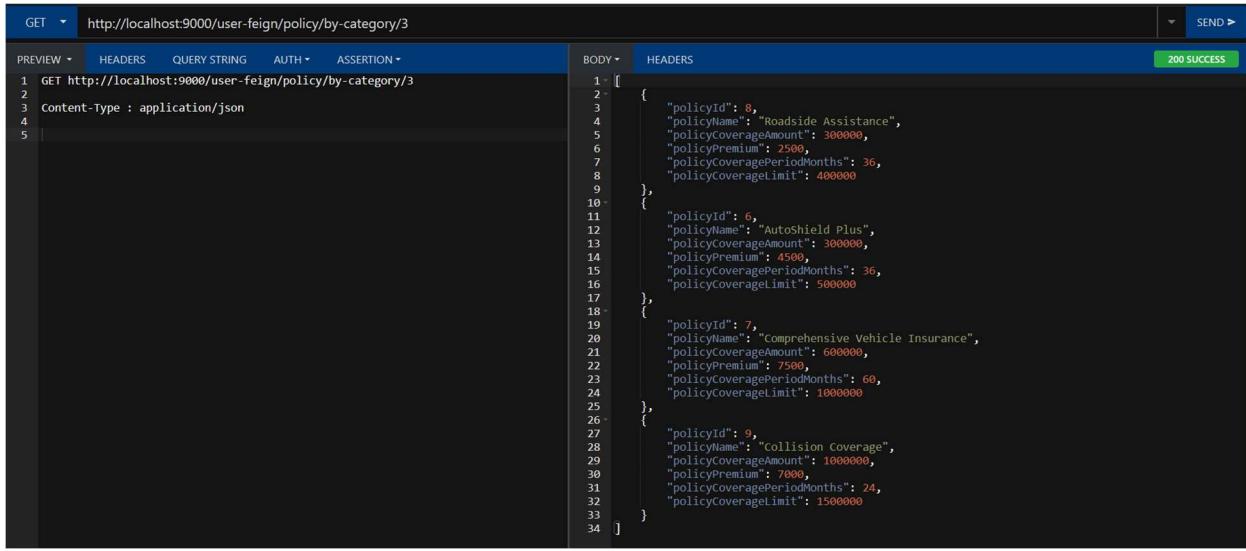


```
GET http://localhost:9000/user-feign/policy/name/Roadside Assistance
Content-Type : application/json
```

PREVIEW	HEADERS	QUERY STRING	AUTH	ASSERTION	BODY	HEADERS	200 SUCCESS
1 GET http://localhost:9000/user-feign/policy/name/Roadside Assistance					1 { 2 "policyId": 8, 3 "policyName": "Roadside Assistance", 4 "policyCoverageAmount": 300000, 5 "policyPremium": 2500, 6 "policyCoveragePeriodMonths": 36, 7 "policyCoverageLimit": 400000 8 }		

Policies By Category

<http://localhost:9000/user-feign/policy/by-category/3>

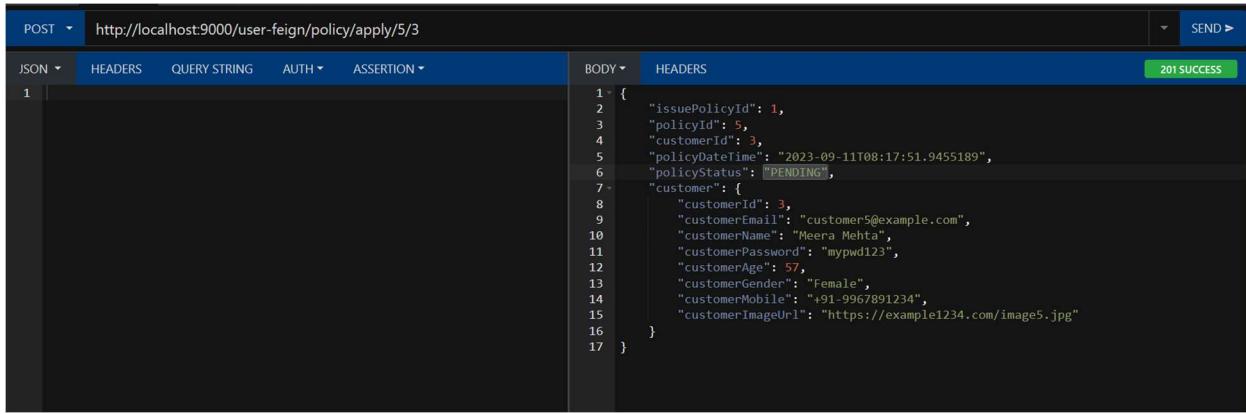


```
1 GET http://localhost:9000/user-feign/policy/by-category/3
2
3 Content-Type : application/json
4
5
6
7
8
9
10 [
11   {
12     "policyId": 8,
13     "policyName": "Roadside Assistance",
14     "policyCoverageAmount": 300000,
15     "policyPremium": 2500,
16     "policyCoveragePeriodMonths": 36,
17     "policyCoverageLimit": 400000
18   },
19   {
20     "policyId": 6,
21     "policyName": "AutoShield Plus",
22     "policyCoverageAmount": 300000,
23     "policyPremium": 4500,
24     "policyCoveragePeriodMonths": 36,
25     "policyCoverageLimit": 500000
26   },
27   {
28     "policyId": 7,
29     "policyName": "Comprehensive Vehicle Insurance",
30     "policyCoverageAmount": 600000,
31     "policyPremium": 7500,
32     "policyCoveragePeriodMonths": 60,
33     "policyCoverageLimit": 1000000
34   },
35   {
36     "policyId": 9,
37     "policyName": "Collision Coverage",
38     "policyCoverageAmount": 1000000,
39     "policyPremium": 7000,
40     "policyCoveragePeriodMonths": 24,
41     "policyCoverageLimit": 1500000
42   }
43 ]
```

Customer Applying for a policy

<http://localhost:9000/user-feign/policy/apply/5/3>

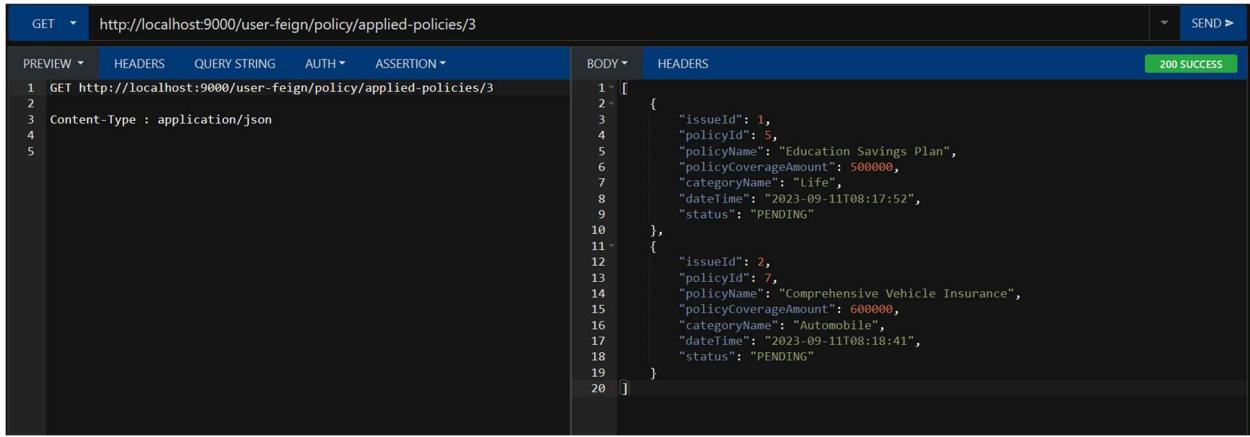
(<http://localhost:9000/user-feign/policy/apply/{policyId}/{userId}>)



```
1 POST http://localhost:9000/user-feign/policy/apply/5/3
2
3 JSON
4
5
6
7
8
9
10
11
12
13
14
15
16
17 { "issuePolicyId": 1, "policyId": 5, "customerId": 3, "policyDateTime": "2023-09-11T08:17:51.9455189", "policyStatus": "PENDING", "customer": { "customerId": 3, "customerEmail": "customer5@example.com", "customerName": "Meera Mehta", "customerPassword": "mypwd123", "customerAge": 57, "customerGender": "Female", "customerMobile": "+91-9967891234", "customerImageUrl": "https://example1234.com/image5.jpg" } }
```

Customer viewing the policy history of all applied policies' status

<http://localhost:9000/user-feign/policy/applied-policies/3>



PREVIEW ▾

HEADERS

QUERY STRING

AUTH ▾

ASSERTION ▾

1 GET http://localhost:9000/user-feign/policy/applied-policies/3

2 Content-Type : application/json

3

4

5

BODY ▾

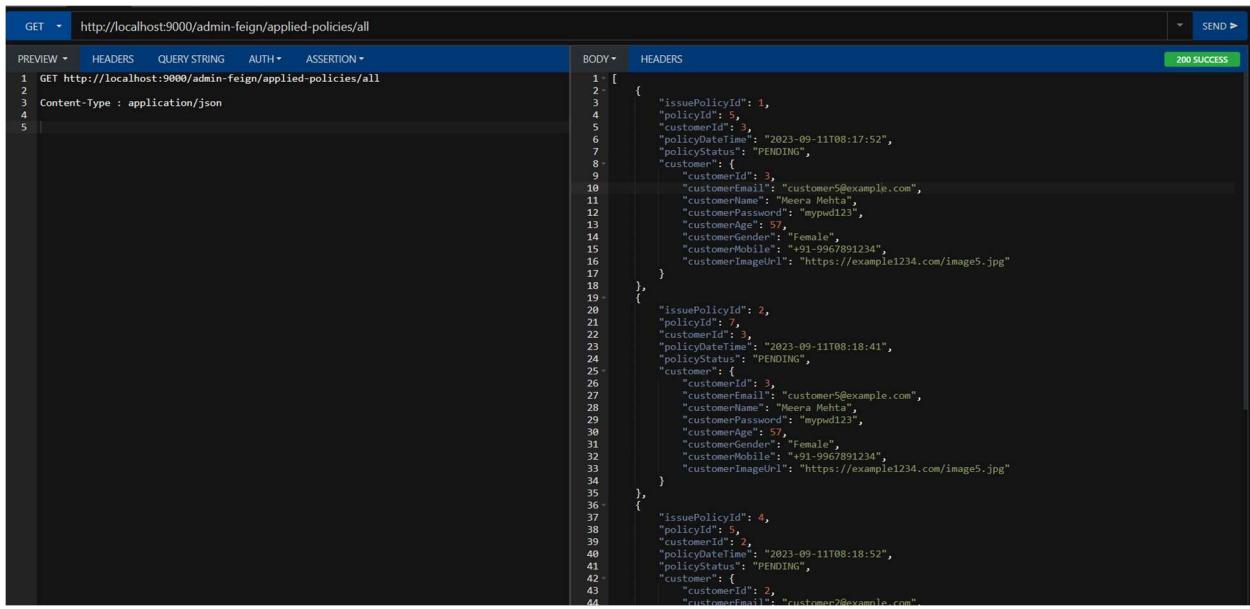
HEADERS

200 SUCCESS

```
1: [  
2:   {  
3:     "issueId": 1,  
4:     "policyId": 5,  
5:     "policyName": "Education Savings Plan",  
6:     "policyCoverageAmount": 500000,  
7:     "categoryName": "Life",  
8:     "dateTime": "2023-09-11T08:17:52",  
9:     "status": "PENDING"  
10:    },  
11:    {  
12:      "issueId": 2,  
13:      "policyId": 7,  
14:      "policyName": "Comprehensive Vehicle Insurance",  
15:      "policyCoverageAmount": 600000,  
16:      "categoryName": "Automobile",  
17:      "dateTime": "2023-09-11T08:18:41",  
18:      "status": "PENDING"  
19:    }  
20: ]
```

Admin Viewing Applied policies

<http://localhost:9000/admin-feign/applied-policies/all>



PREVIEW ▾

HEADERS

QUERY STRING

AUTH ▾

ASSERTION ▾

1 GET http://localhost:9000/admin-feign/applied-policies/all

2 Content-Type : application/json

3

4

5

BODY ▾

HEADERS

200 SUCCESS

```
1: [  
2:   {  
3:     "issuePolicyId": 1,  
4:     "policyId": 5,  
5:     "customerId": 3,  
6:     "policyDateTime": "2023-09-11T08:17:52",  
7:     "policyStatus": "PENDING",  
8:     "customer": {  
9:       "customerId": 3,  
10:      "customerEmail": "customer5@example.com",  
11:      "customerName": "Neera Mehta",  
12:      "customerPassword": "mypwd123",  
13:      "customerAge": 57,  
14:      "customerGender": "Female",  
15:      "customerMobile": "+91-9967891234",  
16:      "customerImageUrl": "https://example1234.com/image5.jpg"  
17:    },  
18:    {  
19:      "issuePolicyId": 2,  
20:      "policyId": 7,  
21:      "customerId": 3,  
22:      "policyDateTime": "2023-09-11T08:18:41",  
23:      "policyStatus": "PENDING",  
24:      "customer": {  
25:        "customerId": 3,  
26:        "customerEmail": "customer5@example.com",  
27:        "customerName": "Neera Mehta",  
28:        "customerPassword": "mypwd123",  
29:        "customerAge": 57,  
30:        "customerGender": "Female",  
31:        "customerMobile": "+91-9967891234",  
32:        "customerImageUrl": "https://example1234.com/image5.jpg"  
33:      },  
34:      {  
35:        "issuePolicyId": 4,  
36:        "policyId": 5,  
37:        "customerId": 2,  
38:        "policyDateTime": "2023-09-11T08:18:52",  
39:        "policyStatus": "PENDING",  
40:        "customer": {  
41:          "customerId": 2,  
42:          "customerEmail": "customer2@example.com",  
43:          "customerName": "Rahul Mehta",  
44:          "customerPassword": "mypwd123",  
45:          "customerAge": 34,  
46:          "customerGender": "Male",  
47:          "customerMobile": "+91-9967891235",  
48:          "customerImageUrl": "https://example1234.com/image5.jpg"  
49:        }  
50:      }  
51:    }  
52:  ]
```

Admin approving/ disapproving the applied policies

<http://localhost:9000/admin-feign/applied-policies/set-status>

PUT http://localhost:9000/admin-feign/applied-policies/set-status

JSON HEADERS QUERY STRING AUTH ASSERTION

```
1 {  
2   "issuePolicyId": 1,  
3   "policyId": 5,  
4   "customerId": 3,  
5   "policyDateTime": "2023-09-11T08:17:52",  
6   "policyStatus": ["APPROVED"],  
7   "customer": {  
8     "customerId": 3,  
9     "customerEmail": "customer5@example.com",  
10    "customerName": "Meera Mehta",  
11    "customerPassword": "mypwd123",  
12    "customerAge": 57,  
13    "customerGender": "Female",  
14    "customerMobile": "+91-9967891234",  
15    "customerImageUrl": "https://example1234.com/image5.jpg"  
16  }  
17 }
```

PUT http://localhost:9000/admin-feign/applied-policies/set-status

200 SUCCESS

JSON HEADERS QUERY STRING AUTH ASSERTION BODY HEADERS

```
1 {  
2   "issuePolicyId": 1,  
3   "policyId": 5,  
4   "customerId": 3,  
5   "policyDateTime": "2023-09-11T08:17:52",  
6   "policyStatus": ["APPROVED"],  
7   "customer": {  
8     "customerId": 3,  
9     "customerEmail": "customer5@example.com",  
10    "customerName": "Meera Mehta",  
11    "customerPassword": "mypwd123",  
12    "customerAge": 57,  
13    "customerGender": "Female",  
14    "customerMobile": "+91-9967891234",  
15    "customerImageUrl": "https://example1234.com/image5.jpg"  
16  }  
17 }
```

```
1 {  
2   "issuePolicyId": 1,  
3   "policyId": 5,  
4   "customerId": 3,  
5   "policyDateTime": "2023-09-11T08:17:52",  
6   "policyStatus": "APPROVED",  
7   "customer": {  
8     "customerId": 3,  
9     "customerEmail": "customer5@example.com",  
10    "customerName": "Meera Mehta",  
11    "customerPassword": "mypwd123",  
12    "customerAge": 57,  
13    "customerGender": "Female",  
14    "customerMobile": "+91-9967891234",  
15    "customerImageUrl": "https://example1234.com/image5.jpg"  
16  }  
17 }
```

PUT http://localhost:9000/admin-feign/applied-policies/set-status

200 SUCCESS

JSON HEADERS QUERY STRING AUTH ASSERTION BODY HEADERS

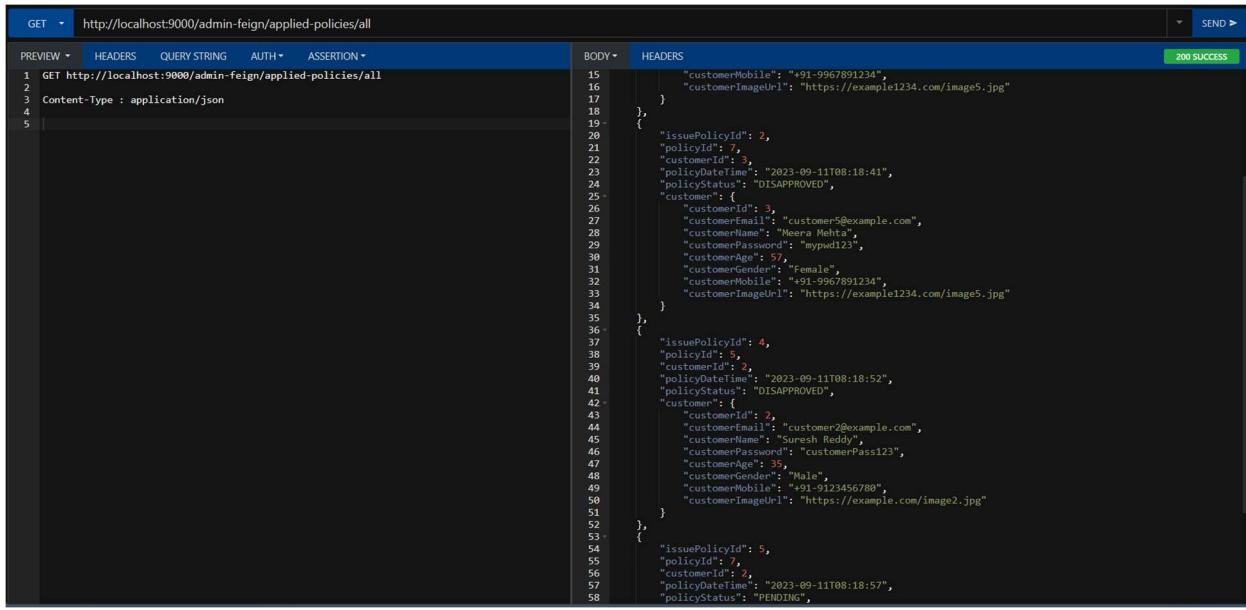
```
1 {  
2   "issuePolicyId": 2,  
3   "policyId": 7,  
4   "customerId": 3,  
5   "policyDateTime": "2023-09-11T08:18:41",  
6   "policyStatus": "DISAPPROVED",  
7   "customer": {  
8     "customerId": 3,  
9     "customerEmail": "customer5@example.com",  
10    "customerName": "Meera Mehta",  
11    "customerPassword": "mypwd123",  
12    "customerAge": 57,  
13    "customerGender": "Female",  
14    "customerMobile": "+91-9967891234",  
15    "customerImageUrl": "https://example1234.com/image5.jpg"  
16  }  
17 }
```

```
1 {  
2   "issuePolicyId": 2,  
3   "policyId": 7,  
4   "customerId": 3,  
5   "policyDateTime": "2023-09-11T08:18:41",  
6   "policyStatus": "DISAPPROVED",  
7   "customer": {  
8     "customerId": 3,  
9     "customerEmail": "customer5@example.com",  
10    "customerName": "Meera Mehta",  
11    "customerPassword": "mypwd123",  
12    "customerAge": 57,  
13    "customerGender": "Female",  
14    "customerMobile": "+91-9967891234",  
15    "customerImageUrl": "https://example1234.com/image5.jpg"  
16  }  
17 }
```

Admin viewing all/ pending/ approved/ disapproved applied policies

1. All

<http://localhost:9000/admin-feign/applied-policies/all>



PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

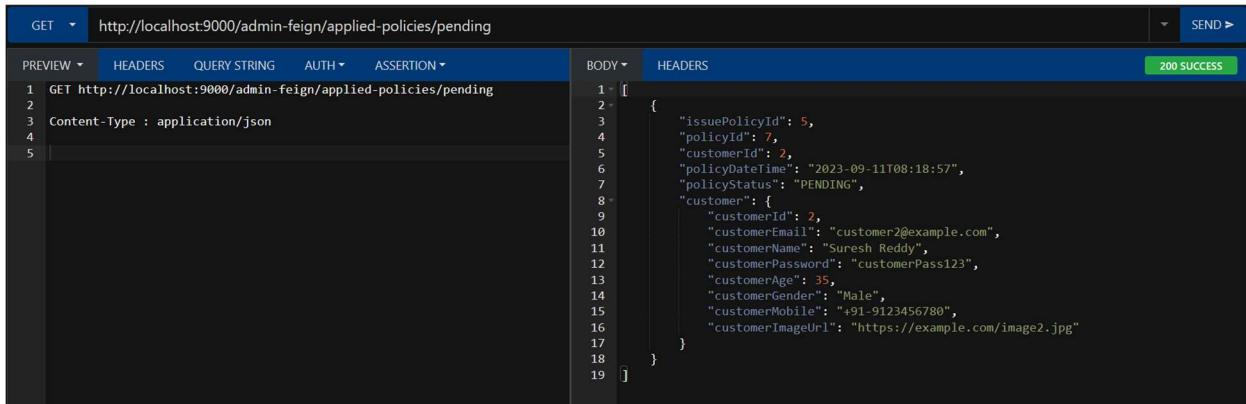
1 GET http://localhost:9000/admin-feign/applied-policies/all
2 Content-Type : application/json
3

BODY ▾ HEADERS

```
15     "customerMobile": "+91-9967891234",
16     "customerImageUrl": "https://example1234.com/image5.jpg"
17   },
18   {
19     "issuePolicyId": 2,
20     "policyId": 7,
21     "customerId": 3,
22     "policyDateTime": "2023-09-11T08:18:41",
23     "policyStatus": "DISAPPROVED",
24     "customer": {
25       "customerId": 3,
26       "customerEmail": "customer5@example.com",
27       "customerName": "Neera Mehta",
28       "customerPassword": "mypwd123",
29       "customerAge": 57,
30       "customerGender": "Female",
31       "customerMobile": "+91-9967891234",
32       "customerImageUrl": "https://example1234.com/image5.jpg"
33     }
34   },
35   {
36     "issuePolicyId": 4,
37     "policyId": 5,
38     "customerId": 2,
39     "policyDateTime": "2023-09-11T08:18:52",
40     "policyStatus": "DISAPPROVED",
41     "customer": {
42       "customerId": 2,
43       "customerEmail": "customer2@example.com",
44       "customerName": "Suresh Reddy",
45       "customerPassword": "customerPass123",
46       "customerAge": 35,
47       "customerGender": "Male",
48       "customerMobile": "+91-9123456780",
49       "customerImageUrl": "https://example.com/image2.jpg"
50     }
51   },
52   {
53     "issuePolicyId": 5,
54     "policyId": 7,
55     "customerId": 2,
56     "policyDateTime": "2023-09-11T08:18:57",
57     "policyStatus": "PENDING",
58   }
```

2. Pending

<http://localhost:9000/admin-feign/applied-policies/pending>



PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

1 GET http://localhost:9000/admin-feign/applied-policies/pending
2 Content-Type : application/json
3

BODY ▾ HEADERS

```
1 [ 2   {
3     "issuePolicyId": 5,
4     "policyId": 7,
5     "customerId": 2,
6     "policyDateTime": "2023-09-11T08:18:57",
7     "policyStatus": "PENDING",
8     "customer": {
9       "customerId": 2,
10      "customerEmail": "customer2@example.com",
11      "customerName": "Suresh Reddy",
12      "customerPassword": "customerPass123",
13      "customerAge": 35,
14      "customerGender": "Male",
15      "customerMobile": "+91-9123456780",
16      "customerImageUrl": "https://example.com/image2.jpg"
17    }
18  ]
```

3. Approved

<http://localhost:9000/admin-feign/applied-policies/approved>

GET http://localhost:9000/admin-feign/applied-policies/approved

PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

```
1 GET http://localhost:9000/admin-feign/applied-policies/approved
2
3 Content-Type : application/json
4
5
```

BODY ▾ HEADERS

```
1 - [
2 -   {
3 -     "issuePolicyId": 1,
4 -     "policyId": 5,
5 -     "customerId": 3,
6 -     "policyDateTime": "2023-09-11T08:17:52",
7 -     "policyStatus": "APPROVED",
8 -     "customer": {
9 -       "customerId": 3,
10 -      "customerEmail": "customer5@example.com",
11 -      "customerName": "Meera Mehta",
12 -      "customerPassword": "mypwd123",
13 -      "customerAge": 57,
14 -      "customerGender": "Female",
15 -      "customerMobile": "+91-9967891234",
16 -      "customerImageUrl": "https://example1234.com/image5.jpg"
17 -    }
18 -  }
19 - ]
```

200 SUCCESS

4. Disapproved

<http://localhost:9000/admin-feign/applied-policies/disapproved>

GET http://localhost:9000/admin-feign/applied-policies/disapproved

PREVIEW ▾ HEADERS QUERY STRING AUTH ▾ ASSERTION ▾

```
1 GET http://localhost:9000/admin-feign/applied-policies/disapproved
2
3 Content-Type : application/json
4
5
```

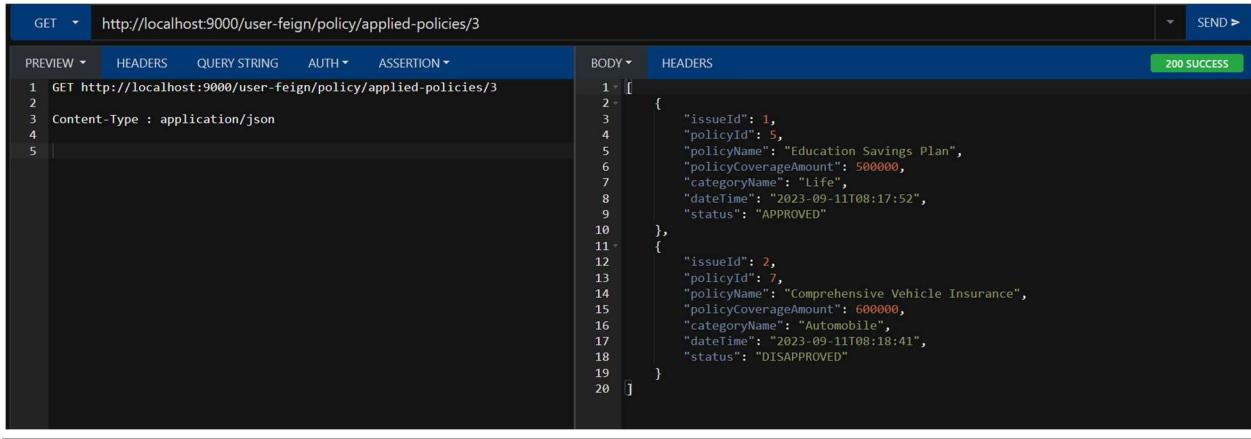
BODY ▾ HEADERS

```
1 - [
2 -   {
3 -     "issuePolicyId": 2,
4 -     "policyId": 7,
5 -     "customerId": 3,
6 -     "policyDateTime": "2023-09-11T08:18:41",
7 -     "policyStatus": "DISAPPROVED",
8 -     "customer": {
9 -       "customerId": 3,
10 -      "customerEmail": "customer5@example.com",
11 -      "customerName": "Meera Mehta",
12 -      "customerPassword": "mypwd123",
13 -      "customerAge": 57,
14 -      "customerGender": "Female",
15 -      "customerMobile": "+91-9967891234",
16 -      "customerImageUrl": "https://example1234.com/image5.jpg"
17 -    },
18 -   },
19 -   {
20 -     "issuePolicyId": 4,
21 -     "policyId": 5,
22 -     "customerId": 2,
23 -     "policyDateTime": "2023-09-11T08:18:52",
24 -     "policyStatus": "DISAPPROVED",
25 -     "customer": {
26 -       "customerId": 2,
27 -       "customerEmail": "customer2@example.com",
28 -       "customerName": "Suresh Reddy",
29 -       "customerPassword": "CustomerPass123",
30 -       "customerAge": 35,
31 -       "customerGender": "Male",
32 -       "customerMobile": "+91-9123456780",
33 -       "customerImageUrl": "https://example.com/image2.jpg"
34 -     }
35 -   }
36 - ]
```

200 SUCCESS

Customer again viewing their applied policy history

<http://localhost:9000/user-feign/policy/applied-policies/3>



PREVIEW ▾

HEADERS

QUERY STRING

AUTH ▾

ASSERTION ▾

BODY ▾

HEADERS

200 SUCCESS

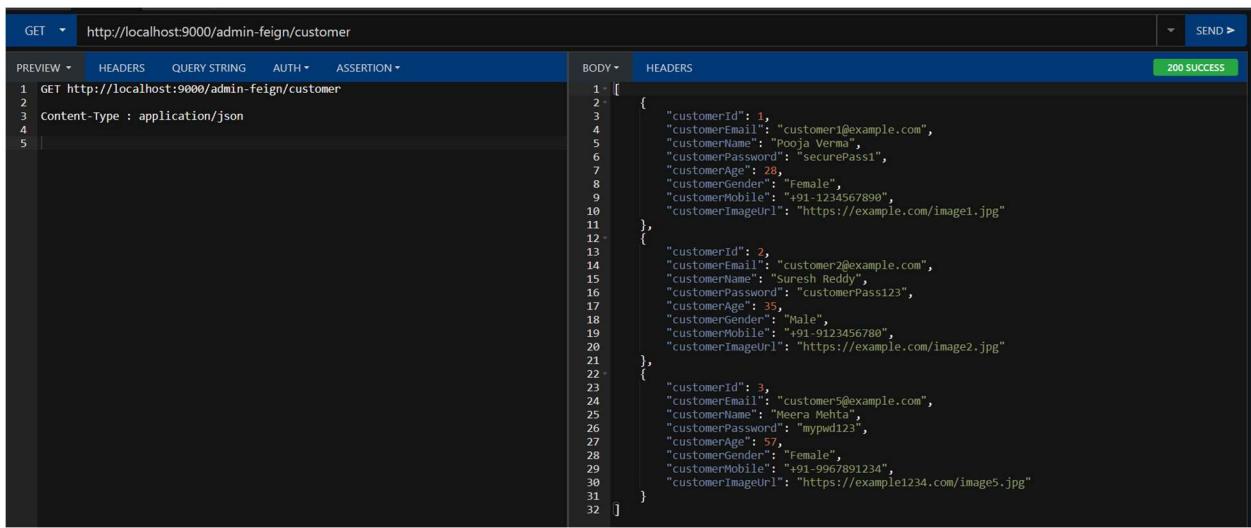
```
1 GET http://localhost:9000/user-feign/policy/applied-policies/3
2 Content-Type : application/json
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

```
[{"issueId": 1, "policyId": 3, "policyName": "Education Savings Plan", "policyCoverageAmount": 500000, "categoryName": "Life", "dateTime": "2023-09-11T08:17:52", "status": "APPROVED"}, {"issueId": 2, "policyId": 7, "policyName": "Comprehensive Vehicle Insurance", "policyCoverageAmount": 600000, "categoryName": "Automobile", "dateTime": "2023-09-11T08:18:41", "status": "DISAPPROVED"}]
```

Admin's Operations with customer details

View all customers with basic information

<http://localhost:9000/admin-feign/customer>



PREVIEW ▾

HEADERS

QUERY STRING

AUTH ▾

ASSERTION ▾

BODY ▾

HEADERS

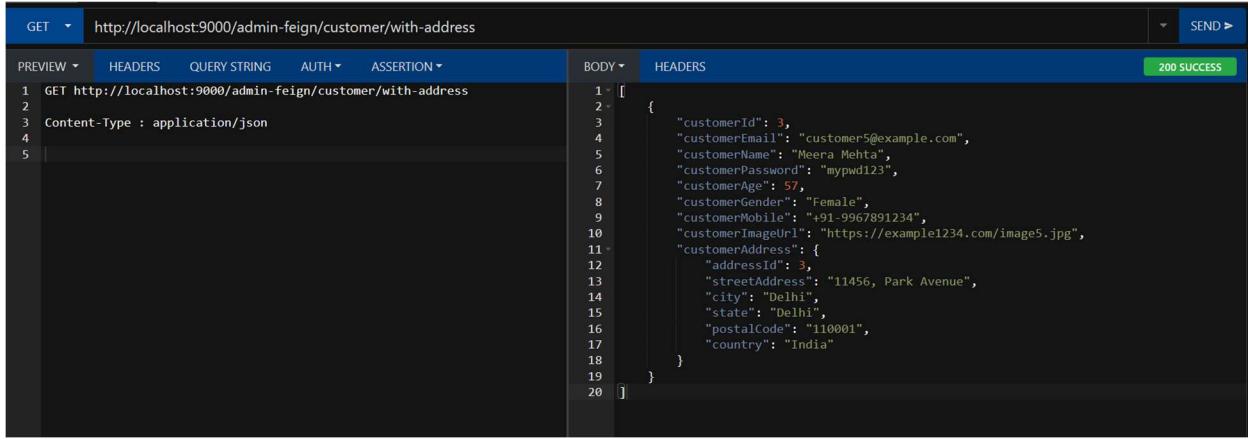
200 SUCCESS

```
1 GET http://localhost:9000/admin-feign/customer
2 Content-Type : application/json
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
```

```
[{"customerId": 1, "customerEmail": "customer1@example.com", "customerName": "Pooja Verma", "customerPassword": "securePass1", "customerAge": 28, "customerGender": "Female", "customerMobile": "+91-1234567890", "customerImageUrl": "https://example.com/image1.jpg"}, {"customerId": 2, "customerEmail": "customer2@example.com", "customerName": "Suresh Reddy", "customerPassword": "customerPass123", "customerAge": 35, "customerGender": "Male", "customerMobile": "+91-9123456780", "customerImageUrl": "https://example.com/image2.jpg"}, {"customerId": 3, "customerEmail": "customer3@example.com", "customerName": "Meera Mehta", "customerPassword": "mypwd123", "customerAge": 57, "customerGender": "Female", "customerMobile": "+91-9967891234", "customerImageUrl": "https://example1234.com/image5.jpg"}]
```

View all customers with a registered address

<http://localhost:9000/admin-feign/customer/with-address>



```
GET http://localhost:9000/admin-feign/customer/with-address
Content-Type : application/json
```

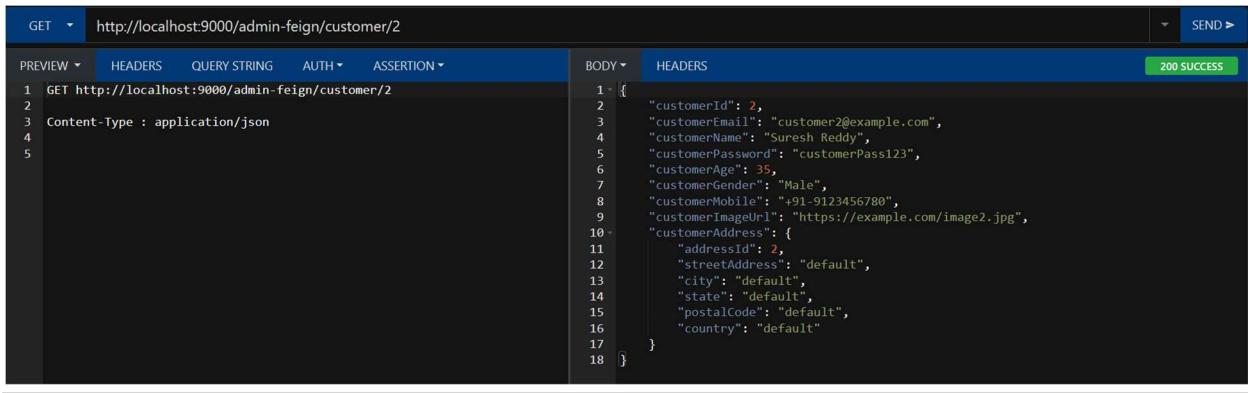
```
1 GET http://localhost:9000/admin-feign/customer/with-address
2
3 Content-Type : application/json
4
5
```

```
1 [
2   {
3     "customerId": 3,
4     "customerEmail": "customer5@example.com",
5     "customerName": "Meera Mehta",
6     "customerPassword": "mypwd123",
7     "customerAge": 57,
8     "customerGender": "Female",
9     "customerMobile": "+91-9967891234",
10    "customerImageUrl": "https://example1234.com/image5.jpg",
11    "customerAddress": {
12      "addressId": 3,
13      "streetAddress": "11456, Park Avenue",
14      "city": "Delhi",
15      "state": "Delhi",
16      "postalCode": "110001",
17      "country": "India"
18    }
19  ]
20 ]
```

Get customer by Customer ID

<http://localhost:9000/admin-feign/customer/2>

(Default values are shown till the customer fills their address)

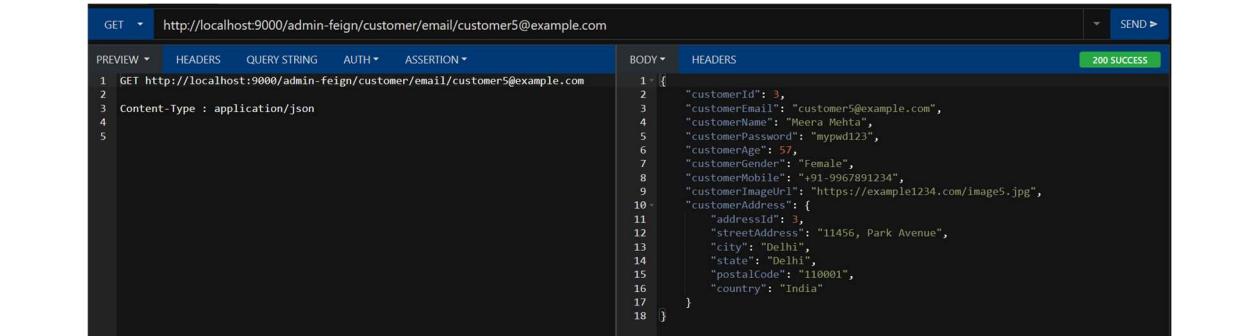


```
GET http://localhost:9000/admin-feign/customer/2
Content-Type : application/json
```

```
1 GET http://localhost:9000/admin-feign/customer/2
2
3 Content-Type : application/json
4
5
```

```
1 [
2   {
3     "customerId": 2,
4     "customerEmail": "customer2@example.com",
5     "customerName": "Suresh Reddy",
6     "customerPassword": "customerPass123",
7     "customerAge": 35,
8     "customerGender": "Male",
9     "customerMobile": "+91-9123456780",
10    "customerImageUrl": "https://example.com/image2.jpg",
11    "customerAddress": {
12      "addressId": 2,
13      "streetAddress": "default",
14      "city": "default",
15      "state": "default",
16      "postalCode": "default",
17      "country": "default"
18    }
19  ]
20 ]
```

Get Customer By Email



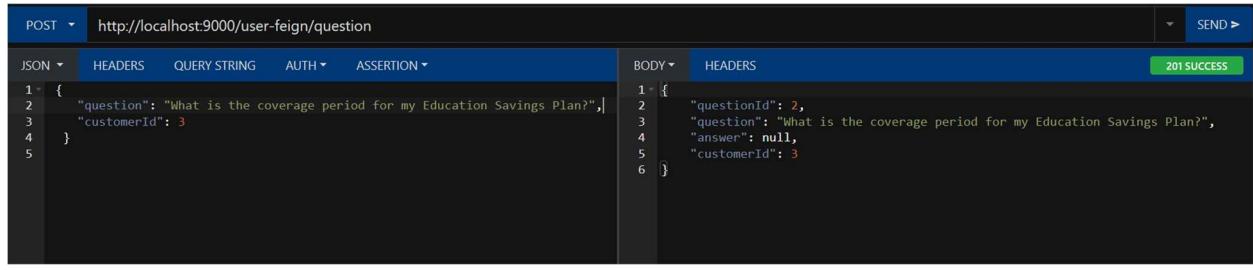
```
GET http://localhost:9000/admin-feign/customer/email/customer5@example.com
Content-Type : application/json
```

```
1 GET http://localhost:9000/admin-feign/customer/email/customer5@example.com
2
3 Content-Type : application/json
4
5
```

```
1 [
2   {
3     "customerId": 3,
4     "customerEmail": "customer5@example.com",
5     "customerName": "Meera Mehta",
6     "customerPassword": "mypwd123",
7     "customerAge": 57,
8     "customerGender": "Female",
9     "customerMobile": "+91-9967891234",
10    "customerImageUrl": "https://example1234.com/image5.jpg",
11    "customerAddress": {
12      "addressId": 3,
13      "streetAddress": "11456, Park Avenue",
14      "city": "Delhi",
15      "state": "Delhi",
16      "postalCode": "110001",
17      "country": "India"
18    }
19  ]
20 ]
```

Customers posting their questions

<http://localhost:9000/user-feign/question>



```
POST http://localhost:9000/user-feign/question

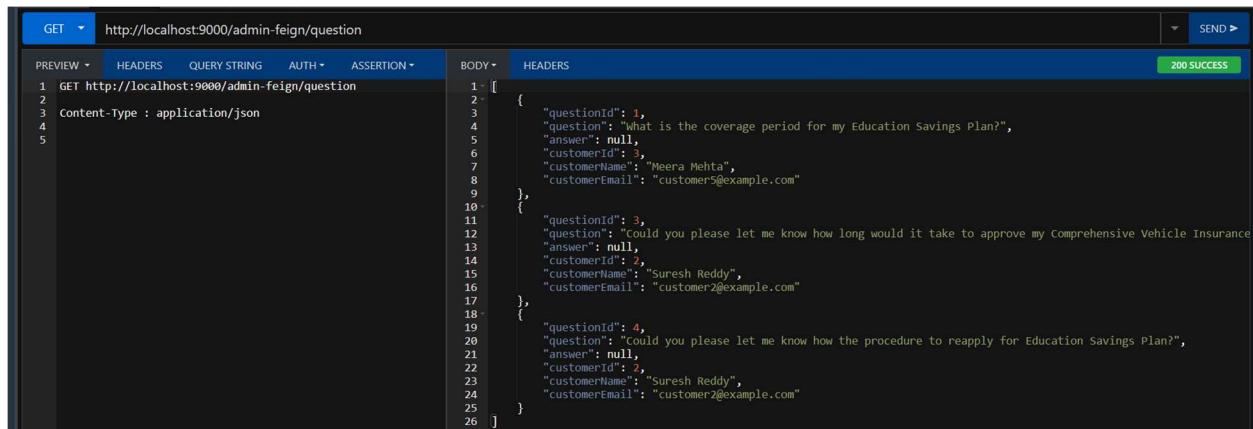
BODY
1 - {
2 -   "question": "What is the coverage period for my Education Savings Plan?", |
3 -   "customerId": 3
4 - }
5

1 - {
2 -   "questionId": 2,
3 -   "question": "What is the coverage period for my Education Savings Plan?",
4 -   "answer": null,
5 -   "customerId": 3
6 }
```

Admin viewing the questions

1. All Questions

<http://localhost:9000/admin-feign/question>

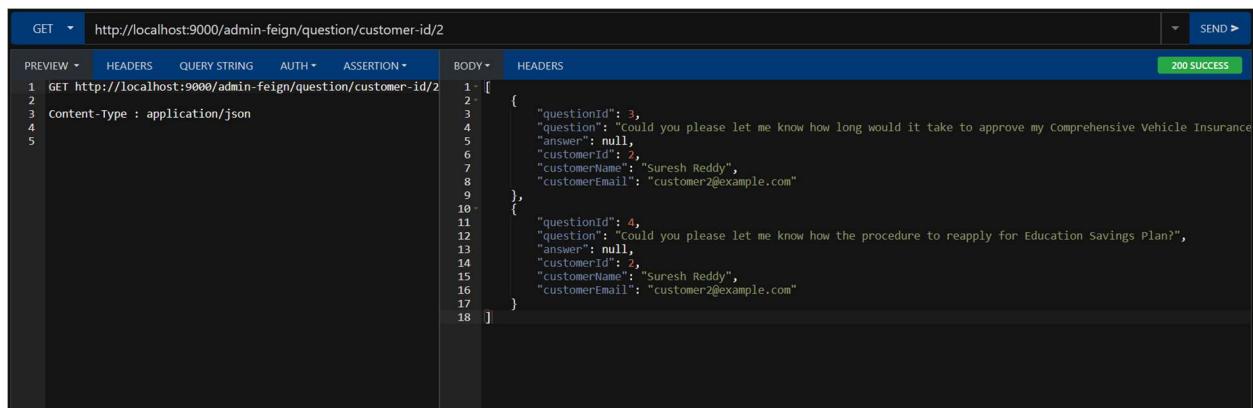


```
GET http://localhost:9000/admin-feign/question

BODY
1 - [
2 -   {
3 -     "questionId": 1,
4 -     "question": "What is the coverage period for my Education Savings Plan?",
5 -     "answer": null,
6 -     "customerId": 3,
7 -     "customerName": "Meera Mehta",
8 -     "customerEmail": "customer5@example.com"
9 -   },
10 -   {
11 -     "questionId": 3,
12 -     "question": "Could you please let me know how long would it take to approve my Comprehensive Vehicle Insurance",
13 -     "answer": null,
14 -     "customerId": 2,
15 -     "customerName": "Suresh Reddy",
16 -     "customerEmail": "customer2@example.com"
17 -   },
18 -   {
19 -     "questionId": 4,
20 -     "question": "Could you please let me know how the procedure to reapply for Education Savings Plan?",
21 -     "answer": null,
22 -     "customerId": 2,
23 -     "customerName": "Suresh Reddy",
24 -     "customerEmail": "customer2@example.com"
25 -   }
26 ]
```

2. Questions by customer ID

<http://localhost:9000/admin-feign/question/customer-id/2>



```
GET http://localhost:9000/admin-feign/question/customer-id/2

BODY
1 - [
2 -   {
3 -     "questionId": 3,
4 -     "question": "Could you please let me know how long would it take to approve my Comprehensive Vehicle Insurance",
5 -     "answer": null,
6 -     "customerId": 2,
7 -     "customerName": "Suresh Reddy",
8 -     "customerEmail": "customer2@example.com"
9 -   },
10 -   {
11 -     "questionId": 4,
12 -     "question": "Could you please let me know how the procedure to reapply for Education Savings Plan?",
13 -     "answer": null,
14 -     "customerId": 2,
15 -     "customerName": "Suresh Reddy",
16 -     "customerEmail": "customer2@example.com"
17 -   }
18 ]
```

Admin adding answers

<http://localhost:9000/admin-feign/question>

The screenshot shows a POST request to `http://localhost:9000/admin-feign/question`. The request body contains the following JSON:

```
1 - {
2     "questionId": 1,
3     "question": "What is the coverage period for my Education Savings Plan?",
4     "answer": "INR 5,00,000",
5     "customerId": 3,
6     "customerName": "Meera Mehta",
7     "customerEmail": "customer5@example.com"
8 }
```

The response status is 200 SUCCESS.

Customer again viewing the questions

<http://localhost:9000/user-feign/question/customer-id/3>

The screenshot shows a GET request to `http://localhost:9000/user-feign/question/customer-id/3`. The request body contains the following JSON:

```
1 - [
2     {
3         "questionId": 1,
4         "question": "What is the coverage period for my Education Savings Plan?",
5         "answer": "INR 5,00,000",
6         "customerId": 3
7     }
8 ]
```

The response status is 200 SUCCESS.

General user filling the contact-form

<http://localhost:9000/user-feign/contact-form>

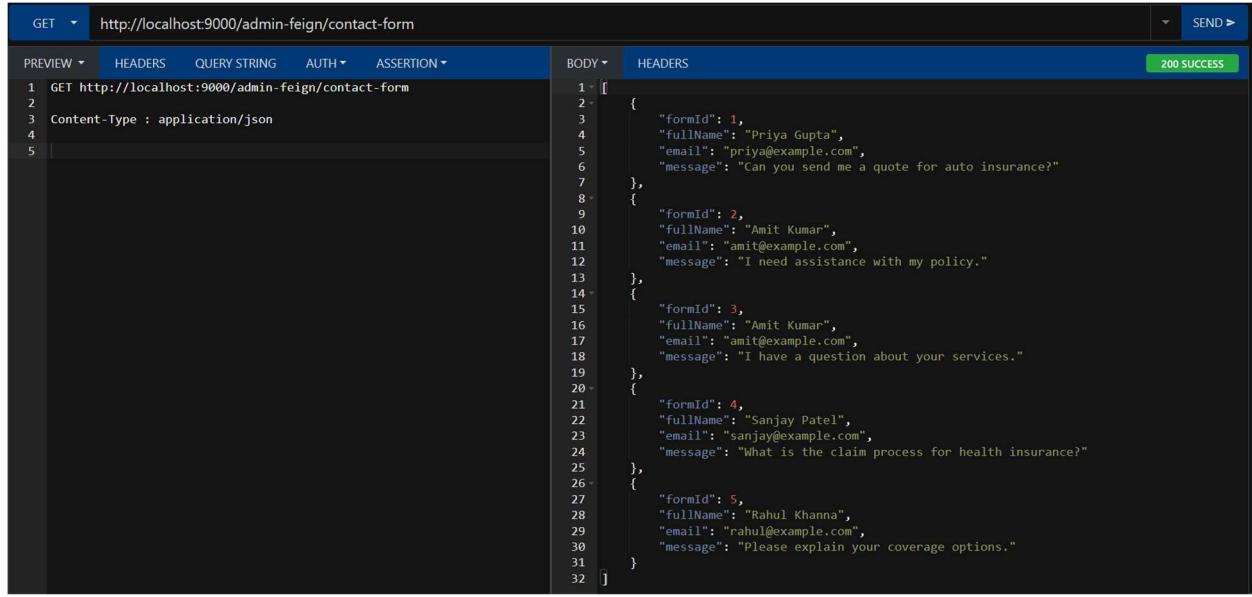
The screenshot shows a POST request to `http://localhost:9000/user-feign/contact-form`. The request body contains the following JSON:

```
1 - {
2     "fullName": "Priya Gupta",
3     "email": "priya@example.com",
4     "message": "Can you send me a quote for auto insurance?"
5 }
6 }
```

The response status is 200 SUCCESS.

Admin viewing the entries in contact form, later contacts them through their e-mail

<http://localhost:9000/admin-feign/contact-form>



PREVIEW ▾ HEADERS QUERY STRING AUTH ASSERTION ▾ BODY ▾ HEADERS 200 SUCCESS

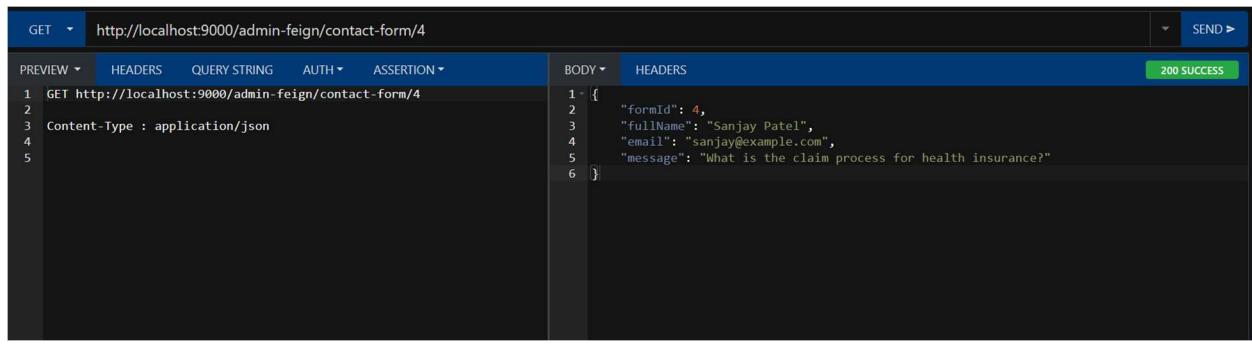
```
1 GET http://localhost:9000/admin-feign/contact-form
2
3 Content-Type : application/json
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32 [ ]
```

The response body contains the following JSON array:

```
[{"formId": 1, "fullName": "Priya Gupta", "email": "priya@example.com", "message": "Can you send me a quote for auto insurance?"}, {"formId": 2, "fullName": "Amit Kumar", "email": "amit@example.com", "message": "I need assistance with my policy."}, {"formId": 3, "fullName": "Amit Kumar", "email": "amit@example.com", "message": "I have a question about your services."}, {"formId": 4, "fullName": "Sanjay Patel", "email": "sanjay@example.com", "message": "What is the claim process for health insurance?"}, {"formId": 5, "fullName": "Rahul Khanna", "email": "rahul@example.com", "message": "Please explain your coverage options."}]
```

1. Contact Form by contact form ID

<http://localhost:9000/admin-feign/contact-form/4>



PREVIEW ▾ HEADERS QUERY STRING AUTH ASSERTION ▾ BODY ▾ HEADERS 200 SUCCESS

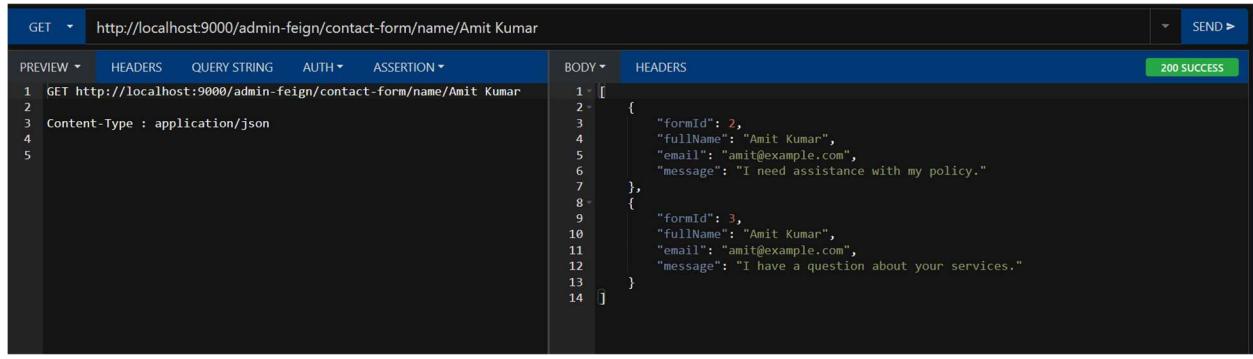
```
1 GET http://localhost:9000/admin-feign/contact-form/4
2
3 Content-Type : application/json
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32 [ ]
```

The response body contains the following JSON object:

```
{ "formId": 4, "fullName": "Sanjay Patel", "email": "sanjay@example.com", "message": "What is the claim process for health insurance?"}
```

2. Contact Form by full name of the person

http://localhost:9000/admin-feign/contact-form/name/Amit Kumar



The screenshot shows a REST client interface with the following details:

- Method: GET
- URL: http://localhost:9000/admin-feign/contact-form/name/Amit Kumar
- Headers:
 - Content-Type : application/json
- Body (JSON array):

```
1 [ ]  
2 [ {  
3     "formId": 2,  
4     "fullName": "Amit Kumar",  
5     "email": "amit@example.com",  
6     "message": "I need assistance with my policy."  
7 },  
8 [ {  
9     "formId": 3,  
10    "fullName": "Amit Kumar",  
11    "email": "amit@example.com",  
12    "message": "I have a question about your services."  
13 }  
14 ]
```
- Response Headers: 200 SUCCESS

Docker Containers

RabbitMQ

```

Microsoft Windows [Version 10_0_22621.2215]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN\>docker run -it --rm --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3.12-management
2023-09-11 03:32:13.15.550169+00:00 [notice] <0.44.0> Application syslog exited with reason: stopped
2023-09-11 03:32:13.15.583643+00:00 [notice] <0.230.0> Logging: switching to configured handler(s); following messages may not be visible in this log output
2023-09-11 03:32:13.15.586914+00:00 [notice] <0.230.0> Logging: configured log handlers are now ACTIVE
2023-09-11 03:32:13.15.565758+00:00 [info] <0.230.0> ra: starting system quorum_queues
2023-09-11 03:32:13.15.565758+00:00 [info] <0.230.0> starting Ra system: quorum_queues in directory: /var/lib/rabbitmq/mnesia/rabbit@fifc049c91bd/quorum/rabbit@fifc049c91bd
2023-09-11 03:32:13.15.565758+00:00 [info] <0.230.0> Ra: meta data store initialised for system quorum_queues. 0 record(s) recovered
2023-09-11 03:32:13.15.565758+00:00 [info] <0.267.0> WAL: ra_log_wal init, open tbls: ra_log_open_mem_tables, closed tbls: ra_log_closed_mem_tables
2023-09-11 03:32:13.15.565758+00:00 [info] <0.230.0> ra: starting Ra system: coordination
2023-09-11 03:32:13.15.565758+00:00 [info] <0.230.0> Starting RabbitMQ 3.12.3 on Erlang 25.3.2.5 [jit]
2023-09-11 03:32:13.15.565758+00:00 [info] <0.230.0> Copyright (c) 2007-2023 VMware, Inc. or its affiliates.
2023-09-11 03:32:13.15.565758+00:00 [info] <0.230.0> Licensed under the MPL 2.0. Website: https://rabbitmq.com

## ## RabbitMQ 3.12.3
## ##
##### Copyright (c) 2007-2023 VMware, Inc. or its affiliates.
#####
## ## Licensed under the MPL 2.0. Website: https://rabbitmq.com

Erlang: 25.3.2.5 [jit]
TLS Library: OpenSSL - OpenSSL 3.1.2 1 Aug 2023
Release series support status: supported

Doc guides: https://rabbitmq.com/documentation.html
Support: https://rabbitmq.com/contact.html
Tutorials: https://rabbitmq.com/getstarted.html
Monitoring: https://rabbitmq.com/monitoring.html

Logs: <stdout>

Config file(s): /etc/rabbitmq/conf.d/18-defaults.conf

Starting broker... 2023-09-11 03:32:15.109917+00:00 [info] <0.230.0>
2023-09-11 03:32:15.109917+00:00 [info] <0.230.0> node : rabbit@fifc049c91bd
2023-09-11 03:32:15.109917+00:00 [info] <0.230.0> home dir : /var/lib/rabbitmq
2023-09-11 03:32:15.109917+00:00 [info] <0.230.0> config file(s) : /etc/rabbitmq/conf.d/18-defaults.conf
2023-09-11 03:32:15.109917+00:00 [info] <0.230.0> cookie hash : E6QCHF3XKGHFWYv+KzVA==
2023-09-11 03:32:15.109917+00:00 [info] <0.230.0> log(s) : <stdout>
2023-09-11 03:32:15.109917+00:00 [info] <0.230.0> data dir : /var/lib/rabbitmq/mnesia/rabbit@fifc049c91bd
2023-09-11 03:32:20.308732+00:00 [info] <0.230.0> Running boot step pre_boot defined by app rabbit
2023-09-11 03:32:29.310661+00:00 [info] <0.230.0> Running boot step rabbit_global_counters defined by app rabbit

```

Zipkin

Viewing Feign Health, Circuit breaker and retry attributes

```
GET http://localhost:9000/actuator/health
Content-Type : application/json
```

```
1 {
  "status": "UP",
  "components": [
    "discoveryClient": {
      "status": "UP",
      "details": {
        "services": [
          "insurance-application-feign",
          "policy-category-service",
          "policy-service",
          "address-service",
          "customer-service",
          "question-service",
          "contact-form-service",
          "admin-service"
        ]
      }
    },
    "eureka": {
      "description": "Remote status from Eureka server",
      "status": "UP",
      "details": {
        "applications": {
          "INSURANCE-APPLICATION-FEIGN": 1,
          "ADDRESS-SERVICE": 1,
          "CONTACT-FORM-SERVICE": 1,
          "CUSTOMER-SERVICE": 1,
          "POLICY-SERVICE": 1,
          "QUESTION-SERVICE": 1,
          "POLICY-CATEGORY-SERVICE": 1,
          "ADMIN-SERVICE": 1
        }
      }
    }
  ],
  "diskSpace": {
    "status": "OK"
  }
}
```

Circuit Breakers

```
GET http://localhost:9000/actuator/circuitbreakers
Content-Type : application/json
```

```
1 {
  "circuitBreakers": [
    "AddressServiceProxy#fallbackForGetAddressById(long,Throwable)",
    "AddressServiceProxy#fallbackForgetAllAddresses(Throwable)",
    "AddressServiceProxy#updateAddress(long,Address)",
    "AddressServiceProxy#getAllAddresses()",
    "AdminServiceProxy#adminLoginCheck(Admin)",
    "AdminServiceProxy#adminLogout(Admin)",
    "AdminServiceProxy#getAdminId(long)",
    "AdminServiceProxy#getAllAdmins()",
    "AdminServiceProxy#registerAdmin(Admin)",
    "AddressServiceProxy#fallbackForGetAddress(long,Address)",
    "ContactFormServiceProxy#fallbackForAddContactForm(ContactForm,Throwable)",
    "CustomerServiceProxy#fallbackForGetCustomerByName(String)",
    "ContactFormServiceProxy#getContactFormById(long)",
    "ContactFormServiceProxy#fallbackForGetCustomers(Throwable)",
    "CustomerServiceProxy#getAllCustomers()",
    "CustomerServiceProxy#getCustomerByEmail(String)",
    "CustomerServiceProxy#getCustomerById(long)",
    "CustomerServiceProxy#registerCustomer(Customer)",
    "PolicyCategoryServiceProxy#addCategory(PolicyCategory)",
    "PolicyCategoryServiceProxy#addPolicyIdToCategory((long,long))",
    "PolicyCategoryServiceProxy#deleteCategoryById((long))",
    "PolicyCategoryServiceProxy#getAllCategories()",
    "PolicyCategoryServiceProxy#getAllCategoriesByNameLike(String)",
    "PolicyCategoryServiceProxy#getCategoryById((long))",
    "PolicyCategoryServiceProxy#movePolicyIdFromCategories((long))",
    "PolicyCategoryServiceProxy#updateCategoryName((long,String))",
    "PolicyCategoryServiceProxy#resetPolicy(Policy)",
    "PolicyServiceProxy#createPolicy(Policy)",
    "PolicyServiceProxy#deleteIssuePolicyByPolicyId((long))",
    "PolicyServiceProxy#getAllPolicies()",
    "PolicyServiceProxy#getAllPoliciesByNameLike(String)",
    "PolicyServiceProxy#getAllPoliciesByCustomerId((long))",
    "PolicyServiceProxy#getPolicyById((long))",
    "PolicyServiceProxy#getName(String)",
    "PolicyServiceProxy#updatePolicy((long,Policy))",
    "PolicyServiceProxy#updatePolicy((long,Policy,int))",
    "QuestionServiceProxy#addQuestion(Question)",
    "QuestionServiceProxy#answerQuestion((long,Question))",
    "QuestionServiceProxy#getAllQuestions()",
    "QuestionServiceProxy#getAllQuestionsByQuestion((Question,Throwable))",
    "QuestionServiceProxy#getAllQuestionsByCustomerId((long))",
    "InsuranceFeignCbs"
  ]
}
```

Retries

```
GET http://localhost:9000/actuator/retries
Content-Type : application/json
```

```
1 {
  "retries": [
    "insurance-feign-retry"
  ]
}
```

Making the Customer, Address and Policy Microservices DOWN

The screenshot shows the Spring Eureka dashboard. At the top, it displays "spring Eureka" and "HOME LAST 1000 SINCE STARTUP". Below this is a "System Status" section with a table showing environment details like "Environment: test", "Data center: default", and various uptime metrics. The "DS Replicas" section lists registered instances under "localhost":

Application	AMIs	Availability Zones	Status
ADMIN-SERVICE	n/a (1)	(1)	UP (1) - admin-service:360d3070c10bfcaba92acdc208498935
CONTACT-FORM-SERVICE	n/a (1)	(1)	UP (1) - contact-form-service:e8ffcafdf594d251b8e82ba8d0f35b247
INSURANCE-APPLICATION-FEIGN	n/a (1)	(1)	UP (1) - host.docker.internal:insurance-application-feign:9000
POLICY-CATEGORY-SERVICE	n/a (1)	(1)	UP (1) - policy-category-service:c7db2fbfb9f36f5d404c322bd3b70add
QUESTION-SERVICE	n/a (1)	(1)	UP (1) - question-service:edede102f65cf2b5ec76878c1895469c

Fallback

The screenshot shows a POSTMAN request to "http://localhost:9000/admin-feign/customer/2". The "PREVIEW" tab shows the JSON response body:

```
1 {
2   "customerId": 2,
3   "customerEmail": "default",
4   "customerName": "default",
5   "customerPassword": "default",
6   "customerAge": 0,
7   "customerGender": "default",
8   "customerMobile": "default",
9   "customerImageUrl": "default",
10  "customerAddress": {
11    "addressId": 2,
12    "streetAddress": "default",
13    "city": "default",
14    "state": "default",
15    "postalcode": "default",
16    "country": "default"
17  }
18 }
```

The "BODY" tab shows an empty array: [].

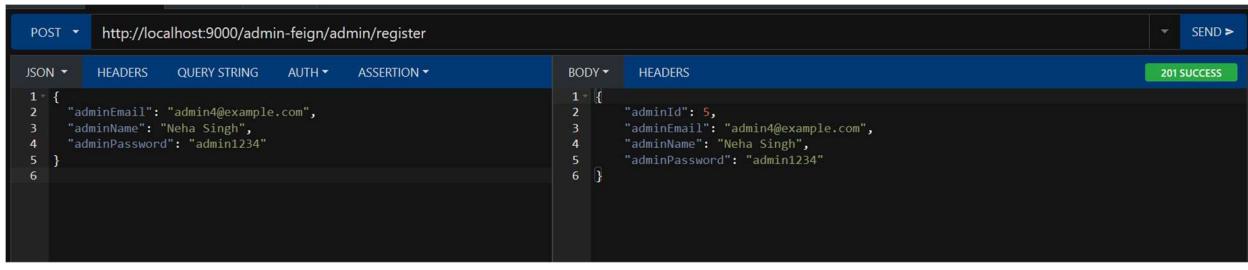
Fallback

The screenshot shows a POSTMAN request to "http://localhost:9000/admin-feign/policy". The "PREVIEW" tab shows the JSON response body:

```
1 [
2 ]
3 ]
```

The "BODY" tab shows an empty array: [].

Success (with admin-service UP)



POST http://localhost:9000/admin-feign/admin/register

JSON Headers Query String Auth Assertion

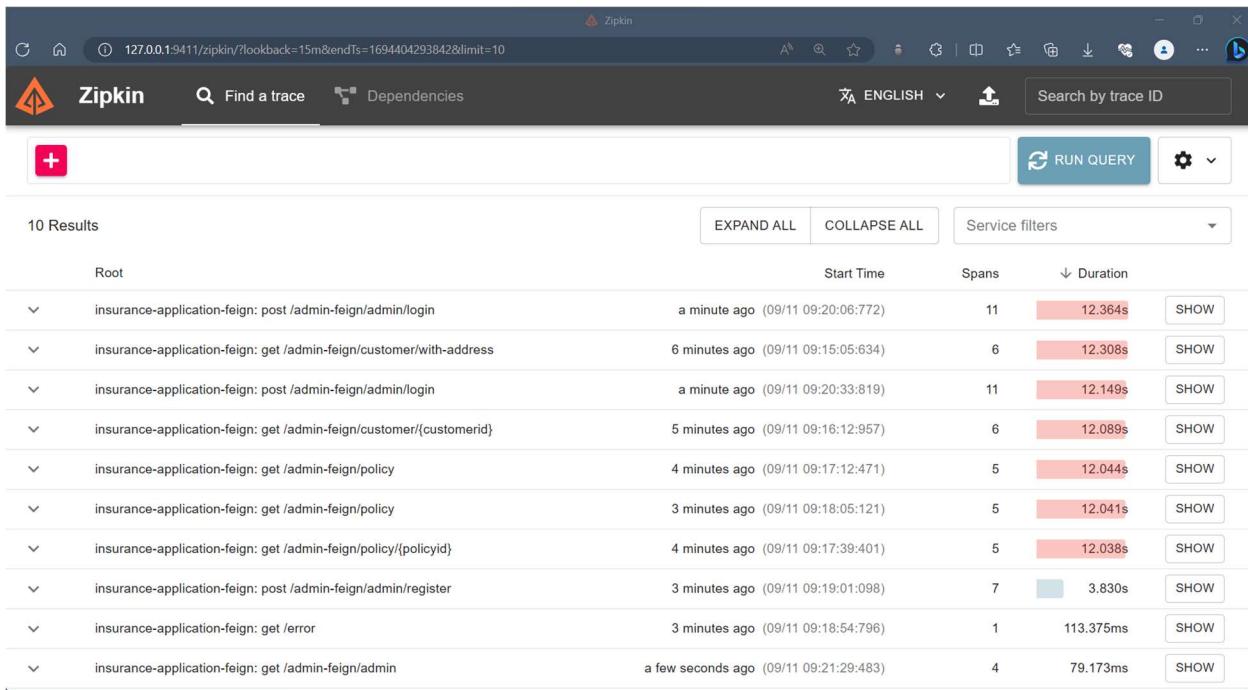
```
1 { "adminEmail": "admin4@example.com",  
2 "adminName": "Neha Singh",  
3 "adminPassword": "admin1234"  
4 }  
5  
6 }
```

BODY Headers

```
1 {  
2 "adminId": 5,  
3 "adminEmail": "admin4@example.com",  
4 "adminName": "Neha Singh",  
5 "adminPassword": "admin1234"  
6 }
```

201 SUCCESS

Viewing Fallbacks and Success events from Zipkin



Zipkin

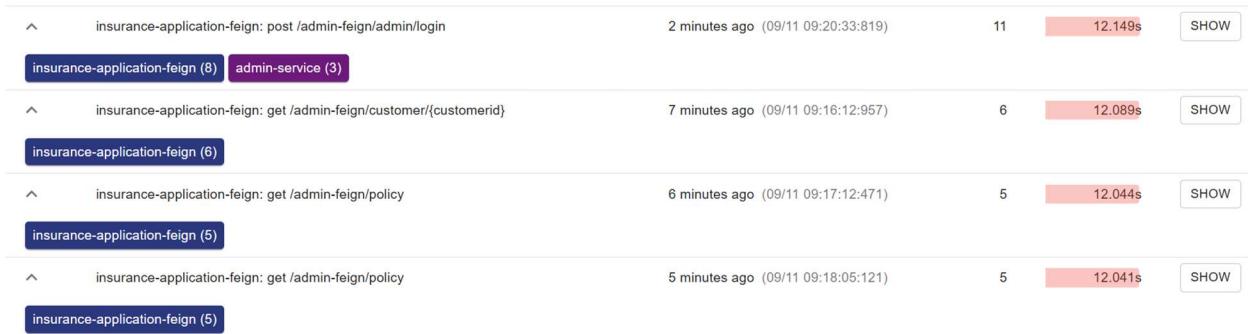
Find a trace Dependencies ENGLISH Search by trace ID

+ RUN QUERY

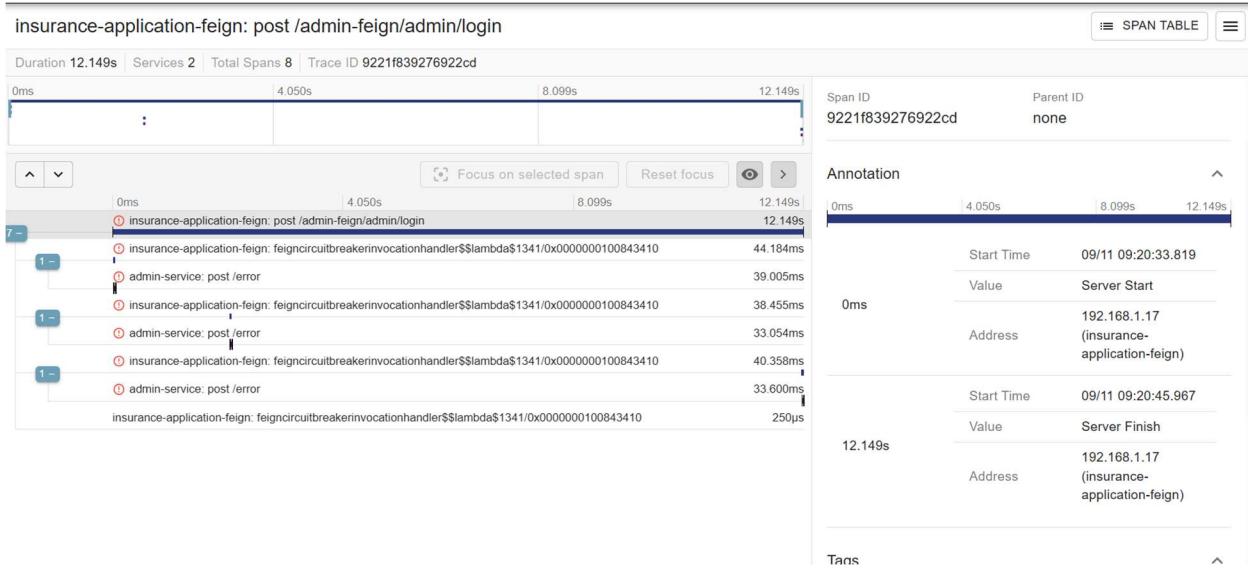
10 Results EXPAND ALL COLLAPSE ALL Service filters

	Start Time	Spans	Duration	Show
Root				
insurance-application-feign: post /admin-feign/admin/login	a minute ago (09/11 09:20:06:772)	11	12.364s	Show
insurance-application-feign: get /admin-feign/customer/with-address	6 minutes ago (09/11 09:15:05:634)	6	12.308s	Show
insurance-application-feign: post /admin-feign/admin/login	a minute ago (09/11 09:20:33:819)	11	12.149s	Show
insurance-application-feign: get /admin-feign/customer/{customerid}	5 minutes ago (09/11 09:16:12:957)	6	12.089s	Show
insurance-application-feign: get /admin-feign/policy	4 minutes ago (09/11 09:17:12:471)	5	12.044s	Show
insurance-application-feign: get /admin-feign/policy	3 minutes ago (09/11 09:18:05:121)	5	12.041s	Show
insurance-application-feign: get /admin-feign/policy/{policyid}	4 minutes ago (09/11 09:17:39:401)	5	12.038s	Show
insurance-application-feign: post /admin-feign/admin/register	3 minutes ago (09/11 09:19:01:098)	7	3.830s	Show
insurance-application-feign: get /error	3 minutes ago (09/11 09:18:54:796)	1	113.375ms	Show
insurance-application-feign: get /admin-feign/admin	a few seconds ago (09/11 09:21:29:483)	4	79.173ms	Show

Failed Admin Login with admin service DOWN



insurance-application-feign: post /admin-feign/admin/login	2 minutes ago (09/11 09:20:33:819)	11	12.149s	Show
insurance-application-feign (8)				
insurance-application-feign: get /admin-feign/customer/{customerid}	7 minutes ago (09/11 09:16:12:957)	6	12.089s	Show
insurance-application-feign (8)				
insurance-application-feign: get /admin-feign/policy	6 minutes ago (09/11 09:17:12:471)	5	12.044s	Show
insurance-application-feign (5)				
insurance-application-feign: get /admin-feign/policy	5 minutes ago (09/11 09:18:05:121)	5	12.041s	Show
insurance-application-feign (5)				



Span Table

Span ID	Service name	Span name	Start time	Duration
9221f839276922cd	insurance-application-feign	post /admin-feign/admin/l...	09/11 09:20:33.819	12.149s
e667a6d009e29481	insurance-application-feign	feigncircuitoerinvocati...	09/11 09:20:33.825	44.184ms
bf370918a63ad10b	admin-service	post /error	09/11 09:20:33.829	39.005ms
ddf4c19fcc1c98a1	insurance-application-feign	feigncircuitoerinvocati...	09/11 09:20:35.878	38.455ms
7276b60f7ae18b06	admin-service	post /error	09/11 09:20:35.880	33.054ms
070ecaf982edcb04	insurance-application-feign	feigncircuitoerinvocati...	09/11 09:20:45.924	40.358ms
ea96992d7a216dac	admin-service	post /error	09/11 09:20:45.928	33.600ms
c5c12ecda1c927b0	insurance-application-feign	feigncircuitoerinvocati...	09/11 09:20:45.965	250.000µs

STS Console

```

:08:13.551 INFO [insurance-application-feign,al93c220c953618,al93c220c953618] 16592 --- [nio-9000-exec-1] o.s.c.openfeign.FeignClientFactoryBean : For 'admin-service' URL not provi
:08:13.755 INFO [insurance-application-feign,al93c220c953618,al93c220c953618] 16592 --- [nio-9000-exec-1] o.s.c.openfeign.FeignClientFactoryBean : For 'customer-service' URL not pr
:08:13.850 INFO [insurance-application-feign,al93c220c953618,al93c220c953618] 16592 --- [nio-9000-exec-1] o.s.c.openfeign.FeignClientFactoryBean : For 'address-service' URL not pr
:08:13.944 INFO [insurance-application-feign,al93c220c953618,al93c220c953618] 16592 --- [nio-9000-exec-1] o.s.c.openfeign.FeignClientFactoryBean : For 'contact-form-service' URL n
:08:14.023 INFO [insurance-application-feign,al93c220c953618,al93c220c953618] 16592 --- [nio-9000-exec-1] o.s.c.openfeign.FeignClientFactoryBean : For 'policy-category-service' URI
:08:14.118 INFO [insurance-application-feign,al93c220c953618,al93c220c953618] 16592 --- [nio-9000-exec-1] o.s.c.openfeign.FeignClientFactoryBean : For 'policy-service' URL not prov
:08:14.227 INFO [insurance-application-feign,al93c220c953618,al93c220c953618] 16592 --- [nio-9000-exec-1] o.s.c.openfeign.FeignClientFactoryBean : For 'question-service' URL not pr

```

Viewing Successful event

Zipkin Find a trace Dependencies ENGLISH Search by trace ID

insurance-application-feign: get /admin-feign/admin

Duration 79.173ms | Services 2 | Total Spans 3 | Trace ID 663fc723e371a220

0ms 26.391ms 52.782ms 79.173ms

Focus on selected span Reset focus

insurance-application-feign: get /admin-feign/admin 0ms 26.391ms 52.782ms 79.173ms
insurance-application-feign: feign.circuitbreaker.invokeHandler\$lambda\$1341\$0x0000000100843410 26.391ms 52.782ms 68.667ms
admin-service: get /admin 52.782ms 68.667ms 64.839ms

Service name: insurance-application-feign, Span name: get /admin-feign/admin, Span ID: 663fc723e371a220, Parent ID: none

Annotation

0ms	26.391ms	52.782ms	79.173ms
Start Time	09/11 09:21:29.483		
Value	Server Start		
Address	192.168.1.17 (insurance-application-feign)		
79.173ms	09/11 09:21:29.563		
Start Time	09/11 09:21:29.563		
Value	Server Finish		
Address	192.168.1.17 (insurance-application-feign)		

Span ID ↑	Service name	Span name	Start time	Duration
663fc723e371a220	insurance-application-feign	get /admin-feign/admin	09/11 09:21:29.483	79.173ms
a3191cdc22a4e420	insurance-application-feign	feign.circuitbreaker.invokeHandler\$lambda\$1341\$0x0000000100843410	09/11 09:21:29.491	68.667ms
c2e4f10876338dee	admin-service	get /admin	09/11 09:21:29.493	64.839ms