

第二次作业-181250090 刘育麟

1. User Service

1.1 User Controller

- 功能：spring中主要与前端交互的接口
- 代码：

```
1  @Autowired
2  private UserService userService;
3
4  @PostMapping("/user")
5  public Object add(@RequestBody User user){
6      return userService.add(user);
7  }
```

- 代码说明：
 - 作为前端的接口，调用后端的service模块完成前端的请求。
 - @Autowired：获取spring框架中对应的service作为成员变量
 - @PostMapping：获取前端/user位置的写入请求
 - @RequestBody：获取写入请求夹带的数据包。

1.2 User Service

- 功能：spring中实现后端逻辑的模块。
- 代码：

```
1  public Object add(User user){
2      return userDao.save(user);
3  }
```

- 代码说明：
 - 调用spring中与数据库交互的接口的方法save在数据库储存一个user对象。

1.3 User Dao

- 功能：spring中实现与数据库交互。
- 代码：

```
1  public interface UserDao extends JpaRepository<User, Long> {
2  }
```

- 代码说明：
 - 调用spring自带的bean来对数据库进行增删改查。

1.4 User

- 功能：定义user数据库
- 代码：

```

1  @Entity
2  @Table(name="user")
3  public class User {
4      @Id
5      @GeneratedValue(strategy=GenerationType.AUTO)
6      private Long id;
7
8      @Column(length = 32)
9      private String name;
10
11     @Column(length = 64)
12     private String pwd;
13 }

```

- 代码说明：
 - 对数据库中的user表进行操作。
 - @Entity：表明该类为一个实体类。
 - @Table：表明映射到哪一个表。
 - @GeneratedValue：启动服务时自动创建表user。
 - @Id：表示主键。
 - @Column：表示需要的列。

2. Admin Service

2.1 User Controller

- 功能：spring中主要与前端交互的接口
- 代码：

```

1  @Autowired
2  private UserService userService;
3
4  @PostMapping("/user")
5  public Object add(@RequestBody @Valid UserDto userDto){
6      return userService.add(userDto);
7  }

```

- 代码说明：
 - 作为前端的接口，调用后端的service模块完成前端的请求。
 - @Autowired：获取spring框架中对应的service作为成员变量。
 - @PostMapping：获取前端/user位置的写入请求。
 - @RequestBody：获取写入请求夹带的数据包。
 - @Valid：添加验证信息的要求。

2.2 User Service

- 功能：spring中实现后端逻辑的模块。
- 代码：

```

1  @Autowired
2  private UserFeign userFeign;
3
4  public Object add(UserDto userDto){
5      //远程http调用
6      return userFeign.add(userDto);
7  }

```

- 代码说明：
 - 远程调用EUREKA中的user service。

2.3 User Service

- 功能：实现admin service与user service交互的接口。
- 代码：

```

1  @FeignClient(name = "user-service")
2  public interface UserFeign {
3
4      @PostMapping("/user")
5      Object add(UserDto userDto);
6
7      @GetMapping("/port")
8      String port();
9  }
10

```

- 代码说明：
 - @FeignClient：指定调用user service服务。
 - @PostMapping：数据写入请求。
 - @GetMapping：数据读取请求。

2.4 UserDto

- 功能：定义从前端获取的user类
- 代码：

```

1  @NotBlank(message = "用户名不能为空")
2  private String name;
3
4  @NotBlank(message = "密码不能为空")
5  private String pwd;

```

- 代码说明：
 - @NotBlank：判断传进来的数据是否为空。

2.5 CustomRule

- 功能：控制负载均衡的类
- 代码：

```

1  public class CustomRule extends AbstractLoadBalancerRule {
2
3      @Override

```

```

4      public Server choose(Object key) {
5
6          ILoadBalancer loadBalancer = getLoadBalancer();
7
8          //获取所有可达服务器列表
9          List<Server> servers = loadBalancer.getReachableServers();
10         if (servers.isEmpty()) {
11             return null;
12         }else{
13             return servers.get(((int)(Math.random() * 1000)) %
servers.size());
14         }
15     }
16
17 }

```

- 代码说明：
 - 负载均衡的调配原则为随机获取服务并返回。
 - AbstractLoadBalancerRule：调配负载均衡的抽象类。

2.6 Handler

- 功能：定义报错时的返回值
- 代码：

```

1  public class Handler {
2
3
4      @org.springframework.web.bind.annotation.ExceptionHandler(MethodArgumentNo
tValidException.class)
5      @ResponseStatus(HttpStatus.BAD_REQUEST)
6      @ResponseBody
7      public Map methodArgumentNotValidExceptionHandler(HttpServletRequest request, MethodArgumentNotValidException e) {
8          Map result = new HashMap();
9          result.put("error",
e.getBindingResult().getAllErrors().get(0).getDefaultMessage());
10         return result;
11     }
12 }

```

- 代码说明：
 - 获取报错时UserDto传出的message信息并返回给前端。
 - @ResponseStatus：返回状态
 - @ResponseBody：返回的数据

3. Eureka Server

3.1 server的application参数配置

```

1  server.port=8080
2
3  eureka.client.service-url.defaultZone=http://localhost:8080
4  eureka.client.fetch-registry=false
5  eureka.client.register-with-eureka=false

```

3.2 client的application参数配置 - admin service

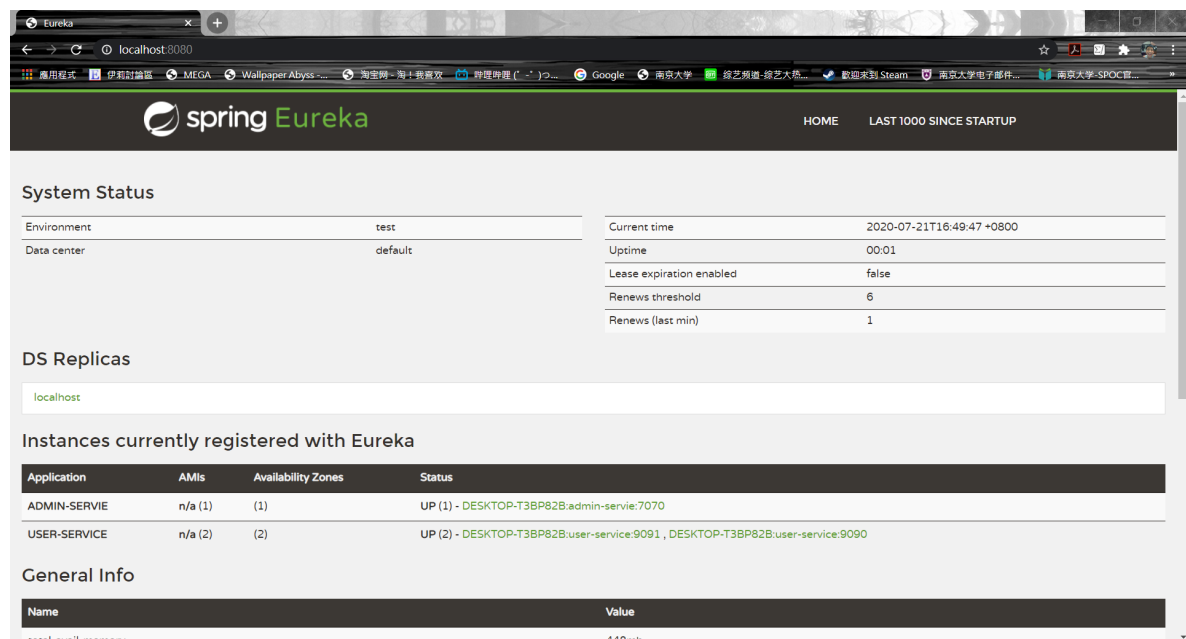
```
1 server.port=7070
2 spring.application.name=admin-servie
3
4 eureka.client.service-
url.defaultZone=${EUREKA_URL:http://localhost:8080/eureka}
5 eureka.instance.prefer-ip-address=true
```

3.3 client的application参数配置 - user service

```
1 spring.application.name=user-service
2 server.port=9090
3
4 spring.datasource.url=jdbc:mysql://localhost:3306/user?
charset=utf8mb4&useSSL=false&allowPublicKeyRetrieval=true
5 spring.datasource.username=root
6 spring.datasource.password=willy2291iu
7 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
8
9 spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
10 spring.jpa.show-sql=true
11 spring.jpa.hibernate.ddl-auto=create
12
13 eureka.client.service-
url.defaultZone=${EUREKA_URL:http://localhost:8080/eureka}
14 eureka.instance.prefer-ip-address=true
```

4. 运行截图

4.1 将服务部署到eureka上



The screenshot shows the Spring Eureka web interface in a browser. The page has a dark header with the Spring Eureka logo and navigation 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	Current time	2020-07-21T16:49:47 +0800
Data center	default	Uptime	00:01
		Lease expiration enabled	false
		Renews threshold	6
		Renews (last min)	1
- DS Replicas:** A section showing the local host as a replica.
- Instances currently registered with Eureka:** A table listing registered services.

Application	AMIs	Availability Zones	Status
ADMIN-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-T3BP82B\admin-servie:7070
USER-SERVICE	n/a (2)	(2)	UP (2) - DESKTOP-T3BP82B:user-service:9091, DESKTOP-T3BP82B:user-service:9090
- General Info:** A table showing system metrics.

Name	Value
total-avail-memory	448mb

4.2 通过admin service注册user

My Workspace

Invite

Launchpad

POST http://localhost:7070/user

No Environment

Untitled Request

Comments 0

POST http://localhost:7070/user

Send

Save

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Cookies

Code

Beautifuly

```
1 {
2   "name": "123",
3   "pwd": "123"
4 }
```

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 443 ms

Size: 197 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": 1,
3   "name": "123",
4   "pwd": "123"
5 }
```

Bootcamp

Build

Browse

My Workspace

Invite

Launchpad

POST http://localhost:7070/user

No Environment

Untitled Request

Comments 0

POST http://localhost:7070/user

Send

Save

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Cookies

Code

Beautifuly

```
1 {
2   "name": "",
3   "pwd": "123"
4 }
```

Body

Cookies

Headers (4)

Test Results

Status: 400 Bad Request

Time: 26 ms

Size: 177 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "error": "用户名不能为空"
3 }
```

Bootcamp

Build

Browse

My Workspace Invite Upgrade

Launchpad POST http://localhost:7070/user No Environment

Untitled Request Comments 0

POST http://localhost:7070/user Send Save

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Code

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

```
1 {
2   "name": "1234",
3   "pwd": ""
4 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 6 ms Size: 174 B Save Response

Pretty Raw Preview Visualize JSON Beautify

```
1 {
2   "error": "密码不能为空"
3 }
```

Bootcamp Build Browse Help

4.3 通过user service注册user

My Workspace Invite Upgrade

Launchpad POST http://localhost:9090/user No Environment

Untitled Request Comments 0

POST http://localhost:9090/user Send Save

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Code

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

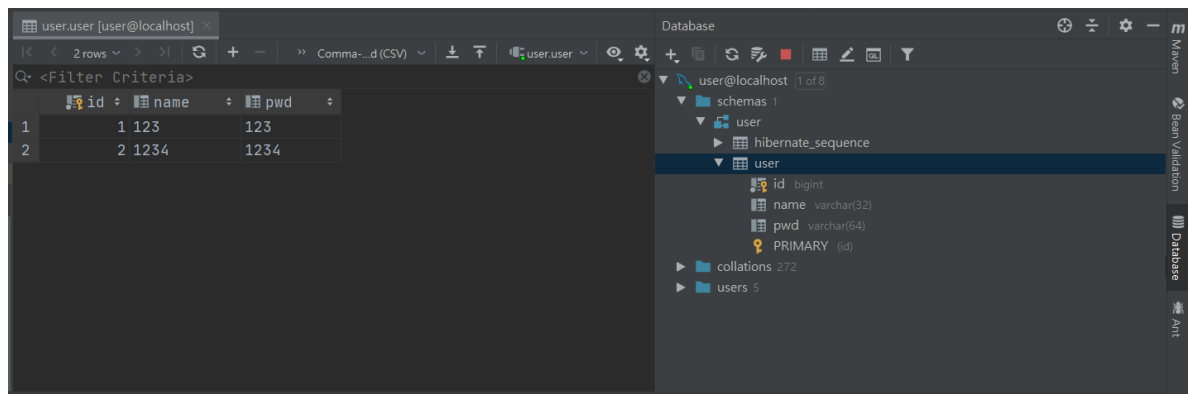
```
1 {
2   "name": "1234",
3   "pwd": "1234"
4 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 158 ms Size: 199 B Save Response

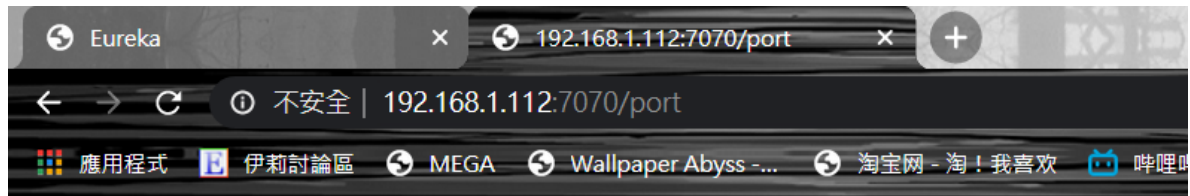
Pretty Raw Preview Visualize JSON Beautify

```
1 {
2   "id": 2,
3   "name": "1234",
4   "pwd": "1234"
5 }
```

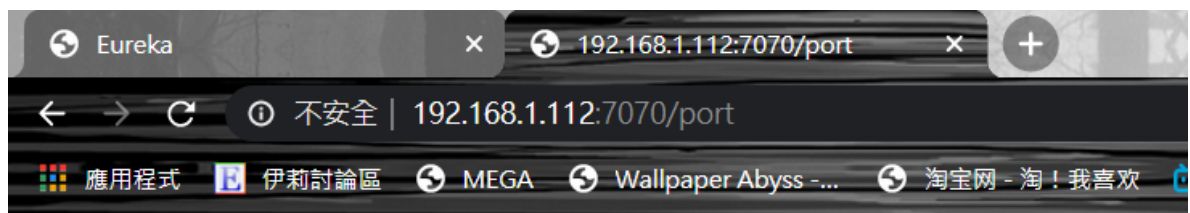
Bootcamp Build Browse Help



4.4 负载均衡测试



9091



9090