Quiz后端-2、测试Spring contrller接收网络请求

日期：2025.07.19 作者：tfzhang

后端主要提供用户与题目的注册、查询、删除等操作。使用到的开发工具：

- IDEA2024; 关注公众号”青椒工具”，发送”IDEA”，获取windows下的IDEA安装包
- mysql 5.7；关注公众号”青椒工具”，发送”mysql”，获取windows下的mysql5.7安装包；

需求1：无参数请求

```
1、接收浏览器发起的/hello请求，给浏览器返回字符串"hello world"
```

步骤1：创建controller类

在com.tfzhang.quiz目录下创建controller这个目录，然后再创建HelloController这个类：

```
package com.tfzhang.quiz.controller;

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

@RestController //表示当前类为请求处理类；
public class HelloController {

    @RequestMapping("/hello") //表示如果接收到浏览器的/hello请求，就执行下面的hello()方法；
    public String hello() {
        System.out.println("hello");
        return "Hello World";
    }
}
```

步骤2：启动函数，实现浏览器访问

postman, apifox

需求2：简单参数请求

```
1、接收浏览器发起的/simpleParam?name=tom&age=10请求，服务器获取name和age两个参数并打印到终端，同时返回ok。
```

步骤1：创建类处理方法

com.tfzhang.quiz.controller，在HelloController类中添加对应的类处理方法：

```
    @RequestMapping("/simpleParam") //表示如果接收到浏览器的/simpleParam，就执行下面的getParam()方法；
    public String getParam(String name, Integer age) {
        System.out.println(name+":"+age);
        return "ok";
    }
```

步骤2：使用apifox访问

同时使用GET和POST方法访问。

需求3：实体参数请求

```
1、来自浏览器发起请求的参数不止两个，如果参数的数量远大于2个，比如10个；
那么在请求处理方法中，就不能再一个个的写变量；最好可以封装到一个类中；那此处我们以需求2为例，实现如何采用实体类来获取参数。
```

步骤1：在model中创建SimpleUser这个类

```
public class SimpleUser{
    private String name;
    private Integer age;
    //...get,set,tostring等方法；
}
```

步骤2：创建类处理方法simpleUser

```
    @RequestMapping("/simpleUser") //表示如果接收到浏览器的/simpleUser，就执行下面的getUser()方法；
    public String getUser(SimpleUser user) {
        System.out.println(user);
        return "ok";
    }
```

需求4：统一响应结果

修改上述的需求3的返回结果，让其返回所接收到的对象；

现在发现需求3和需求1返回的数据结果更异，于是出现需求4：

```
后端返回的数据呈现不同的返回形式，
    现在希望后台返回的数据格式可以一致。
```

步骤1：创建统一的返回类

在model目录下，创建一个Result返回类别：

```
public Class Result{
    private Integer code; //1是成功，0是失败；
    private String msg; //成功消息或者失败消息；
    private Object data; //放数据；
    //...get,set,toString
    //构造方法；
    //静态success方法和error方法；
}
```

步骤2：重构各个类处理方法

返回的结果设置为Result。

步骤**3**：测试查看结果

apifox