外卖订餐项目

1、项目需求

客户端：针对普通用户，用户登录，用户退出，菜品订购，我的订单

后台管理系统：针对管理员，管理员登录，管理员退出，添加菜品，查询菜品，删除菜品，订单处理，添加用户，查询用户，删除用户

2、微服务拆分（业务）

2.1、服务提供者

account 提供账户服务：用户和管理员的登录退出。

menu 提供菜品服务：添加菜品，删除菜品，修改菜品，查询菜品

order 提供订单服务：添加订单，查询订单，删除订单，处理订单

user 提供用户服务：添加用户、查询用户，删除用户

2.2、服务消费者

分离出一个服务消费者，调用以上4个服务提供者。

服务消费者包含了客户端的前端页面和后台接口、后台管理系统的前端页面和后台接口。

用户/管理员直接访问的资源都保存在服务消费者中，服务消费者根据具体的需要调用四个服务提供者的业务逻辑，通过Feign实现负载均衡。

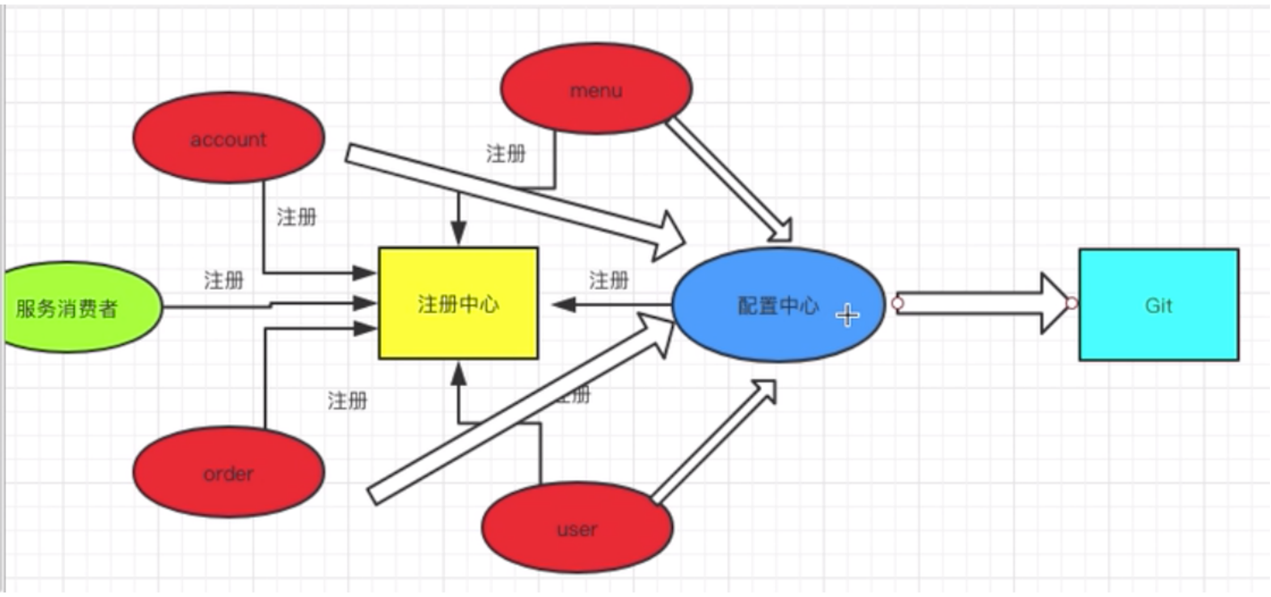
2.3、注册中心

四个服务提供者和一个服务消费者都需要在注册中心进行注册

2.4、配置中心

同时可以使用配置中心来对配置文件进行统一集中管理

2.5、微服务流程



3、项目创建

3.1、创建父工程

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.ww</groupId>

 <artifactId>aispringclouddemo</artifactId>
    <packaging>pom</packaging>
    <version>1.0-SNAPSHOT</version>
    <modules>
        <module>eurekaserver</module>
        <module>configserver</module>
        <module>order</module>
    </modules>
    <!--spring boot-->
    <parent>

 <groupId>org.springframework.boot</groupId
>
        <artifactId>spring-boot-starter-
parent</artifactId>
        <version>2.0.7.RELEASE</version>
    </parent>

    <dependencies>
        <dependency>

 <groupId>org.springframework.boot</groupId
>
```

```
            <artifactId>spring-boot-
starter-web</artifactId>
        </dependency>
        <!--jdk 11 缺失jar-->
        <dependency>

 <groupId>javax.xml.bind</groupId>
            <artifactId>jaxb-
api</artifactId>
        </dependency>
        <dependency>

 <groupId>com.sun.xml.bind</groupId>
            <artifactId>jaxb-
impl</artifactId>
            <version>2.3.0</version>
        </dependency>
        <dependency>

 <groupId>com.sun.xml.bind</groupId>
            <artifactId>jaxb-
core</artifactId>
            <version>2.3.0</version>
        </dependency>
        <dependency>
 <groupId>javax.activation</groupId>

 <artifactId>activation</artifactId>
            <version>1.1.1</version>
        </dependency>
        <dependency>

 <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <optional>true</optional>
        </dependency>
    </dependencies>
<!--springcloud 相关管理-->
    <dependencyManagement>
        <dependencies>
            <dependency>

 <groupId>org.springframework.cloud</groupI
d>
                <artifactId>spring-cloud-
dependencies</artifactId>

 <version>Finchley.SR2</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>
</project>
```

3.2、搭建注册中心：eurekaserver

3.2.1、pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project
xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"

xsi:schemaLocation="http://maven.apache.org
/POM/4.0.0
http://maven.apache.org/xsd/maven-
4.0.0.xsd">
    <parent>

 <artifactId>aispringclouddemo</artifactId>
        <groupId>com.ww</groupId>
        <version>1.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <artifactId>eurekaserver</artifactId>
    <dependencies>
        <dependency>

 <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>

 <version>2.0.2.RELEASE</version>
        </dependency>
    </dependencies>

</project>
```

3.2.2、application.yml

```
server:
  port: 8761
eureka:
  client:
    service-url:
      defaultZone:
http://localhost:8761/eureka/
    register-with-eureka: false
    fetch-registry: false
```

3.2.3、EurekaServerApplication.java

```java
package com.ww;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication
@EnableEurekaServer
public class EurekaServerApplication {
    public static void main(String[] args) {
        SpringApplication.run(EurekaServerApplication.class,args);
    }
}
```

3.2.4、测试

http://localhost:8761/

3.3、创建配置中心：configserver

3.3.1、pom.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <parent>
        <artifactId>aispringclouddemo</artifactId>
        <groupId>com.ww</groupId>
        <version>1.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <artifactId>configserver</artifactId>
    <dependencies>
        <dependency>
```

```
 <groupId>org.springframework.cloud</groupI
d>
            <artifactId>spring-cloud-
config-server</artifactId>

 <version>2.0.2.RELEASE</version>
        </dependency>
    </dependencies>
</project>
```

3.3.2、application.yml

```
server:
  port: 8762
spring:
  application:
    name: configserver
  profiles:
      active: native
  cloud:
    config:
      server:
        native:
          search-locations:
classpath:/shared
```

在shared路径下创建各个微服务对应的配置文件（如：order-dev.yml）

order-dev.yml

```
server:
  port: 8010
```

3.3.3、ConfigServerApplication.java

```
package com.ww;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.config.server.EnableConfigServer;

@SpringBootApplication
@EnableConfigServer
public class ConfigServerApplication {
    public static void main(String[] args) {
        SpringApplication.run(ConfigServerApplication.class, args);
    }
}
```

3.4、服务提供者：order

3.4.1、pom.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project
xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"

xsi:schemaLocation="http://maven.apache.org
/POM/4.0.0
http://maven.apache.org/xsd/maven-
4.0.0.xsd">
    <parent>

 <artifactId>aispringclouddemo</artifactId>
        <groupId>com.ww</groupId>
        <version>1.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <artifactId>order</artifactId>
    <dependencies>
        <dependency>
```

```
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
            <version>2.0.2.RELEASE</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-config</artifactId>
            <version>2.0.2.RELEASE</version>
        </dependency>

    </dependencies>

</project>
```

3.4.2、bootstrap.yml

```yaml
spring:
  application:
    name: order
  profiles:
    active: dev
  cloud:
    config:
      uri: http://localhost:8762
      fail-fast: true
```

3.4.3、测试类

OrderHandler.java

```java
package com.ww.controller;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/order")
public class OrderHandler {
    @Value("${server.port}")
    private String port;
    @GetMapping("/index")
    public String index(){
        return "order端口："+this.port;
    }
}
```

OrderApplication.java

```
package com.ww;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class OrderApplication {
    public static void main(String[] args) {

        SpringApplication.run(OrderApplication.class,args);
    }
}
```

3.5、分别启动服务

注册中心服务：

配置中心服务：

服务提供者order:

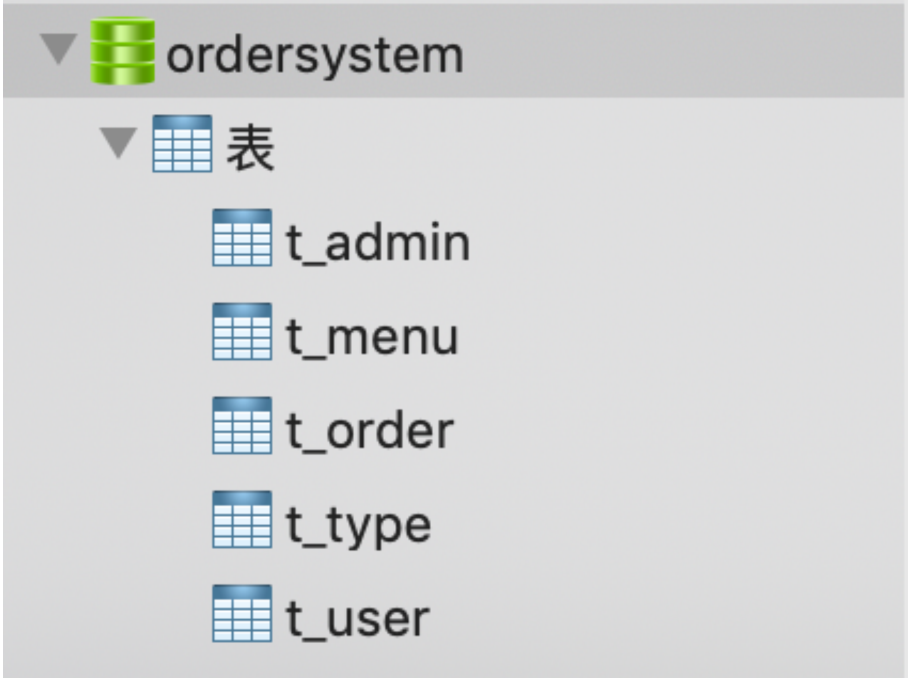
http://localhost:8761/

http://localhost:8010/order/index

4、数据库设计

ordersystem

------t_admin/t_menu/t_order/t_type/t_user



ordersystem

表

t_admin

t_menu

t_order

t_type

t_user

```
/*
 Navicat MySQL Data Transfer

 Source Server         : localhost
 Source Server Type    : MySQL
 Source Server Version : 80011
 Source Host           : localhost
 Source Database       : ordersystem

 Target Server Type    : MySQL
 Target Server Version : 80011
 File Encoding         : utf-8

 Date: 10/16/2020 21:57:49 PM
*/

SET NAMES utf8;
SET FOREIGN_KEY_CHECKS = 0;

-- ----------------------------
--  Table structure for `t_admin`
-- ----------------------------
DROP TABLE IF EXISTS `t_admin`;
CREATE TABLE `t_admin` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT
CHARSET=utf8;

-- ----------------------------
--  Records of `t_admin`
-- ----------------------------
BEGIN;
```

```
INSERT INTO `t_admin` VALUES ('1', 'admin',
'admin');
COMMIT;

-- ----------------------------
--  Table structure for `t_menu`
-- ----------------------------
DROP TABLE IF EXISTS `t_menu`;
CREATE TABLE `t_menu` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) DEFAULT NULL,
  `price` decimal(10,0) DEFAULT NULL,
  `flavor` varchar(255) DEFAULT NULL,
  `tid` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT
CHARSET=utf8;

-- ----------------------------
--  Records of `t_menu`
-- ----------------------------
BEGIN;
```

```
INSERT INTO `t_menu` VALUES ('1', '香酥鸡', '34', '香', '1'), ('2', '梅菜扣肉', '65', '香', '1'), ('3', '三杯鸡', '33', '香', '1'), ('4', '毛血旺', '68', '辣', '1'), ('5', '菠菜抖粉丝', '12', '辣', '2'), ('6', '凉拌豆腐皮', '34', '苦', '2'), ('7', '酱牛肉', '32', '微辣', '2'), ('8', '鱼头豆腐汤', '32', '香', '3'), ('9', '瘦肉鸡蛋白菜汤', '21', '甜', '3'), ('10', '西葫芦虾仁蒸饺', '10', '香', '4'), ('11', '蛋炒饭', '15', '咸', '4'), ('12', '椰蓉面包', '8', '香', '5');
COMMIT;

-- ----------------------------
--  Table structure for `t_order`
-- ----------------------------
DROP TABLE IF EXISTS `t_order`;
CREATE TABLE `t_order` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `uid` int(11) DEFAULT NULL,
  `mid` int(11) DEFAULT NULL,
  `aid` int(11) DEFAULT NULL,
  `date` date DEFAULT NULL,
  `state` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT
CHARSET=utf8;

-- ----------------------------
--  Records of `t_order`
-- ----------------------------
BEGIN;
INSERT INTO `t_order` VALUES ('1', '1',
'7', '1', '2020-10-07', '1'), ('2', '1',
'2', '1', '2020-10-07', '1'), ('3', '1',
'5', '1', '2020-10-07', '1'), ('4', '1',
'9', '1', '2020-10-07', '1'), ('5', '1',
'10', '1', '2020-10-07', '1'), ('6', '1',
'10', null, '2020-10-07', '0'), ('7', '1',
'10', '1', '2020-10-07', '1'), ('8', '1',
'6', '1', '2020-10-07', '1'), ('9', '1',
'10', '1', '2020-10-07', '1'), ('10', '1',
'7', '1', '2020-10-07', '1'), ('11', '2',
'8', null, '2020-10-07', '0'), ('12', '2',
'12', null, '2020-10-07', '0');
COMMIT;

-- ----------------------------
--  Table structure for `t_type`
-- ----------------------------
DROP TABLE IF EXISTS `t_type`;
```

```
CREATE TABLE `t_type` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT
CHARSET=utf8;

-- ----------------------------
--  Records of `t_type`
-- ----------------------------
BEGIN;
INSERT INTO `t_type` VALUES ('1', '热菜'),
('2', '凉菜'), ('3', '烫羹'), ('4', '主食'),
('5', '烘焙');
COMMIT;

-- ----------------------------
--  Table structure for `t_user`
-- ----------------------------
DROP TABLE IF EXISTS `t_user`;
CREATE TABLE `t_user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  `nickname` varchar(255) DEFAULT NULL,
  `gender` varchar(255) DEFAULT NULL,
```

```sql
  `telephone` varchar(255) DEFAULT NULL,
  `registerdate` date DEFAULT NULL,
  `address` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT
CHARSET=utf8;

-- ----------------------------
--  Records of `t_user`
-- ----------------------------
BEGIN;
INSERT INTO `t_user` VALUES ('1',
'zhangsan', 'zhangsan', '张三', '男',
'17890980967', '2020-09-09', '北京'), ('2',
'lisi', 'lisi', '李四', '女', '18867851543',
'2020-10-10', '天津');
COMMIT;

SET FOREIGN_KEY_CHECKS = 1;
```

5、服务提供者-menu

5.1、配置环境

5.1.1、pom.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <parent>
        <artifactId>aispringclouddemo</artifactId>
        <groupId>com.ww</groupId>
        <version>1.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <artifactId>menu</artifactId>
<dependencies>
    <dependency>
```

```xml
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
        <version>2.0.2.RELEASE</version>
    </dependency>
    <dependency>
        <groupId>org.mybatis.spring.boot</groupId>
        <artifactId>mybatis-spring-boot-starter</artifactId>
        <version>1.3.1</version>
    </dependency>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.11</version>
    </dependency>

    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-config</artifactId>
        <version>2.0.2.RELEASE</version>
    </dependency>
</dependencies>

</project>
```

5.1.2、menu-dev.yml

configserver->shared->menu-dev.yml

```
server:
  port: 8020
```

5.1.3、bootstrap.yml

```
spring:
  application:
    name: menu
  profiles:
    active: dev
  cloud:
    config:
      uri: http://localhost:8762
      fail-fast: true
```

5.1.4、测试类

MenuApplication.java

```java
package com.ww;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class MenuApplication {
    public static void main(String[] args) {
        SpringApplication.run(MenuApplication.class,args);
    }
}
```

MenuHandler.java

```java
package com.ww.controller;
```

```java
import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/menu")
public class MenuHandler {
    @Value("${server.port}")
    private String port;
    @GetMapping("/index")
    public String index(){
        return "menu端口："+this.port;
    }
}
```

5.1.5、分别启动服务

注册中心服务：

配置中心服务：

服务提供者order:

服务提供者menu:

异常:

```
***************************
APPLICATION FAILED TO START
***************************

Description:

Failed to configure a DataSource: 'url'
attribute is not specified and no embedded
datasource could be configured.

Reason: Failed to determine a suitable
driver class


Action:

```

```
Consider the following:
  If you want an embedded database (H2,
HSQL or Derby), please put it on the
classpath.
  If you have database settings to be
loaded from a particular profile you may
need to activate it (no profiles are
currently active).


Process finished with exit code 1
```

解决方案：

配置数据源（menu-dev.yml）

```
server:
  port: 8020
spring:
  application:
    name: menu
  datasource:
    name: ordersystem
    url:
jdbc:mysql://localhost:3306/ordersystem
    username: root
    password: Root123456
```

```
###可加，可不加
eureka:
  client:
    service-url:
      defaultZone: http://localhost:8761/eureka/
  instance:
    prefer-ip-address: true
```

http://localhost:8761/

http://localhost:8010/order/index

http://localhost:8020/menu/index

5.2、数据连接测试

5.2.1、配置实体类

菜单对应的数据库表为：t_menu

Menu.java

```
package com.ww.entity;

import lombok.Data;

@Data
public class Menu {
    private long id;
    private String name;
    private double price;
    private String flavor;
}
```

5.2.2、编写CRUD接口

MenuRepository.java

```
package com.ww.repository;

import com.ww.entity.Menu;

import java.util.List;

public interface MenuRepository {
    public List<Menu> findAll(int index, int limit);
    public int count();

    public void save(Menu menu);
    public void update(Menu menu);
    public void delete(long id);
    public Menu findById(long id);
}
```

5.2.3、配置CRUD映射文件

menu->resources->mapping->MenuRepository.xml

MenuRepository.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.ww.repository.MenuRepository">
    <!-- 配置查询所有操作 -->
    <select id="findAll" resultType="com.ww.entity.Menu">
        select * from t_menu limit #{param1},#{param2}
    </select>
    <select id="count" resultType="int">
```

```
        select count(id) from t_menu
    </select>
    <select id="findById"
parameterType="long"
resultType="com.ww.entity.Menu">
        select * from t_menu where id=#{id}
    </select>
    <insert id="save"
parameterType="com.ww.entity.Menu">
        insert into
t_menu(mame,price,flavor)values(#{name},#
{price},#{flavor})
    </insert>
    <update id="update"
parameterType="com.ww.entity.Menu">
        update t_menu set name=#
{name},price=#{price},flavor=#{flavor}
where id=#{id}
    </update>
    <delete id="delete"
parameterType="long">
        delete from t_menu where id=#{id}
    </delete>
</mapper>
```

5.2.4、修改配置文件menu-dev.yml

menu-dev.yml

```
server:
  port: 8020
spring:
  application:
    name: menu
  datasource:
    name: ordersystem
    url: jdbc:mysql://localhost:3306/ordersystem
    username: root
    password: Root123456
mybatis:
  mapper-locations: classpath:/mapping/*.xml
  type-aliases-package: com.ww.entity #配置文件中可省略前缀
```

5.2.5、修改MenuHandler.java

MenuHandler.java

```
package com.ww.controller;
```

```java
import com.ww.entity.Menu;
import com.ww.repository.MenuRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
@RequestMapping("/menu")
public class MenuHandler {
    @Value("${server.port}")
    private String port;

    @Autowired
    private MenuRepository menuRepository;

    @GetMapping("/index")
    public String index(){
        return "menu端口："+this.port;
    }

    @GetMapping("/findAll/{index}/{limit}")
    public List<Menu> findAll(@PathVariable("index") int index,@PathVariable("limit") int limit){
        return menuRepository.findAll(index,limit);
    }
}
```

配置对象扫描

```java
package com.ww;

import org.mybatis.spring.annotation.MapperScan;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
@MapperScan("com.ww.repository")
public class MenuApplication {
    public static void main(String[] args) {
        SpringApplication.run(MenuApplication.class,args);
    }
}
```

5.2.6、分别启动服务

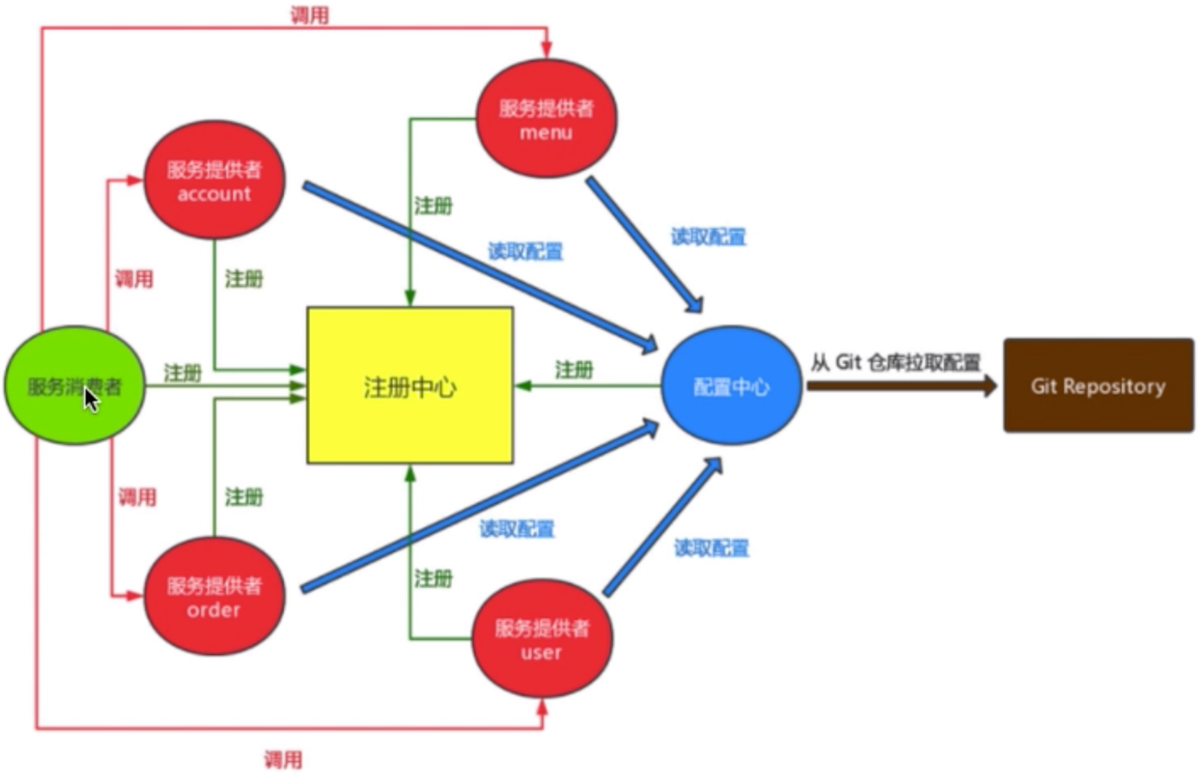
注册中心服务：

配置中心服务：

服务提供者menu:

http://localhost:8020/menu/findAll/0/10

6、服务消费者-client->menu



6.1、创建服务消费者模块-client

6.1.1、pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project
xmlns="http://maven.apache.org/POM/4.0.0"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"

xsi:schemaLocation="http://maven.apache.org
/POM/4.0.0
http://maven.apache.org/xsd/maven-
4.0.0.xsd">
    <parent>

 <artifactId>aispringclouddemo</artifactId>
        <groupId>com.ww</groupId>
        <version>1.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <artifactId>client</artifactId>
    <dependencies>
        <dependency>

 <groupId>org.springframework.cloud</groupI
d>
            <artifactId>spring-cloud-
starter-netflix-eureka-client</artifactId>

 <version>2.0.2.RELEASE</version>
```

```
        </dependency>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-openfeign</artifactId>
            <version>2.0.2.RELEASE</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-config</artifactId>
            <version>2.0.2.RELEASE</version>
        </dependency>
```

```
    </dependencies>

</project>
```

6.1.2、bootstrap.yml

```
spring:
  application:
    name: client
  profiles:
    active: dev
  cloud:
    config:
      uri: http://localhost:8762
      fail-fast: true
```

6.1.3、client-dev.yml

```
server:
  port: 8030
spring:
  application:
    name: client
    thymeleaf:
      prefix: classpath:/static/
      suffix: .html
```

6.1.4、测试类

ClientApplication.java

```java
package com.ww;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class ClientApplication {
    public static void main(String[] args) {
        SpringApplication.run(ClientApplication.class,args);
    }
}
```

Menu.java

```java
package com.ww.entity;

import lombok.Data;

@Data
public class Menu {
    private long id;
    private String name;
    private double price;
    private String flavor;
}
```

MenuFeign.java

```java
package com.ww.feign;

import com.ww.entity.Menu;
import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import java.util.List;

@FeignClient(value="menu")
public interface MenuFeign {

 @GetMapping("/menu/findAll/{index}/{limit}")
    public List<Menu>
findAll(@PathVariable("index") int index,
@PathVariable("limit") int limit);
}
```

ClientHandler.java

```
package com.ww.controller;

import com.ww.entity.Menu;
import com.ww.feign.MenuFeign;
import
org.springframework.beans.factory.annotation.Autowired;
import
org.springframework.web.bind.annotation.GetMapping;
```

```java
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
@RequestMapping("/client")
public class ClientHandler {
    @Autowired
    private MenuFeign menuFeign;
    @GetMapping("/findAll/{index}/{limit}")
    public List<Menu> findAll(@PathVariable("index") int index, @PathVariable("limit") int limit){
        return menuFeign.findAll(index,limit);
    }
}
```

修改ClientApplication.java

```java
package com.ww;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.openfeign.EnableFeignClients;

@SpringBootApplication
@EnableFeignClients
public class ClientApplication {
    public static void main(String[] args) {

        SpringApplication.run(ClientApplication.class,args);
    }
}
```

6.1.5、分别启动服务

注册中心服务：

配置中心服务：

服务提供者menu:

服务消费者client:

http://localhost:8030/client/findAll/0/10

6.2、前端展示（查询所有数据）

6.2.1、前端UI框架

创建static文件夹和index.html页面

Layui

下载：https://www.layui.com/

解压后将框架放入到static.layui文件夹（client->static）

Jquery

下载：https://jquery.com/download/ 或http://www.ab173.com/jquery/

解压后将框架放入到static.js文件夹（client->static）

6.2.2、index.html

thymeleaf：Thymeleaf 最为显著的特征是增强属性，任何属性都可以通过th:xx 来完成交互，例如th:value最终会覆盖value属性。

教程：https://www.jianshu.com/p/908b48b10702

```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" th:href="@{/layui/css/layui.css}" media="all">
</head>
<body>
<div class="layui-container" style="width: 700px;height: 600px;margin-top: 0px;padding-top: 60px;">
    <div style="margin-left: 460px; width: 200px;">
        欢迎回来！&nbsp;&nbsp;&nbsp;<button class="layui-btn layui-btn-warm layui-btn-radius">退出</button></a>
    </div>

    <table class="layui-hide" id="test" lay-filter="test"></table>
    <script type="text/html" id="barDemo">
        <a class="layui-btn layui-btn-xs" lay-event="order">订购</a>
    </script>
    <script th:src="@{/layui/layui.js}" charset="utf-8"></script>
    <script>
        layui.use('table', function(){
            var table = layui.table;

            table.render({
                elem: '#test'
                ,url:'/client/findAll'
                ,title: '菜单列表'
                ,cols: [
                    [
                        {field:'id', width:100, title: '编号', sort: true}
                        ,{field:'name', width:200, title: '菜品'}
                        ,{field:'price', width:100, title: '单价'}
                        ,{field:'flavor',
width:100, title: '口味'}
                        ,
{field:'tid',width:100,  title: '分
类',templet:function(data){
                            return "热菜"
                        }
                    }
                        ,{fixed: 'right',
title:'操作', toolbar: '#barDemo', width:70}
                    ]
                ]
                ,page: true
            });

            //监听行工具事件
            table.on('tool(test)',
function(obj){
                var data = obj.data;
                if(obj.event === 'order'){

 window.location.href="/order/save/"+data.i
d;
                }
            });
        });
```

```
    </script>
</div>
</body>
</html>
```

访问：

http://localhost:8030/index.html

问题：

不能直接访问html文件，不然无法加载th

解决方法：ClientHandler.java

```
package com.ww.controller;

import com.ww.entity.Menu;
import com.ww.feign.MenuFeign;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;

import java.util.List;

```

```
@Controller
@RequestMapping("/client")
public class ClientHandler {
    @Autowired
    private MenuFeign menuFeign;
    @GetMapping("/findAll/{index}/{limit}")
    public List<Menu> findAll(@PathVariable("index") int index, @PathVariable("limit") int limit){
        return menuFeign.findAll(index,limit);
    }
    @GetMapping("/redirect/{location}")
    public String redirect(@PathVariable("location") String location){
        return location;
    }
}
```

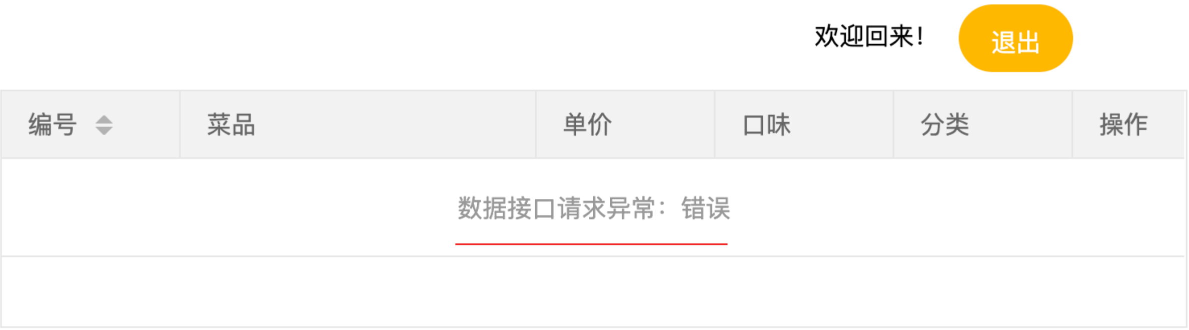
问题：

如果出现500

解决方法：

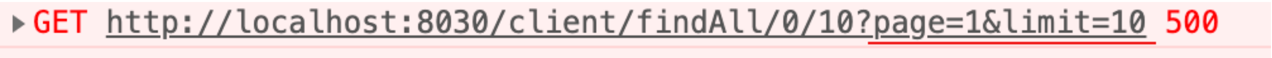
创建templates文件夹，将前端内容放到templates文件夹中

6.2.3、数据展示



编号	菜品	单价	口味	分类	操作
数据接口请求异常：错误					

传参方式：



```
GET http://localhost:8030/client/findAll/0/10?page=1&limit=10 500
```

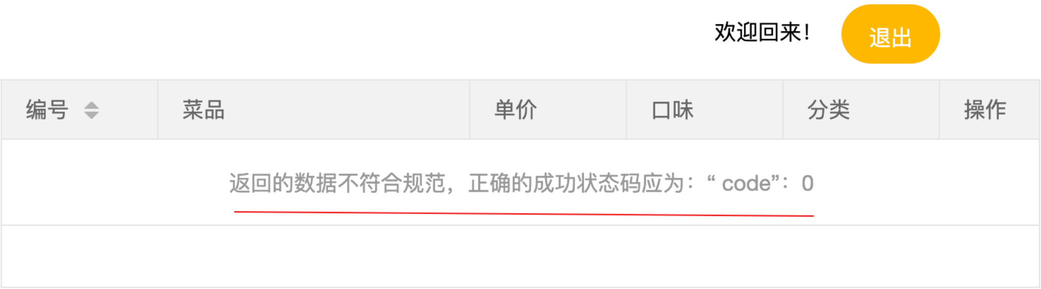
修改ClientHandler.java

```java
package com.ww.controller;

import com.ww.entity.Menu;
import com.ww.feign.MenuFeign;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;
import java.util.List;

@Controller
@RequestMapping("/client")
public class ClientHandler {
    @Autowired
    private MenuFeign menuFeign;
    @GetMapping("/findAll")
    public List<Menu>
findAll(@RequestParam("page") int page,
@RequestParam("limit") int limit){
        int index = (page-1)*limit;
        return
menuFeign.findAll(index,limit);
    }
    @GetMapping("/redirect/{location}")
    public String
redirect(@PathVariable("location") String
location){
        return location;
    }
}
```

index.html

```
url:'/client/findAll'
```

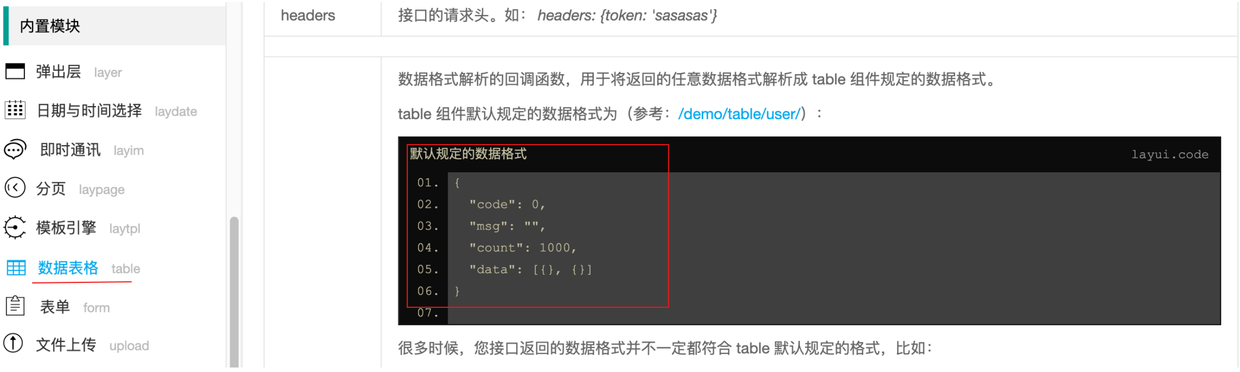
6.2.4、返回数据格式不对



编号	菜品	单价	口味	分类	操作

返回的数据不符合规范，正确的成功状态码应为："code"：0

数据正确正确返回格式：



headers	接口的请求头。如：*headers: {token: 'sasasas'}*

数据格式解析的回调函数，用于将返回的任意数据格式解析成 table 组件规定的数据格式。

table 组件默认规定的数据格式为（参考：/demo/table/user/）：

```
{
  "code": 0,
  "msg": "",
  "count": 1000,
  "data": [{}, {}]
}
```

很多时候，您接口返回的数据格式并不一定都符合 table 默认规定的格式，比如：

A、修改menu模块

MenuVO.java

```
package com.ww.entity;

import lombok.Data;
import java.util.List;

@Data
public class MenuVO {
    private int code;
    private String msg;
    private int count;
    private List<Menu> data;
}
```

修改MenuHandler.java

```
package com.ww.controller;

import com.ww.entity.MenuVO;
import com.ww.repository.MenuRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;


import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
@RequestMapping("/menu")
public class MenuHandler {
    @Value("${server.port}")
    private String port;

    @Autowired
    private MenuRepository menuRepository;

    @GetMapping("/index")
    public String index(){
        return "menu端口："+this.port;
    }

    @GetMapping("/findAll/{index}/{limit}")
    public MenuVO findAll(@PathVariable("index") int index, @PathVariable("limit") int limit){
        return new MenuVO(0,"",1000,menuRepository.findAll(index,limit));
    }
}
```

启动服务测试：

http://localhost:8020/menu/findAll/0/10

B、修改client模块

entity->MenuVO.java 拷入

修改MenuFeign.java

```
package com.ww.feign;

import com.ww.entity.MenuVO;
import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;

@FeignClient(value="menu")
public interface MenuFeign {

 @GetMapping("/menu/findAll/{index}/{limit}")
    public MenuVO findAll(@PathVariable("index") int index, @PathVariable("limit") int limit);
}
```

修改ClientHandler.java

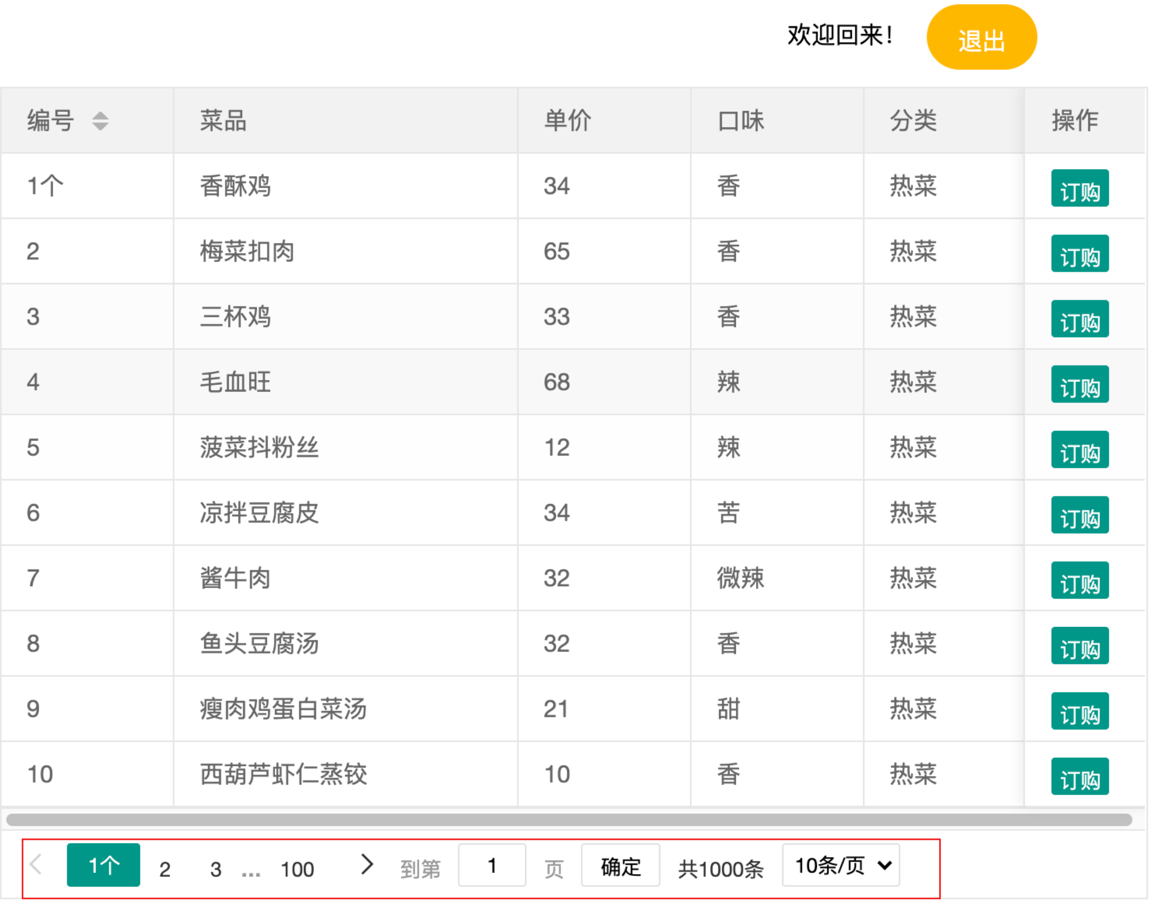
```
package com.ww.controller;

```

```
import com.ww.entity.MenuVO;
import com.ww.feign.MenuFeign;
import
org.springframework.beans.factory.annotatio
n.Autowired;
import
org.springframework.stereotype.Controller;
import
org.springframework.web.bind.annotation.*;

import java.util.List;

@Controller
@RequestMapping("/client")
public class ClientHandler {
    @Autowired
    private MenuFeign menuFeign;
    @GetMapping("/findAll")
    @ResponseBody
    public MenuVO
findAll(@RequestParam("page") int page,
@RequestParam("limit") int limit){
        int index = (page-1)*limit;
        return
menuFeign.findAll(index,limit);
```

```
    }
    @GetMapping("/redirect/{location}")
    public String
redirect(@PathVariable("location") String
location){
        return location;
    }
}
```

6.3、功能填充

6.3.1、分页问题



欢迎回来！

编号 ⬍	菜品	单价	口味	分类	操作
1个	香酥鸡	34	香	热菜	订购
2	梅菜扣肉	65	香	热菜	订购
3	三杯鸡	33	香	热菜	订购
4	毛血旺	68	辣	热菜	订购
5	菠菜抖粉丝	12	辣	热菜	订购
6	凉拌豆腐皮	34	苦	热菜	订购
7	酱牛肉	32	微辣	热菜	订购
8	鱼头豆腐汤	32	香	热菜	订购
9	瘦肉鸡蛋白菜汤	21	甜	热菜	订购
10	西葫芦虾仁蒸饺	10	香	热菜	订购

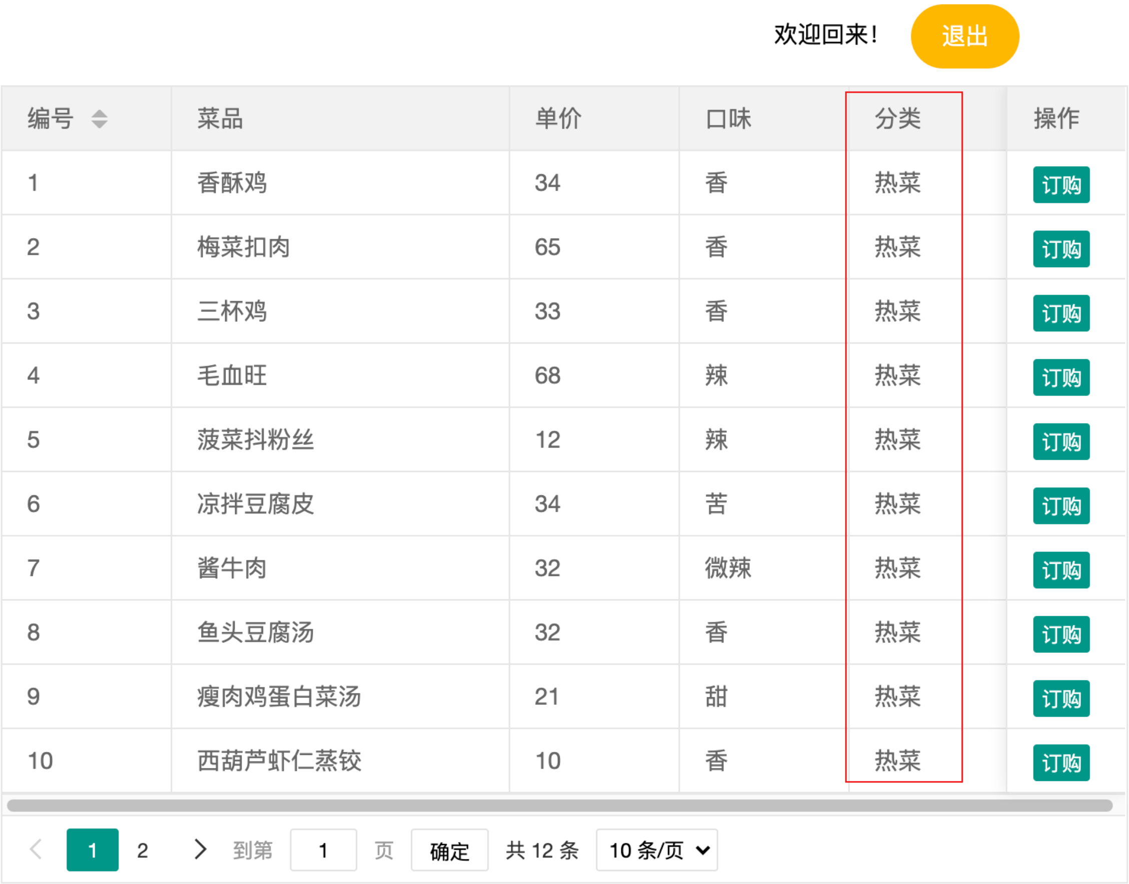
< 1个 2 3 ... 100 > 到第 1 页 确定 共1000条 10条/页

修改Menu模块中的内容

MenuHandler.java

```java
package com.ww.controller;

import com.ww.entity.MenuVO;
import com.ww.repository.MenuRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
@RequestMapping("/menu")
public class MenuHandler {
    @Value("${server.port}")
    private String port;

    @Autowired
    private MenuRepository menuRepository;

    @GetMapping("/index")
    public String index(){
        return "menu端口："+this.port;
    }

    @GetMapping("/findAll/{index}/{limit}")
    public MenuVO
findAll(@PathVariable("index") int index,
@PathVariable("limit") int limit){
        return new
MenuVO(0,"",menuRepository.count(),menuRepo
sitory.findAll(index,limit));
    }
}
```

6.3.2、菜品分类问题



编号	菜品	单价	口味	分类	操作
1	香酥鸡	34	香	热菜	订购
2	梅菜扣肉	65	香	热菜	订购
3	三杯鸡	33	香	热菜	订购
4	毛血旺	68	辣	热菜	订购
5	菠菜抖粉丝	12	辣	热菜	订购
6	凉拌豆腐皮	34	苦	热菜	订购
7	酱牛肉	32	微辣	热菜	订购
8	鱼头豆腐汤	32	香	热菜	订购
9	瘦肉鸡蛋白菜汤	21	甜	热菜	订购
10	西葫芦虾仁蒸饺	10	香	热菜	订购

A、修改Menu模块中的内容

实体类Type.java

```java
package com.ww.entity;

import lombok.Data;
@Data
public class Type {
    private long id;
    private String name;
}
```

TypeRepository.java

```java
package com.ww.repository;

import com.ww.entity.Type;

public interface TypeRepository {
    public Type findById(long id);
}
```

TypeRepository.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.ww.repository.TypeRepository">
    <select id="findById" parameterType="long" resultType="com.ww.entity.Type">
        select * from t_type where id=#{id}
    </select>
</mapper>
```

修改Menu.java

```java
package com.ww.entity;

import lombok.Data;

@Data
public class Menu {
    private long id;
    private String name;
    private double price;
    private String flavor;
    private Type type;
}
```

修改MenuRepository.xml

```
<resultMap id="menuMap" type="com.ww.entity.Menu">
        <id column="id" property="id"></id>
        <result column="name" property="name"></result>
        <result column="price" property="price"></result>
        <result column="flavor" property="flavor"></result>
        <association property="type" select="com.ww.repository.TypeRepository.findById" column="tid"></association>
    </resultMap>
    <select id="findAll" resultMap="menuMap">
        select * from t_menu limit #{param1},#{param2}
    </select>
```

分别启动服务

http://localhost:8020/menu/findAll/0/10

B、修改client模块中的内容

加入Type.java

修改Menu.java

```
package com.ww.entity;

import lombok.Data;

@Data
public class Menu {
    private long id;
    private String name;
    private double price;
    private String flavor;
    private Type type;
}
```

修改index.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet"
th:href="@{/layui/css/layui.css}"
media="all">
</head>
<body>
```

```
<div class="layui-container" style="width: 700px;height: 600px;margin-top: 0px;padding-top: 60px;">
    <div style="margin-left: 460px; width: 200px;">
        欢迎回来! &nbsp;&nbsp;&nbsp;<button class="layui-btn layui-btn-warm layui-btn-radius">退出</button></a>
    </div>

    <table class="layui-hide" id="test" lay-filter="test"></table>
    <script type="text/html" id="barDemo">
        <a class="layui-btn layui-btn-xs" lay-event="order">订购</a>
    </script>
    <script th:src="@{/layui/layui.js}" charset="utf-8"></script>
    <script>
        layui.use('table', function(){
            var table = layui.table;

            table.render({
                elem: '#test'
                ,url:'/client/findAll'
                ,title: '菜单列表'
                ,cols: [
                    [
                        {field:'id',
width:100, title: '编号', sort: true}
                        ,{field:'name',
width:200, title: '菜品'}
                        ,{field:'price',
width:100, title: '单价'}
                        ,{field:'flavor',
width:100, title: '口味'}
                        ,
{field:'tid',width:100,  title: '分
类',templet:function(data){
                            return
data.type.name
                        }
                    }
                        ,{fixed: 'right',
title:'操作', toolbar: '#barDemo', width:70}
                    ]
                ]
                ,page: true
            });

            //监听行工具事件
            table.on('tool(test)',
function(obj){
                var data = obj.data;
                if(obj.event === 'order'){

 window.location.href="/order/save/"+data.i
d;
                }
            });
        });
    </script>
</div>
</body>
</html>
```

分别启动服务

http://localhost:8030/client/redirect/index

6.3.3、操作问题

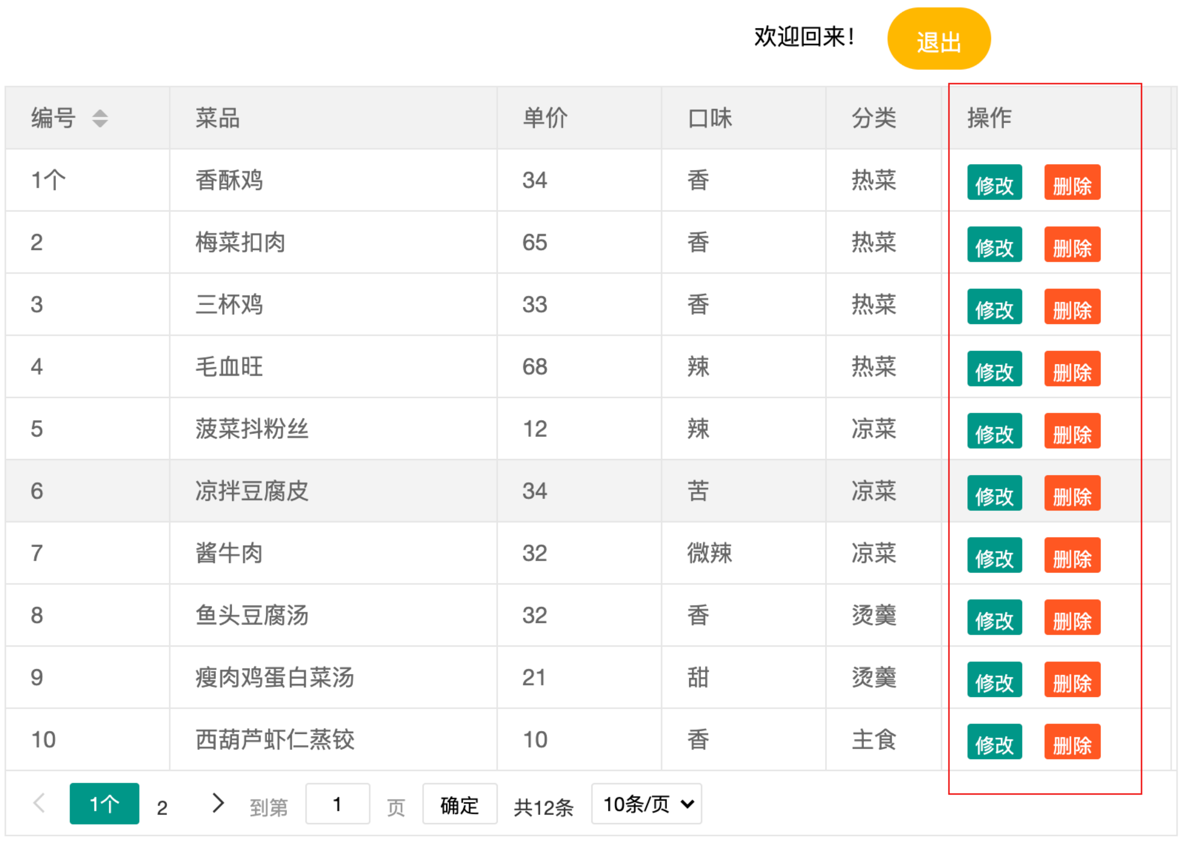
A、修改操作菜品的页面展示

调整页面index.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" th:href="@{/layui/css/layui.css}" media="all">
</head>
<body>
<div class="layui-container" style="width: 750px;height: 600px;margin-top: 0px;padding-top: 60px;">
    <div style="margin-left: 460px; width: 200px;">
        欢迎回来! &nbsp;&nbsp;&nbsp;<button class="layui-btn layui-btn-warm layui-btn-radius">退出</button></a>
    </div>

    <table class="layui-hide" id="test" lay-filter="test"></table>
    <script type="text/html" id="barDemo">
        <a class="layui-btn layui-btn-xs" lay-event="update">修改</a>
        <a class="layui-btn layui-btn-danger layui-btn-xs" lay-event="del">删除</a>
    </script>
    <script th:src="@{/layui/layui.js}"
charset="utf-8"></script>
    <script>
        layui.use('table', function(){
            var table = layui.table;

            table.render({
                elem: '#test'
                ,url:'/client/findAll'
                ,title: '菜单列表'
                ,cols: [
                    [
                        {field:'id',
width:100, title: '编号', sort: true}
                        ,{field:'name',
width:200, title: '菜品'}
                        ,{field:'price',
width:100, title: '单价'}
                        ,{field:'flavor',
width:100, title: '口味'}
                        ,
{field:'tid',width:70,  title: '分
类',templet:function(data){
                            return
data.type.name
```

```
                        }
                    }
                        ,{fixed: 'right',
title:'操作', toolbar: '#barDemo',
width:140}
                    ]
                ]
                ,page: true
            });

            //监听行工具事件
            table.on('tool(test)',
function(obj){
                var data = obj.data;
                if(obj.event === 'order'){

 window.location.href="/order/save/"+data.i
d;
                }
            });
        });
    </script>
</div>
</body>
</html>
```



欢迎回来！ 退出

编号	菜品	单价	口味	分类	操作
1个	香酥鸡	34	香	热菜	修改 删除
2	梅菜扣肉	65	香	热菜	修改 删除
3	三杯鸡	33	香	热菜	修改 删除
4	毛血旺	68	辣	热菜	修改 删除
5	菠菜抖粉丝	12	辣	凉菜	修改 删除
6	凉拌豆腐皮	34	苦	凉菜	修改 删除
7	酱牛肉	32	微辣	凉菜	修改 删除
8	鱼头豆腐汤	32	香	烫羹	修改 删除
9	瘦肉鸡蛋白菜汤	21	甜	烫羹	修改 删除
10	西葫芦虾仁蒸饺	10	香	主食	修改 删除

< 1个 2 > 到第 1 页 确定 共12条 10条/页

B、完成修改与删除的事件监听

修改index.html

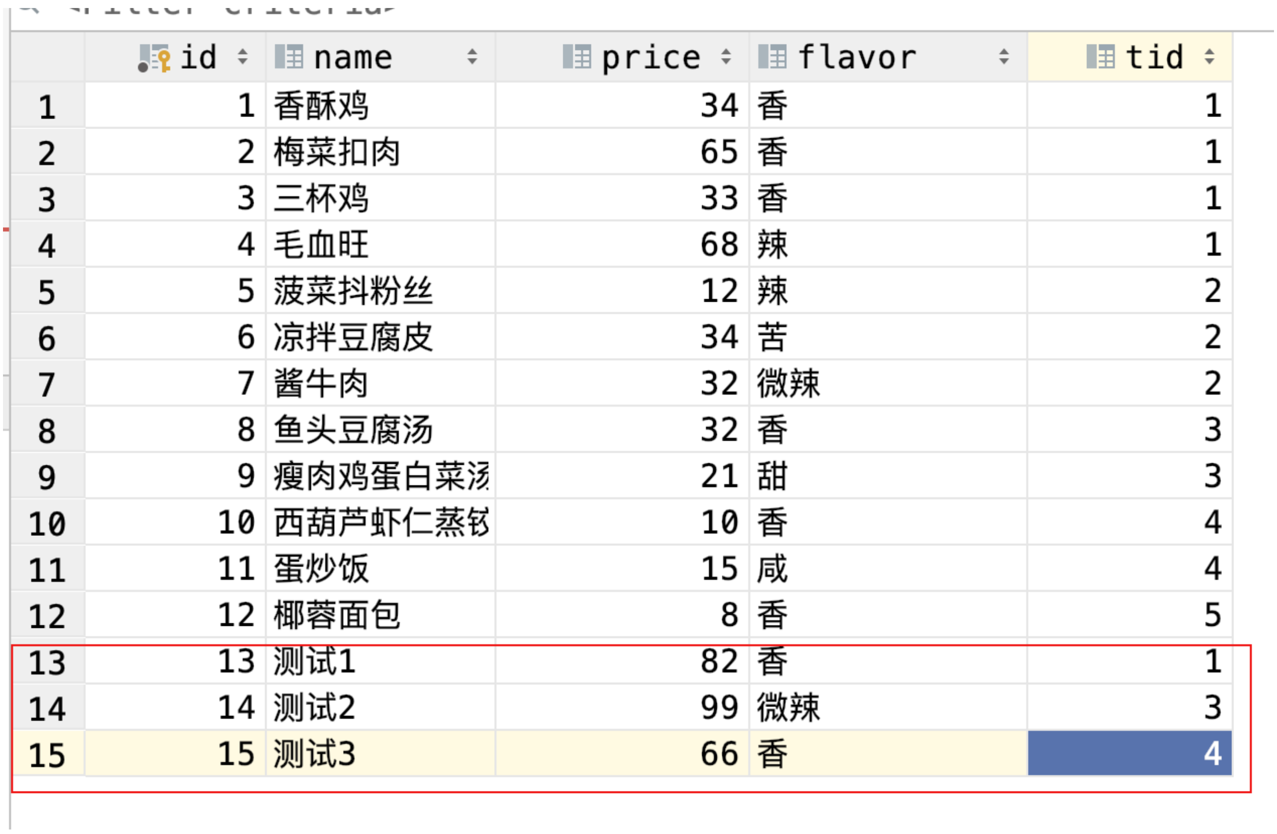
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet"
th:href="@{/layui/css/layui.css}"
media="all">
</head>
<body>
```

```
<div class="layui-container" style="width: 750px;height: 600px;margin-top: 0px;padding-top: 60px;">
    <div style="margin-left: 460px; width: 200px;">
        欢迎回来！&nbsp;&nbsp;&nbsp;<button class="layui-btn layui-btn-warm layui-btn-radius">退出</button></a>
    </div>

    <table class="layui-hide" id="test" lay-filter="test"></table>
    <script type="text/html" id="barDemo">
        <a class="layui-btn layui-btn-xs" lay-event="update">修改</a>
        <a class="layui-btn layui-btn-danger layui-btn-xs" lay-event="del">删除</a>
    </script>
    <script th:src="@{/layui/layui.js}" charset="utf-8"></script>
    <script>
        layui.use('table', function(){
            var table = layui.table;

            table.render({
                elem: '#test'
                ,url:'/client/findAll'
                ,title: '菜单列表'
                ,cols: [
                    [
                        {field:'id',
width:100, title: '编号', sort: true}
                        ,{field:'name',
width:200, title: '菜品'}
                        ,{field:'price',
width:100, title: '单价'}
                        ,{field:'flavor',
width:100, title: '口味'}
                        ,
{field:'tid',width:70,  title: '分
类',templet:function(data){
                            return
data.type.name
                        }
                    }
                        ,{fixed: 'right',
title:'操作', toolbar: '#barDemo',
width:140}
                    ]
                ]
                ,page: true
```

```
            });

            //监听行工具事件
            table.on('tool(test)',
function(obj){
                var data = obj.data;
                if(obj.event === 'update'){
                    layer.alert("确认修改数
据？ ");
                }
                if(obj.event === 'del'){
                    layer.confirm("你是否确认
删除数据？ ",function(index){

 window.location.href="/order/save/"+data.i
d;
                        layer.close(index);
                    });
                }
            });
        });
    </script>
</div>
</body>
</html>
```

C、完成删除操作

加入测试数据



	id	name	price	flavor	tid
1	1	香酥鸡	34	香	1
2	2	梅菜扣肉	65	香	1
3	3	三杯鸡	33	香	1
4	4	毛血旺	68	辣	1
5	5	菠菜抖粉丝	12	辣	2
6	6	凉拌豆腐皮	34	苦	2
7	7	酱牛肉	32	微辣	2
8	8	鱼头豆腐汤	32	香	3
9	9	瘦肉鸡蛋白菜汤	21	甜	3
10	10	西葫芦虾仁蒸饺	10	香	4
11	11	蛋炒饭	15	咸	4
12	12	椰蓉面包	8	香	5
13	13	测试1	82	香	1
14	14	测试2	99	微辣	3
15	15	测试3	66	香	4

修改menu模块里的内容

MenuHandler.java

```
package com.ww.controller;


import com.ww.entity.MenuVO;
import com.ww.repository.MenuRepository;
import
org.apache.ibatis.annotations.Delete;
```

```java
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/menu")
public class MenuHandler {
    @Value("${server.port}")
    private String port;

    @Autowired
    private MenuRepository menuRepository;

    @GetMapping("/index")
    public String index(){
        return "menu端口："+this.port;
    }

    @GetMapping("/findAll/{index}/{limit}")
    public MenuVO findAll(@PathVariable("index") int index, @PathVariable("limit") int limit){
        return new MenuVO(0,"",menuRepository.count(),menuRepository.findAll(index,limit));
    }
    @DeleteMapping("/deleteById/{id}")
    public void deleteById(@PathVariable("id") long id){
        menuRepository.delete(id);
    }
}
```

修改client模块里的内容

MenuFeign.java

```
package com.ww.feign;

import com.ww.entity.MenuVO;
import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;

@FeignClient(value="menu")
public interface MenuFeign {

 @GetMapping("/menu/findAll/{index}/{limit}")
    public MenuVO findAll(@PathVariable("index") int index, @PathVariable("limit") int limit);
    @DeleteMapping("/menu/deleteById/{id}")
    public void deleteById(@PathVariable("id") long id);
}
```

ClientHandler.java

```
package com.ww.controller;

import com.ww.entity.MenuVO;
import com.ww.feign.MenuFeign;

```

```
import
org.springframework.beans.factory.annotatio
n.Autowired;
import
org.springframework.stereotype.Controller;
import
org.springframework.web.bind.annotation.*;

import java.util.List;

@Controller
@RequestMapping("/client")
public class ClientHandler {
    @Autowired
    private MenuFeign menuFeign;
    @GetMapping("/findAll")
    @ResponseBody
    public MenuVO
findAll(@RequestParam("page") int page,
@RequestParam("limit") int limit){
        int index = (page-1)*limit;
        return
menuFeign.findAll(index,limit);
    }
    @GetMapping("/redirect/{location}")
    public String redirect(@PathVariable("location") String location){
        return location;
    }
    @GetMapping("/deleteById/{id}")
    public String deleteById(@PathVariable("id") long id){
        menuFeign.deleteById(id);
        return "redirect:/client/redirect/index";
    }
}
```

修改index.html

```
//监听行工具事件
table.on('tool(test)', function(obj){
                var data = obj.data;
                if(obj.event === 'update'){
                    layer.alert("确认修改数据？");
                }
                if(obj.event === 'del'){
                    layer.confirm("你是否确认删除数据？",function(index){
                        window.location.href="/client/deleteById/"+data.id;
                        layer.close(index);
                    });
                }
            });
```

分别启动服务

http://localhost:8030/client/redirect/index

D、完成增加操作

页面展示部分

加入页面menu_add.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" th:href="@{/layui/css/layui.css}" media="all">
</head>
<body>
<div class="layui-container" style="margin-top: 50px;">

    <div class="layui-container" style="width: 500px;height: 330px;padding-top: 60px;">
        <form class="layui-form" action="/menu/save" method="post">
            <div class="layui-form-item">
                <label class="layui-form-label">菜品：</label>
                <div class="layui-inline">
```

```
                    <input type="text"
name="name" lay-verify="required"
autocomplete="off" placeholder="请输入用户名"
class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-
label">单价：</label>
                <div class="layui-inline">
                    <input type="text"
name="price" lay-verify="required"
placeholder="请输入密码" autocomplete="off"
class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-
label">口味：</label>
                <div class="layui-inline">
                    <input type="text"
name="flavor" lay-verify="required"
placeholder="请输入年龄" autocomplete="off"
class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">分类：</label>
                <div class="layui-input-inline">
                    <select name="type.id">
                        <option th:each="type:${list}" th:text="${type.name}" th:value="${type.id}"></option>
                    </select>
                </div>
            </div>
            <div class="layui-form-item">
                <button class="layui-btn" lay-submit="" lay-filter="demo2" style="margin-left: 160px;">提交</button>
            </div>
        </form>
    </div>
</div>
<script th:src="@{/layui/layui.js}" charset="utf-8"></script>
<script>
    layui.use(['form', 'element'], function(){
        var form = layui.form;
        var element = layui.element;

        //自定义验证规则
        form.verify({
            price: [/(^[1-9]([0-9]+)?(\.[0-9]{1,2})?$)|(^(0){1}$)|(^[0-9]\.[0-9]([0-9])?$)/, "请输入正确的价格"]
        });

    });
</script>
</body>
</html>
```

修改menu模块里的内容

修改TypeRepository.java

```java
package com.ww.repository;

import com.ww.entity.Type;

import java.util.List;

public interface TypeRepository {
    public Type findById(long id);
    public List<Type> findTypes();
}
```

修改TypeRepository.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.ww.repository.TypeRepository">
    <select id="findById" parameterType="long" resultType="com.ww.entity.Type">
        select * from t_type where id=#{id}
    </select>
    <select id="findTypes" resultType="com.ww.entity.Type">
        select * from t_type
    </select>
</mapper>
```

修改MenuHandler.java

```java
package com.ww.controller;

import com.ww.entity.MenuVO;
import com.ww.entity.Type;
import com.ww.repository.MenuRepository;
import com.ww.repository.TypeRepository;
import
org.apache.ibatis.annotations.Delete;
import
org.springframework.beans.factory.annotation.Autowired;
import
org.springframework.beans.factory.annotation.Value;
import
org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/menu")
public class MenuHandler {
    @Value("${server.port}")
    private String port;

    @Autowired
    private MenuRepository menuRepository;

    @Autowired
    private TypeRepository typeRepository;

    @GetMapping("/index")
    public String index(){
        return "menu端口："+this.port;
    }

    @GetMapping("/findAll/{index}/{limit}")
    public MenuVO
findAll(@PathVariable("index") int index,
@PathVariable("limit") int limit){
        return new
MenuVO(0,"",menuRepository.count(),menuRepo
sitory.findAll(index,limit));
    }
    @DeleteMapping("/deleteById/{id}")
    public void
deleteById(@PathVariable("id") long id){
        menuRepository.delete(id);
    }
    @GetMapping("/findTypes")
    public List<Type> findTypes(){
        return typeRepository.findTypes();
    }
}
```

修改client模块里的内容

将OrderHandler.java改名为MenuHandler.java

注意：url地址order都改成menu

修改MenuFeign.java

```java
package com.ww.feign;

import com.ww.entity.MenuVO;
import com.ww.entity.Type;
import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;

import java.util.List;

@FeignClient(value="menu")
public interface MenuFeign {
 @GetMapping("/menu/findAll/{index}/{limit}")
    public MenuVO findAll(@PathVariable("index") int index, @PathVariable("limit") int limit);
    @DeleteMapping("/menu/deleteById/{id}")
    public void deleteById(@PathVariable("id") long id);
    @GetMapping("/menu/findTypes")
    public List<Type> findTypes();
}
```

修改MenuHandler.java

```
package com.ww.controller;

import com.ww.entity.MenuVO;
import com.ww.entity.Type;
import com.ww.feign.MenuFeign;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;
```

```java
import org.springframework.web.servlet.ModelAndView;

import java.util.List;

@Controller
@RequestMapping("/menu")
public class MenuHandler {
    @Autowired
    private MenuFeign menuFeign;
    @GetMapping("/findAll")
    @ResponseBody
    public MenuVO findAll(@RequestParam("page") int page, @RequestParam("limit") int limit){
        int index = (page-1)*limit;
        return menuFeign.findAll(index,limit);
    }
    @GetMapping("/redirect/{location}")
    public String redirect(@PathVariable("location") String location){
        return location;
    }
    @GetMapping("/deleteById/{id}")
    public String
deleteById(@PathVariable("id") long id){
        menuFeign.deleteById(id);
        return
"redirect:/menu/redirect/index";
    }
    @GetMapping("/findTypes")
    public ModelAndView findTypes(){
        List<Type>
list=menuFeign.findTypes();
        ModelAndView modelAndView=new
ModelAndView();

 modelAndView.setViewName("menu_add");

 modelAndView.addObject("list",list);
        return modelAndView;
    }
}
```

数据增加部份

修改menu模块里的内容

修改MenuRepository.xml

```
<insert id="save"
parameterType="com.ww.entity.Menu">
        insert into
t_menu(name,price,flavor,tid)values(#
{name},#{price},#{flavor},#{type.id})
</insert>
```

修改MenuHandler.java

```
@PostMapping("/save")
public void save(@RequestBody Menu menu){
        menuRepository.save(menu);
}
```

修改client模块里的内容

修改MenuFeign.java

```
  @PostMapping("/menu/save")
    public void save(@RequestBody Menu
menu);
```

修改MenuHandler.java

```
@PostMapping("/save")
public String save(Menu menu){
        menuFeign.save(menu);
        return
"redirect:/menu/redirect/index";
}
```

分别启动服务

http://localhost:8030/menu/findTypes

E、完成修改操作

修改menu模块里的内容

修改MenuRepository.xml

```
<select id="findById" parameterType="long"
resultMap="menuMap">
        select * from t_menu where id=#{id}
</select>
<update id="update"
parameterType="com.ww.entity.Menu">
        update t_menu set name=#
{name},price=#{price},flavor=#
{flavor},tid=#{type.id} where id=#{id}
</update>
```

修改MenuHandler.java

```
@GetMapping("/findById/{id}")
public Menu findById(@PathVariable("id") long id){
        return menuRepository.findById(id);
}
@PostMapping("/update")
public void update(@RequestBody Menu menu){
        menuRepository.update(menu);
}
```

修改client模块里的内容

修改MenuFeign.java

```
@GetMapping("/menu/findById/{id}")
public Menu findById(@PathVariable("id") long id);
@PostMapping("/menu/update")
public void update(@RequestBody Menu menu);
```

修改MenuHandler.java

```
 @GetMapping("/findById/{id}")
    public ModelAndView
findById(@PathVariable("id") long id){
        Menu menu= menuFeign.findById(id);
        ModelAndView modelAndView=new
ModelAndView();

 modelAndView.addObject("menu",menu);

 modelAndView.setViewName("menu_add");
        return modelAndView;
    }
    @PostMapping("/update")
    public String update(Menu menu){
        menuFeign.update(menu);
        return
"redirect:/menu/redirect/index";
    }
```

7、服务提供者-user

7.1、pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project
xmlns="http://maven.apache.org/POM/4.0.0"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"

xsi:schemaLocation="http://maven.apache.org
/POM/4.0.0
http://maven.apache.org/xsd/maven-
4.0.0.xsd">
    <parent>

 <artifactId>aispringclouddemo</artifactId>
        <groupId>com.ww</groupId>
        <version>1.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <artifactId>user</artifactId>
    <dependencies>
        <dependency>

 <groupId>org.springframework.cloud</groupI
d>
            <artifactId>spring-cloud-
starter-netflix-eureka-client</artifactId>

 <version>2.0.2.RELEASE</version>
```

```
        </dependency>
        <dependency>
            <groupId>org.mybatis.spring.boot</groupId>
            <artifactId>mybatis-spring-boot-starter</artifactId>
            <version>1.3.1</version>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.11</version>
        </dependency>

        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-config</artifactId>
            <version>2.0.2.RELEASE</version>
        </dependency>
    </dependencies>
```

```
</project>
```

7.2、user-dev.yml

Configserver->shared->user-dev.yml

```
server:
  port: 8040
spring:
  application:
    name: user
  datasource:
    name: ordersystem
    url:
jdbc:mysql://localhost:3306/ordersystem
    username: root
    password: Root123456
mybatis:
  mapper-locations:
classpath:/mapping/*.xml
  type-aliases-package: com.ww.entity
```

7.3、bootstrap.yml

```yaml
spring:
  application:
    name: user
  profiles:
    active: dev
  cloud:
    config:
      uri: http://localhost:8762
      fail-fast: true
```

7.4、UserApplication.java

```java
package com.ww;

import org.mybatis.spring.annotation.MapperScan;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
@MapperScan("com.ww.repository")
public class UserApplication {
    public static void main(String[] args) {

        SpringApplication.run(UserApplication.class,args);
    }
}
```

7.5、entity

User.java

```
package com.ww.entity;

import lombok.Data;

import java.util.Date;

@Data
public class User {
 private long id;
 private String username;
 private String password;
 private String nickname;
 private String gender;
 private String telephone;

```

```java
 private Date registerdate;
 private String address;
}
```

UserVO.java

```java
package com.ww.entity;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.util.List;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class UserVO {
    private int code;
    private String msg;
    private int count;
    private List<User> data;

}
```

7.6、repository

UserRepository.java

```java
package com.ww.repository;

import com.ww.entity.User;

import java.util.List;

public interface UserRepository {
    public List<User> findAll(int index, int limit);
    public int count();
    public void save(User user);
    public void update(User user);
    public void delete(long id);
    public User findById(long id);
}
```

UserRepostory.xml

mapping->UserRepostory.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.ww.repository.UserRepository">

    <select id="findAll" resultType="User">
        select * from t_user limit #{param1},#{param2}
    </select>
    <select id="count" resultType="int">
        select count(id) from t_user
    </select>

    <select id="findById" parameterType="long" resultType="User">
        select * from t_user where id=#{id}
    </select>
    <insert id="save" parameterType="User">
```

```
        insert into t_user(username,password,nickname,gender,telephone,registerdate,address)values(#{username},#{password},#{nickname},#{gender},#{telephone},#{registerdate},#{address})
    </insert>
    <update id="update" parameterType="User">
        update t_user set username=#{username},password=#{password},nickname=#{nickname},gender=#{gender},telephone=#{telephone},registerdate=#{registerdate},address=#{address} where id=#{id}
    </update>
    <delete id="delete" parameterType="long">
        delete from t_user where id=#{id}
    </delete>
</mapper>
```

7.7、controller

UserHandler.java

```
package com.ww.controller;

import com.ww.entity.User;
import com.ww.entity.UserVO;
import com.ww.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("/user")
public class UserHandler {
    @Autowired
    private UserRepository userRepository;

    @GetMapping("/findAll/{index}/{limit}")
    public UserVO findAll(@PathVariable("index") int index,@PathVariable("limit")int limit){
        return new UserVO(0,"",userRepository.count(),userRepository.findAll(index,limit));
    }
    @DeleteMapping("/deleteById/{id}")
    public void
deleteById(@PathVariable("id") long id){
        userRepository.delete(id);
    }
    @PostMapping("/save")
    public void save(@RequestBody User
user){
        userRepository.save(user);
    }
    @GetMapping("/findById/{id}")
    public User
findById(@PathVariable("id") long id){
        return userRepository.findById(id);
    }
    @PostMapping("/update")
    public void update(@RequestBody User
user){
        userRepository.update(user);
    }
    @GetMapping("/count")
    public int count()
{userRepository.count()};
}
```

7.8、分别启动服务

http://localhost:8040/user/findById/1

http://localhost:8040/user/findAll/0/10

8、服务消费者client->user

8.1、复制实体类

将User.java/UserVO.java复制到client模块中有entity包中

8.2、UserFeign.java

```
package com.ww.feign;

import com.ww.entity.User;
import com.ww.entity.UserVO;
import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.web.bind.annotation.*;

@FeignClient(value="user")
```

```
public interface UserFeign {

 @GetMapping("/user/findAll/{index}/{limit}
")
    public UserVO
findAll(@PathVariable("index") int
index,@PathVariable("limit") int limit);
    @DeleteMapping("/user/deleteById/{id}")
    public void delete(@PathVariable("id")
long id);
    @PostMapping("/user/save")
    public void save(@RequestBody User
user);
    @GetMapping("/user/findById/{id}")
    public User
findById(@PathVariable("id") long id);
    @PostMapping("/user/update")
    public void update(@RequestBody User
user);
    @GetMapping("/user/count")
    public int count();

}
```

8.3、UserHandler.java

```java
package com.ww.controller;

import com.ww.entity.MenuVO;
import com.ww.entity.User;
import com.ww.entity.UserVO;
import com.ww.feign.UserFeign;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;

@Controller
@RequestMapping("/user")
public class UserHandler {
    @Autowired
    private UserFeign userFeign;

    @GetMapping("/findAll")
    @ResponseBody
    public UserVO
findAll(@RequestParam("page") int page,
@RequestParam("limit") int limit){
        int index = (page-1)*limit;
        return
userFeign.findAll(index,limit);
    }
    @DeleteMapping("/deleteById/{id}")
    @ResponseBody
    public void
deleteById(@PathVariable("id") long id){
        userFeign.delete(id);
    }
    @PostMapping("/save")
    @ResponseBody
    public void save(@RequestBody User
user){
        userFeign.save(user);
    }
    @GetMapping("/findById/{id}")
    @ResponseBody
    public User
findById(@PathVariable("id") long id){
        return userFeign.findById(id);
    }
    @PostMapping("/update")
    @ResponseBody
    public void update(@RequestBody User
user){
        userFeign.update(user);
    }
    @GetMapping("/count")
    @ResponseBody
    public int count(){return
userFeign.count();}

}
```

8.4、分别启动服务

postman完成接口测试

http://localhost:8030/user/findAll?page=1&limit=10 get

http://localhost:8030/user/findById/1 get

http://localhost:8030/user/count get

http://localhost:8030/user/save post

http://localhost:8030/user/update post

http://localhost:8030/user/deleteById/7 get

9、服务消费者中user操作

9.1、前端页面

user_add.html

```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" th:href="@{/layui/css/layui.css}" media="all">
</head>
<body>
<div class="layui-container" style="margin-top: 50px;">

    <div class="layui-container" style="width: 500px;height: 330px;padding-top: 60px;">
        <form class="layui-form" action="/user/save" method="post">
            <div class="layui-form-item">
```

```
                <label class="layui-form-label">用户名：</label>
                <div class="layui-inline">
                    <input type="text" name="username" lay-verify="username" autocomplete="off" placeholder="请输入用户名" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">密码：</label>
                <div class="layui-inline">
                    <input type="password" name="password" lay-verify="password" placeholder="请输入密码" autocomplete="off" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">昵称：</label>
                <div class="layui-inline">
```

```
                    <input type="text" name="nickname" lay-verify="required" placeholder="请输入昵称" autocomplete="off" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">性别：</label>
                <div class="layui-input-inline">
                    <input type="radio" name="gender" value="男" title="男" checked="">
                    <input type="radio" name="gender" value="女" title="女">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">联系电话：</label>
                <div class="layui-inline">
                    <input type="text" name="telephone" lay-verify="telephone" placeholder="请输入联系电话" autocomplete="off" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">地址：</label>
                <div class="layui-inline">
                    <input type="text" name="address" lay-verify="required" placeholder="请输入地址" autocomplete="off" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <button class="layui-btn" lay-submit="" style="margin-left: 160px;">提交</button>
            </div>
        </form>
    </div>
</div>
<script th:src="@{/layui/layui.js}" charset="utf-8"></script>
<script>
    layui.use(['form', 'element'], function(){
        var form = layui.form;
        var element = layui.element;

        //自定义验证规则
        form.verify({
            username: function(value){
                if(value.length == 0){
                    return "用户名不能为空";
                }
            },
            password: [/^[A-Za-z0-9]+$/,"密码必须由数字和字母组成"],
            telephone: [/^(0|86|17951)?(13[0-9]|15[012356789]|17[678]|18[0-9]|14[57])[0-9]{8}$/,"请输入正确的电话"]
        });

    });
</script>
</body>
</html>
```

user_manager.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" th:href="@{/layui/css/layui.css}" media="all">
</head>
<body class="layui-layout-body">
<div class="layui-container" style="margin-top: 50px;width: 1000px;">

    <table class="layui-hide" id="test" style="width: 300px" lay-filter="test"></table>
    <script type="text/html" id="barDemo">
        <a class="layui-btn layui-btn-danger layui-btn-xs" lay-event="del">删除</a>
    </script>
    <script th:src="@{/layui/layui.js}" charset="utf-8"></script>
    <script>
        layui.use('table', function(){
            var table = layui.table;

            table.render({
                elem: '#test'
                ,url:'/user/findAll'
                ,title: '用户列表'
                ,cols: [
                    [
                        {field:'id',
width:70, title: '编号', sort: true}
                        ,{field:'username',
width:100, title: '用户名'}
                        ,{field:'nickname',
width:100, title: '昵称'}
                        ,{field:'gender',
width:70, title: '性别'}
                        ,
{field:'telephone', width:160, title: '联系
电话'}
                        ,
{field:'registerdate', width:220, title:
'注册日期'}
                        ,{field:'address',
width:200, title: '地址'}
                        ,{fixed: 'right',
title:'操作', toolbar: '#barDemo', width:80}
                ]
                ]
                ,page: true
            });
            //监听行工具事件
            table.on('tool(test)',
function(obj){
                var data = obj.data;
                if(obj.event === 'del'){
                    layer.confirm('确定要删除
吗？', function(index){

 window.location.href="/user/deleteById/"+d
ata.id;
                        layer.close(index);
                    });
                }
            });
        });
    </script>

</div>
</body>
</html>
```

9.2、查询用户

修改UserHandler.java

加入页面转向的方法

```
@GetMapping("/redirect/{location}")
    public String
redirect(@PathVariable("location") String
location){
        return location;
    }
```

分别启动服务

http://localhost:8030/user/redirect/user_manage

9.3、删除用户

修改UserHandler.java

```
@GetMapping("/deleteById/{id}")
    public String
deleteById(@PathVariable("id") long id){
        userFeign.delete(id);
        return
"redirect:/user/redirect/user_manage";
    }
```

分别启动服务

http://localhost:8030/user/redirect/user_manage

9.4、增加用户

修改UserHandler.java

```
 @PostMapping("/save")
    public String save(User user){
        user.setRegisterdate(new Date());
        userFeign.save(user);
        return
"redirect:/user/redirect/user_manage";
    }
```

分别启动服务

http://localhost:8030/user/redirect/user_add

10、服务提供者-account

10.1、pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project
xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"
```

```
xsi:schemaLocation="http://maven.apache.org
/POM/4.0.0
http://maven.apache.org/xsd/maven-
4.0.0.xsd">
    <parent>

 <artifactId>aispringclouddemo</artifactId>
        <groupId>com.ww</groupId>
        <version>1.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <artifactId>account</artifactId>
    <dependencies>
        <dependency>

 <groupId>org.springframework.cloud</groupI
d>
            <artifactId>spring-cloud-
starter-netflix-eureka-client</artifactId>

 <version>2.0.2.RELEASE</version>
        </dependency>
        <dependency>
 <groupId>org.mybatis.spring.boot</groupId>
            <artifactId>mybatis-spring-boot-starter</artifactId>
            <version>1.3.1</version>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.11</version>
        </dependency>

        <dependency>

 <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-config</artifactId>

 <version>2.0.2.RELEASE</version>
        </dependency>
    </dependencies>

</project>
```

10.2、account-dev.yml

```
server:
  port: 8050
spring:
  application:
    name: account
  datasource:
    name: ordersystem
    url:
jdbc:mysql://localhost:3306/ordersystem
    username: root
    password: Root123456
mybatis:
  mapper-locations:
classpath:/mapping/*.xml
  type-aliases-package: com.ww.entity
```

10.3、bootstrap.yml

```yaml
spring:
  application:
    name: account
  profiles:
    active: dev
  cloud:
    config:
      uri: http://localhost:8762
      fail-fast: true
```

10.4、AccountApplication.java

```java
package com.ww;

import org.mybatis.spring.annotation.MapperScan;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
@MapperScan("com.ww.repository")
public class AccountApplication {
    public static void main(String[] args) {
        SpringApplication.run(AccountApplication.class,args);
    }
}
```

10.5、entity

复制User到account->entity

Admin.java

```
package com.ww.entity;

import lombok.Data;

@Data
public class Admin {
    private long id;
    private String username;
    private String password;
}
```

10.6、repository

UserRepository.java

```java
package com.ww.repository;

import com.ww.entity.User;

public interface UserRepository {
    public User login(String username, String password);
}
```

UserRepository.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-
//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-
mapper.dtd">
<mapper
namespace="com.ww.repository.UserRepository
">
    <select id="login" resultType="User">
        select * from t_user where
username=#{param1} and password=#{param2}
    </select>
</mapper>
```

AdminRepository.java

```
package com.ww.repository;

import com.ww.entity.Admin;

public interface AdminRepository {
    public Admin login(String username,
String password);
}
```

AdminRepository.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.ww.repository.AdminRepository">
    <select id="login" resultType="Admin">
        select * from t_admin where username=#{param1} and password=#{param2}
    </select>
</mapper>
```

10.7、controller

AccountHandler.java

```java
package com.ww.controller;

import com.ww.repository.AdminRepository;
import com.ww.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/account")
public class AccountController {
    @Autowired
    private UserRepository userRepository;
    @Autowired
    private AdminRepository adminRepository;

 @GetMapping("/login/{username}/{password}/{type}")
```

```
    public Object login(@PathVariable("username") String username,@PathVariable("password") String password,@PathVariable("type") String type){
        Object object=null;
        switch (type){
            case "user":object=userRepository.login(username,password);break;
            case "admin":object=adminRepository.login(username,password);break;
        }
        return object;
    }
}
```

10.8、分别启动服务

http://localhost:8050/account/login/admin/admin/admin

http://localhost:8050/account/login/zhangsan/zhangsan/user

11、服务消费者client->user/admin

11.1、展示页面

login.html

```
```

11.2、entity

复制Admin到client->entity

11.3、feign

```
```

11.4、完善后台管理页面操作

11.4.1、页面调整

index.html改为menu_manage.html

在复制一个index.html

11.4.2、MenuHandler.java调整

URL：index 改为menu_manage

11.4.3、完善增加、退出操作

修改menu_manage.html

```
<div style="margin-left: 460px; width: 200px;">
        欢迎回来！<a href="" th:text="${session.admin.username}"></a>
        <a href="/account/logout"><button class="layui-btn layui-btn-warm layui-btn-radius">退出</button></a>
        <br/>
        <a href="/menu/findTypes"><button class="layui-btn layui-btn-warm layui-btn-radius">增加</button></a>
</div>
```

index.html与menu_manager.html的退出一样，所有都是调用AccountHandler->logout方法

11.5、controller

AccountHandler.java

```java
package com.ww.controller;

import com.ww.entity.Admin;
import com.ww.entity.User;
import com.ww.feign.AccountFeign;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;

import javax.servlet.http.HttpSession;
import java.util.LinkedHashMap;

@Controller
@RequestMapping("/account")
public class AccountHandler {
    @Autowired
    private AccountFeign accountFeign;
    @PostMapping("/login")
    public String
login(@RequestParam("username") String
username, @RequestParam("password") String
password, @RequestParam("type") String
type, HttpSession session){
        Object
object=accountFeign.login(username,password
,type);
        LinkedHashMap<String,Object>
linkedHashMap=(LinkedHashMap<String,
Object>) object;
        String result="";
        if(object==null){
            result="login";
        }else{
            switch (type){
                case "user":
                    long id=
(Integer)linkedHashMap.get("id");
                    String nickname=
(String)linkedHashMap.get("nickname");
                    User user=new User();
                    user.setId(id);

 user.setNickname(nickname);
 session.setAttribute("user",user);
                    result="index";
                    break;
                case "admin":
                    Admin admin=new
Admin();
                    long aid=
(Integer)linkedHashMap.get("id");
                    String ausername=
(String)linkedHashMap.get("username");
                    admin.setId(aid);

 admin.setUsername(ausername);

 session.setAttribute("admin",admin);
                    result="menu_manage";
                    break;
            }
        }

        return result;
    }
    @GetMapping("/logout")
    public String logout(HttpSession
session){
        session.invalidate();
        return "login";
    }
}
```

12、服务提供者-order

12.1、entity

复制User、Admin、Menu、Type

Order.java

```
package com.ww.entity;

import lombok.Data;

import java.util.Date;

@Data
public class Order {
    private long id;
    private User user;
    private Menu menu;
    private Admin admin;
    private Date date;
    private int state;
}
```

OrderVO.java

```
package com.ww.entity;

import lombok.Data;

import java.util.List;

@Data
public class OrderVO {
    private int code;
    private String msg;
    private int count;
    private List<Order> data;
}
```

12.2、修改pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project
xmlns="http://maven.apache.org/POM/4.0.0"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"

xsi:schemaLocation="http://maven.apache.org
/POM/4.0.0
http://maven.apache.org/xsd/maven-
4.0.0.xsd">
    <parent>

 <artifactId>aispringclouddemo</artifactId>
        <groupId>com.ww</groupId>
        <version>1.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <artifactId>order</artifactId>
    <dependencies>
        <dependency>

 <groupId>org.springframework.cloud</groupI
d>
            <artifactId>spring-cloud-
starter-netflix-eureka-client</artifactId>

 <version>2.0.2.RELEASE</version>
```

```
        </dependency>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-starter-config</artifactId>
            <version>2.0.2.RELEASE</version>
        </dependency>
        <dependency>
            <groupId>org.mybatis.spring.boot</groupId>
            <artifactId>mybatis-spring-boot-starter</artifactId>
            <version>1.3.1</version>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.11</version>
        </dependency>

    </dependencies>
```

```
</project>
```

12.3、修改order-dev.yml

```
server:
  port: 8010
spring:
  application:
    name: order
  datasource:
    name: ordersystem
    url:
jdbc:mysql://localhost:3306/ordersystem
    username: root
    password: Root123456
mybatis:
  mapper-locations:
classpath:/mapping/*.xml
  type-aliases-package: com.ww.entity
```

12.4、repository

OrderRepository.java

OrderRepository.xml

12.5、修改OrderHandler.java