实现JSON数据的请求与响应

一、主要知识点

1, @ResponseBody

将Controller的方法返回的对象,通过适当的HttpMessageConverter转换为指定格式后(案例里指 json数据),写入到Response对象的body(响应体)数据区。

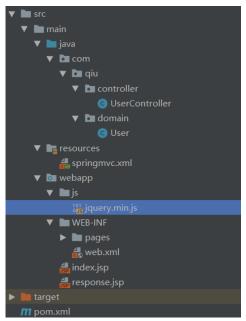
2, json:

是一种轻量级的数据交换格式,采用一种**"键:值"**对的文本格式来存储和表示数据

```
{
"ID": 1001,
"name": "张三",
"age": 24
}
```

二、实现步骤

目录结构



项目准备

- 1、导入jqurey文件jquery.min.js
- 2、导入json的jar包 (方便对 json 数据进行转换成java属性)

```
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.9.0</version>
</dependency>
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-core</artifactId>
  <version>2.9.0</version>
</dependency>
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-annotations</artifactId>
  <version>2.9.0</version>
</dependency>
3、在springmvc.xml配置;避免静态资源被拦截
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"</p>
   xmlns:mvc="http://www.springframework.org/schema/mvc"
   xmlns:context="http://www.springframework.org/schema/context"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd">
  <!-- 开启注解扫描 -->
  <context:component-scan base-package="com.giu"/>
  <!-- 视图解析器对象 -->
  <bean id="internalResourceViewResolver"</pre>
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    contentprefixvalue"/>
    cproperty name="suffix" value=".jsp"/>
  </bean>
  <!--前端控制器,哪些静态资源不拦截-->
  <mvc:resources location="/js/" mapping="/js/**"/>
  <!-- 开启SpringMVC框架注解的支持 -->
  <mvc:annotation-driven />
```

开始项目

4、配置jsp文件进行请求;编写json格式,设置属性和值;并显示后台响应数据 <%--Created by IntelliJ IDEA. User: 彼岸の烟火 Date: 2020/4/7 Time: 21:44 To change this template use File | Settings | File Templates. <%@ page contentType="text/html;charset=UTF-8" language="java" %> <html> <head> <title>Title</title> <script src="js/jquery.min.js"></script> <script> // 页面加载,绑定单击事件 \$(function(){ \$("#btn").click(function(){ \$.ajax({ // 编写json格式,设置属性和值 url:"user/testAjax", //请求后台方法的路径 contentType: application/json; charset = UTF-8, data:'{"username":"hehe","password":"123","age":30}', //前端数据 dataType:"json", type:"post", success:function(data){ // data服务器端响应的json的数据,进行解析 alert(data); alert(data.username); alert(data.password); alert(data.age); } **})**; **})**; }); </script> </head> <body> <button id="btn">发送ajax的请求</button>

```
</body>
```

5、后台响应

```
@Controller
@RequestMapping("/user")
public class UserController {
  /**
  * 模拟异步请求响应
  */
  @RequestMapping("/testAjax")
  public @ResponseBody User testAjax(@RequestBody User user){
    System.out.println("testAjax方法执行了...");
    // 客户端发送ajax的请求,传的是json字符串,后端把json字符串封装到user对象中
    System.out.println(user);
    // 做响应,模拟查询数据库
    user.setUsername("haha");
    user.setAge(40);
    // 做响应
    return user;
  }
}
```