详细参看品优购完整资料

创建工程（参看前面）

# 注册服务

## 注册服务

注册服务，就是在服务上添加Eureka的客户端依赖，客户端代码会自动把服务注册到EurekaServer中。

## 修改itcast-service-provider工程

1. 在pom.xml中，添加springcloud的相关依赖。

2. 在application.yml中，添加springcloud的相关依赖。

3. 在引导类上添加注解，把服务注入到eureka注册中心。

## 修改itcast-service-consumer工程

接下来我们修改itcast-service-consumer，尝试从EurekaServer获取服务。

方法与消费者类似，只需要在项目中添加EurekaClient依赖，就可以通过服务名称来获取信息了！

类似

## 修改Pom

<**spring-cloud.version**>Finchley.SR2</**spring-cloud.version**>

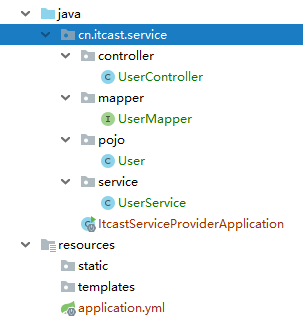
一定区分开来，那面是\*-server



*<!-- SpringCloud的依赖 -->*



# 改造服务提供方



## 配置

### Yml

本工程微服务名

映射到注册中心



这里我们添加了spring.application.name属性来指定应用名称，将来会作为应用的id使用。

### 配置启用Eureka客户端

在引导类上开启Eureka客户端功能，通过添加@EnableDiscoveryClient来开启Eureka客户端功能

*// 启用Eureka客户端，Spring Could 提供的注解; @EnableEurekaClient 什么组件提供的注解（忽略）*

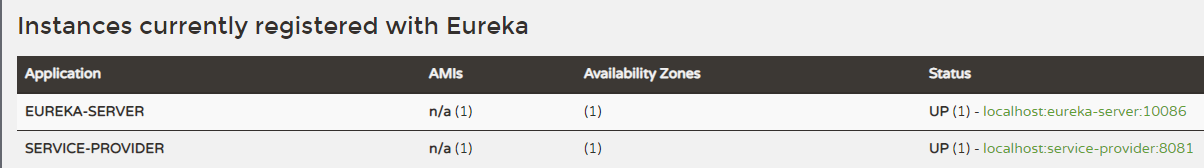


## 测试

### 一、访问地址

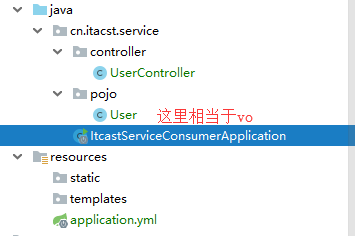
### 二、监控页面

重启项目，访问Eureka监控页面查看



我们发现service-provider服务已经注册成功了

# 改造服务调用方

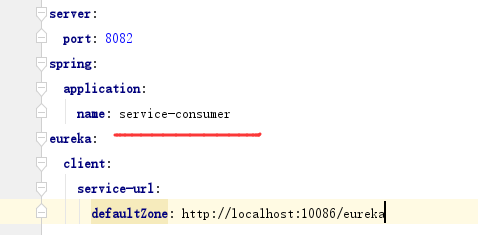


## 配置

### Yml

本工程微服务名

映射到注册中心



这里我们添加了spring.application.name属性来指定应用名称，将来会作为应用的id使用。

### 配置启用Eureka客户端

在引导类上开启Eureka客户端功能，通过添加@EnableDiscoveryClient来开启Eureka客户端功能

*// 启用Eureka客户端，Spring Could 提供的注解; @EnableEurekaClient 什么组件提供的注解（忽略）*



## 编码

### Pojo（vo）

这里相当于一个vo

### Controller

修改UserController代码，用DiscoveryClient类的方法，根据服务名称，获取服务实例：



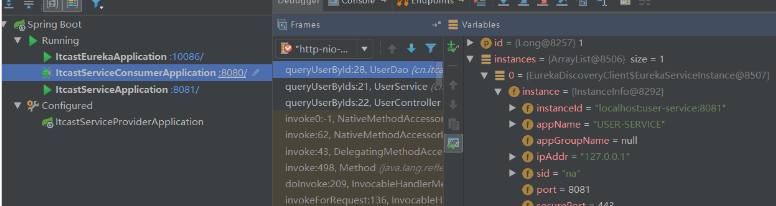


## 测试

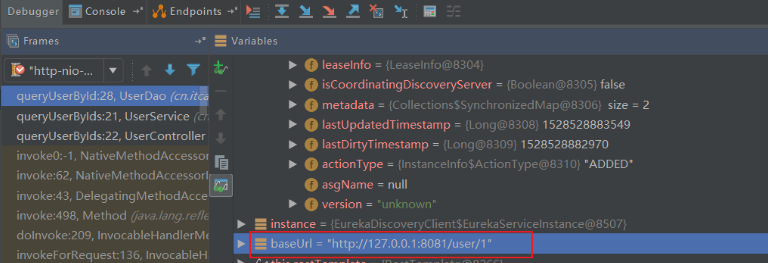
### 访问地址

前提：服务方必须启动

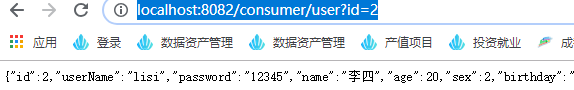
启动项目，访问接口：<http://localhost:8082/consumer/user?id=2>



生成的URL：

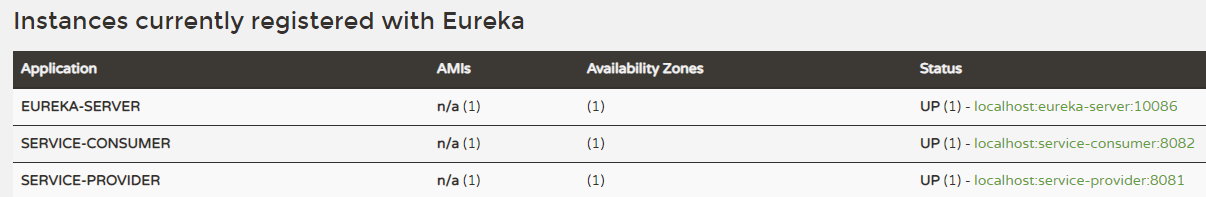


访问结果：



### 监控页面

重启项目，访问Eureka监控页面查看



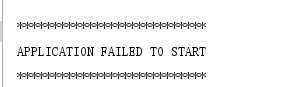
我们发现service-provider服务已经注册成功了

# 问题

## ym启动报错

检查编码是不是都是utf-8

## The bean 'dataSource', defined in BeanDefinition defined in class path resource



Spring boot 和 Spring Could 搭配不好

https://blog.csdn.net/izibeike/article/details/86292130