

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、修改菜品、删除菜品、订单处理、添加用户、查询用户、删除用户。



account 提供账户服务：用户和管理的登录退出。

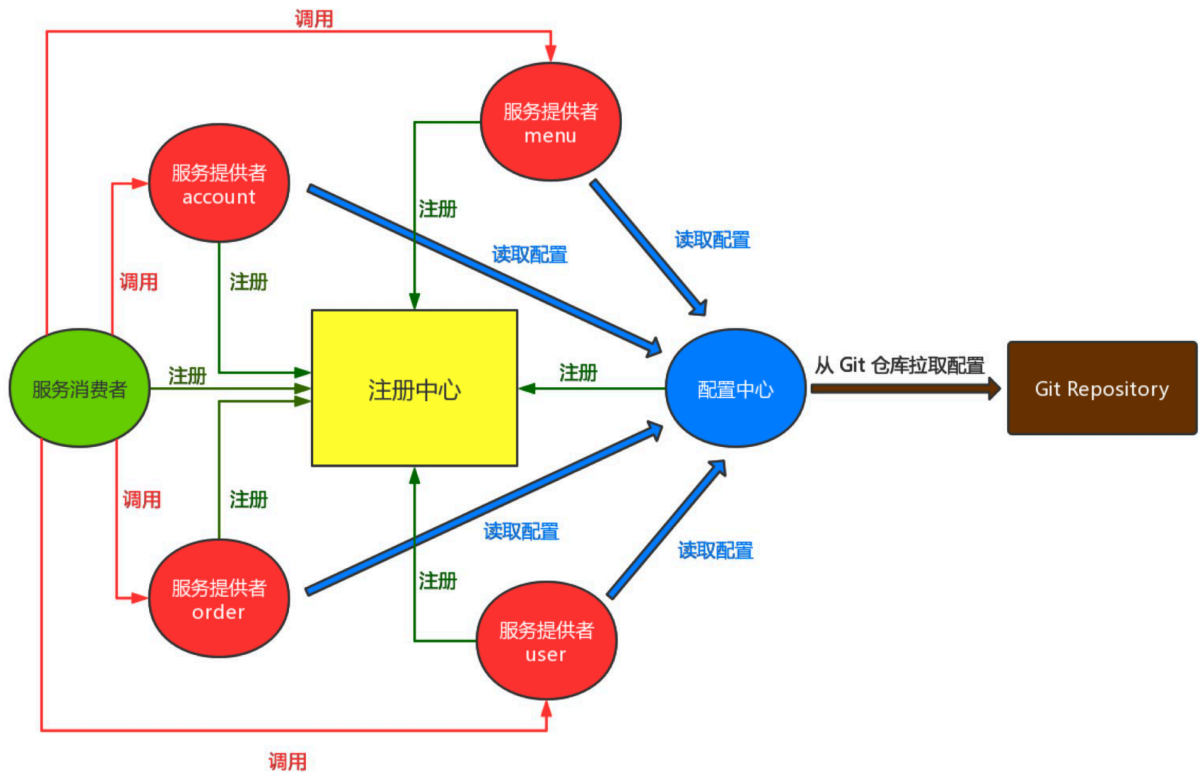
menu 提供菜品服务：添加菜品、删除菜品、修改菜品、查询菜品。

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。

分离出一个服务消费者，调用以上四个服务提供者，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求调用四个服务提供者的业务逻辑，通过 Feign 实现负载均衡。

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，同时可以使用配置中心来对配置文件进行统一集中管理。



- 创建父工程, pom.xml

```
<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 9 缺失jar -->
  <dependency>
    <groupId>javax.xml.bind</groupId>
    <artifactId>jaxb-api</artifactId>
    <version>2.3.0</version>
  </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>

<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-dependencies</artifactId>
            <version>Finchley.SR2</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>

```

注册中心

- pom.xml

```

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

```

- application.yml

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

- 启动类

```
package com.southwind;

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);
    }
}
```

配置中心

- pom.xml

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

- 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

client-dev.yml

```

server:
  port: 8030
spring:
  application:
    name: client
  thymeleaf:
    prefix: classpath:/static/
    suffix: .html
eureka:
  client:
    service-url:
      defaultZone: http://localhost:8761/eureka/
  instance:
    prefer-ip-address: true

```

menu-dev.yml

```

server:
  port: 8020
spring:
  application:
    name: menu
  datasource:
    name: orderingsystem
    url: jdbc:mysql://localhost:3306/orderingsystem?
useUnicode=true&characterEncoding=UTF-8
    username: root
    password: root
eureka:
  client:
    service-url:

```

```
    defaultZone: http://localhost:8761/eureka/
instance:
    prefer-ip-address: true
mybatis:
    mapper-locations: classpath:/mapping/*.xml
    type-aliases-package: com.southwind.entity
```

- 启动类

```
package com.southwind;

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);
    }
}
```

服务提供者 order

- pom.xml

```
<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>
```

- bootstrap.yml

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

- Handler

```
package com.southwind.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的端口: "+this.port;
    }
}
```

- 启动类

```
package com.southwind;

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);
    }
}
```

- pom.xml

```
<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.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>
```

- bootstrap.yml

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

- Handler

```
package com.southwind.controller;

import com.southwind.entity.Menu;
import com.southwind.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 this.port;
    }

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

```

- 启动类

```

package com.southwind;

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

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

```

- Menu 实体类

```
package com.southwind.entity;

import lombok.Data;

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

- 创建 MenuRepository 接口

```
package com.southwind.repository;

import com.southwind.entity.Menu;

import java.util.List;

public interface MenuRepository {
    public List<Menu> findAll();
    public int count();
    public Menu findById(long id);
    public void save(Menu menu);
    public void update(Menu menu);
    public void deleteById(long id);
}
```

- resources 路径下创建 mapping 文件夹，存放 Mapper.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.southwind.repository.MenuRepository">

    <select id="findAll" resultType="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="Menu">
        select * from t_menu where id = #{id}
    </select>
```

```

<insert id="save" parameterType="Menu">
    insert into t_menu(name,price,flavor) values(#{name},#{price},#{flavor})
</insert>

<update id="update" parameterType="Menu">
    update t_menu set name = #{name},price = #{price},flavor = #{flavor}
where id = #{id}
</update>

<delete id="deleteById" parameterType="long">
    delete from t_menu where id = #{id}
</delete>
</mapper>

```

服务消费者 client

- pom.xml

```

<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-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>

```

- bootstrap.yml

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

- 启动类

```
package com.southwind;

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);
    }
}
```