# 搭建用户系统

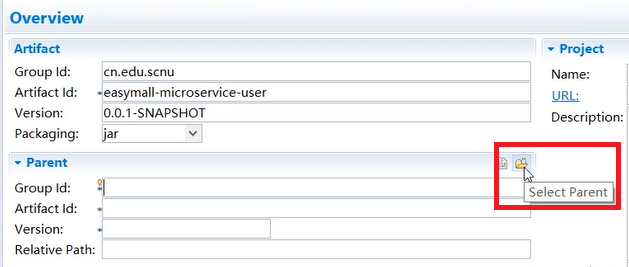
## 搭建步骤

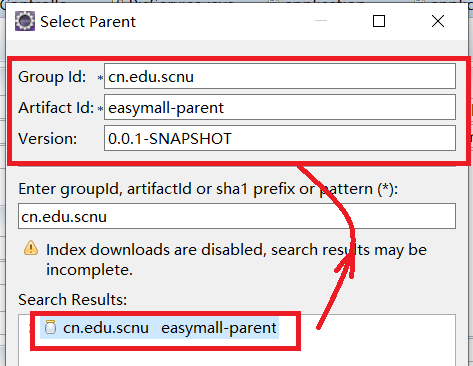
### 1.1.1 创建一个quickstart工程

**easymall-microservice-user**

### 1.1.2 pom文件

#### 继承easymall-parent





<!-- 继承自己的父工程 -->

<parent>

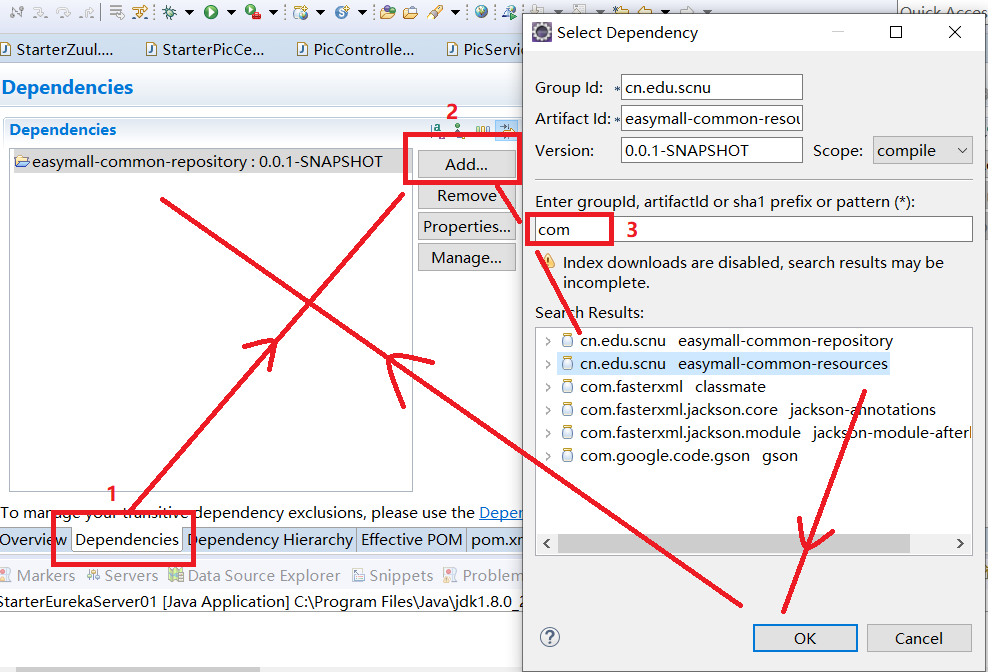
<groupId>cn.edu.scnu</groupId>

<artifactId>easymall-parent</artifactId>

<version>0.0.1-SNAPSHOT</version>

</parent>

#### 依赖easymall-common-resources



<dependency>

<groupId>cn.edu.scnu</groupId>

<artifactId>easymall-common-resources</artifactId>

<version>0.0.1-SNAPSHOT</version>

</dependency>

#### 依赖easymall-common-repository

<dependency>

<groupId>cn.edu.scnu</groupId>

<artifactId>easymall-common-repository</artifactId>

<version>0.0.1-SNAPSHOT</version>

</dependency>

#### properties文件

* + 端口,访问路径
  + 数据源使用德鲁伊连接数据源
  + mybatis(别名包com.easymall.common.pojo)

图形用户界面, 文本, 应用程序

描述已自动生成

#### 启动类(eureka客户端,扫描mapper接口包)

文本

描述已自动生成

# 功能开发

## t\_user表格结构

图形用户界面

中度可信度描述已自动生成

user\_id:主键

user\_name:用户名

user\_password:加密的密码

user\_nickname:昵称

user\_email:邮箱

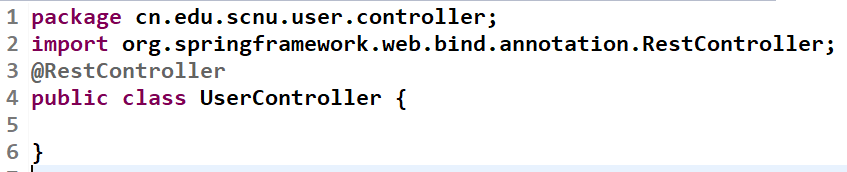
user\_type:默认值是0的,类似lev

## 搭建代码架构

### 2.2.1控制器层

包名：

**cn.edu.scnu.user.controller**



### 2.2.2 Service层

service层包名：

**cn.edu.scnu.user.service**

图片包含 文本

描述已自动生成

### 2.2.3 持久层

持久层包名：

**cn.edu.scnu.user.mapper**

文本

低可信度描述已自动生成

### 2.2.4 映射文件

文本

描述已自动生成

UserMapper.xml(清空拷贝的标签和修改namespace)

<?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=*"****cn.edu.scnu.user.mapper.UserMapper****"*>

</mapper>

图形用户界面, 文本, 应用程序

中度可信度描述已自动生成

## 注册功能

### 2.3.1用户名校验工作

#### 2.3.1.1功能分析

js代码分析

图形用户界面, 文本, 应用程序

描述已自动生成

Nginx配置

图形用户界面, 文本, 应用程序

描述已自动生成

Js请求地址：

[http://www.easymall.com**/user**/checkUserName](http://www.easymall.com/user/checkUserName)

Nginx转发

[http://127.0.0.1:9005/zuul-user/user/manage**/checkUserName**](http://127.0.0.1:9005/zuul-user/user/manage/checkUserName)

**网关路由配置：**

zuul.routes.zuul-user.path=/zuul-user/\*\*

zuul.routes.zuul-user.serviceId=userservice

路由转发：

[http://userservice/**user/manage/checkUserName**](http://userservice/user/manage/checkUserName)

ribbon抓取

[http://127.0.0.1:10003/**user/manage/checkUserName**](http://127.0.0.1:10003/user/manage/checkUserName)

**接口文件：**

|  |  |
| --- | --- |
| js请求地址 | <http://www.easymall.com/user/checkUserName> |
| 后台接收 | /user/manage/checkUserName |
| 请求方式 | Post |
| 请求参数 | String userName |
| 返回数据 | 返回SysResult对象的json,其结构:  Integer status; 200表示用户名不存在,可以使用,其他表示不可用  String msg;成功返回 “ok”,失败返回其他信息  Object data;根据需求携带其他数据 |
| 备注 | 返回200 sql返回0 select count(user\_id) from t\_user where user\_name=#{userName},否则返回sysresult的status==其他值 |

#### 2.3.1.2代码编写

##### UserController

文本

描述已自动生成

##### UserService

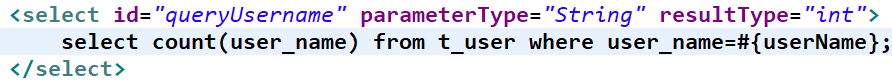
图形用户界面, 文本

描述已自动生成

##### UserMapper



##### 映射文件UserMapper.xml



#### 2.3.1.3 前端整合测试

##### zuul网关添加路由

zuul.routes.zuul-user.path=/zuul-user/\*\*

zuul.routes.zuul-user.serviceId=userservice

##### nginx调整/user转向网关配置

location /user {

proxy\_pass <http://127.0.0.1:9005/zuul-user/user/manage>;

add\_header 'Access-Control-Allow-Credentials' 'true';

add\_header 'Access-Control-Allow-Origin' '\*';

}

##### 测试

1. 启动Eureka（2个）
2. 启动Zuul网关
3. 启动product服务 ribbon
4. 启动user服务

### 2.3.2表单提交注册

#### 2.3.2.1功能分析

日程表

描述已自动生成

Js请求地址：

[http://www.easymall.com**/user**/save](http://www.easymall.com/user/save)

Nginx转发

[http://127.0.0.1:9005/zuul-user/user/manage**/save**](http://127.0.0.1:9005/zuul-user/user/manage/save)

**网关路由配置：**

zuul.routes.zuul-user.path=/zuul-user/\*\*

zuul.routes.zuul-user.serviceId=userservice

路由转发：

[http://userservice/**user/manage/save**](http://userservice/user/manage/save)

ribbon抓取

[http://127.0.0.1:10003/**user/manage/save**](http://127.0.0.1:10003/user/manage/save)

#### 2.3.2.2.接口文件

|  |  |
| --- | --- |
| js请求地址 | <http://www.easymall.com/user/save> |
| 后台接收 | /user/manage/save |
| 请求方式 | Post |
| 请求参数 | User user 缺少用户id,password需要加密 |
| 返回数据 | 返回SysResult对象的json,其结构:  Integer status; 200表示成功,其他表示失败  String msg;成功返回 “ok”,失败返回其他信息  Object data;根据需求携带其他数据 |

#### 2.3.2.3代码编写

##### 1.控制器层

图形用户界面, 文本, 应用程序

描述已自动生成

##### 2.Service层

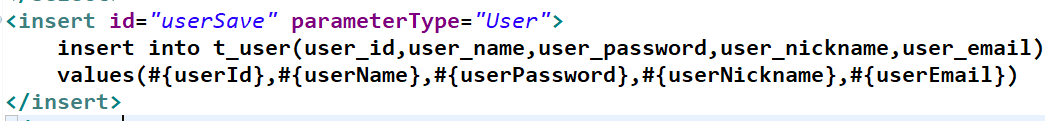
文本

描述已自动生成

##### 3.持久层



##### 4.映射文件



#### 2.3.2.4整合测试

启动Eureka（2个）

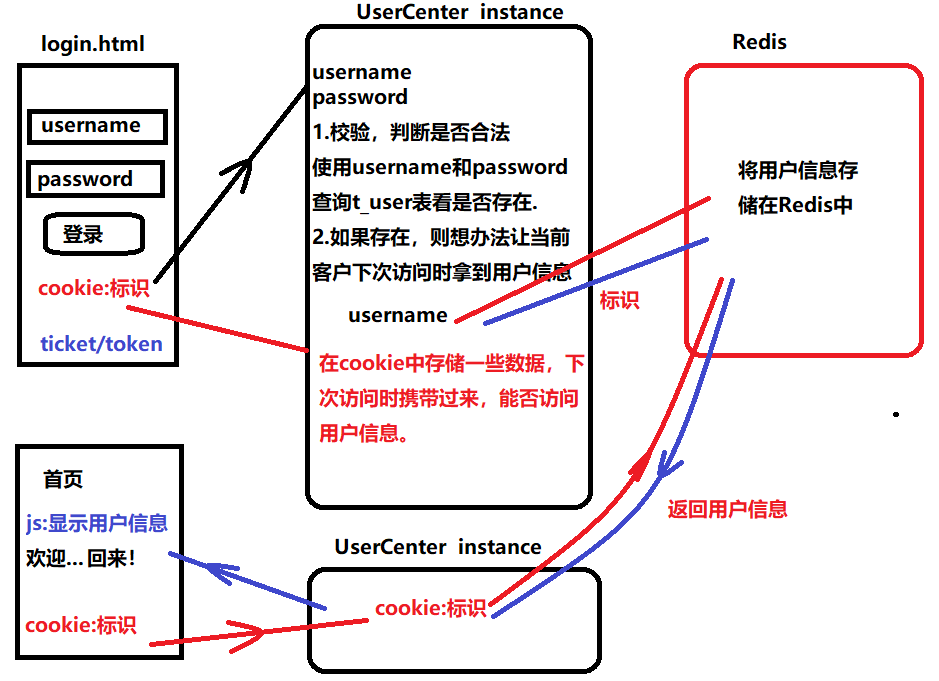
启动Zuul网关

启动product服务 ribbon

启动user服务

## 2.4登录功能

**登录流转过程(下次访问时携带登录状态)**



**暂停代码,进入redis的学习**