



链滴

springboot|Controller 接收处理 GET,POST 请求入参

作者: [xiaodaojava](#)

原文链接: <https://ld246.com/article/1572752117707>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



javaDEMO

本网站记录了最全的各种JavaDEMO ,保证下载,复制就是可用的,包括基础的, 集合的, spring的, Mybatis的等等各种,助力你从菜鸟到大牛,记得收藏哦~~

<https://www.javastudy.cloud>

Controller 接收入参

入参通常来说分两个情况,一种是Get, 一种是POST

Get请求的入参处理

```
/**
 * @Author https://www.javastudy.cloud
 * @CreateTime 2019/11/3
 **/
@RestController
public class ParamController {

    /**
     * 这里使用了 GetMapping 和 @RequestMapping(method = RequestMethod.GET)是一样的
     * 在方法入参的位置,把每个参数都平铺开
     * @return
     */
    @GetMapping("param/handleGetParam1")
    public String handleGetParam1(String param1, String param2){
        String result = String.format("in handle1 param1 is %s, param2 is %s", param1, param2);
        return result;
    }
}
```

```

/**
 * 把入参封装成一个实体,如果请求参数过多,一般大于5个时,用这种方法
 * @param demo
 * @return
 */
@GetMapping("param/handleGetParam2")
public String handleGetParam2(ParamDemo demo){
    String result = String.format("in handle2 param1 is %s, param2 is %s", demo.param1, d
mo.param2);
    return result;
}

/**
 * 使用静态内部类,简单定义一个入参实体类
 */
static class ParamDemo{
    private String param1;
    private String param2;

    public String getParam1() {
        return param1;
    }

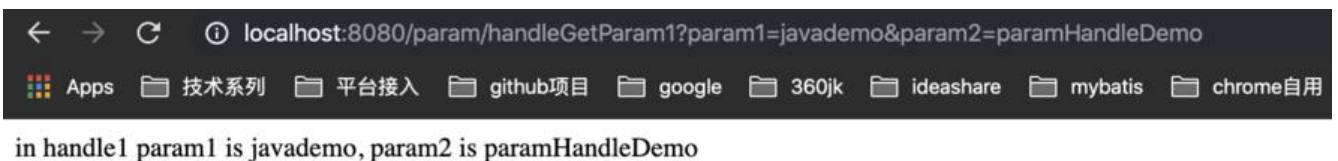
    public void setParam1(String param1) {
        this.param1 = param1;
    }

    public String getParam2() {
        return param2;
    }

    public void setParam2(String param2) {
        this.param2 = param2;
    }
}
}

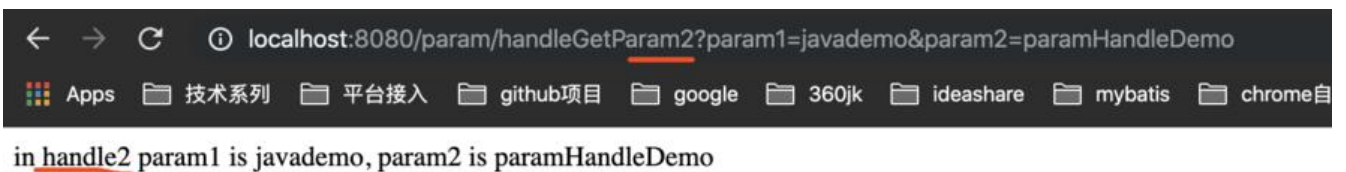
```

结果如下:



localhost:8080/param/handleGetParam1?param1=javademo¶m2=paramHandleDemo

in handle1 param1 is javademo, param2 is paramHandleDemo



localhost:8080/param/handleGetParam2?param1=javademo¶m2=paramHandleDemo

in handle2 param1 is javademo, param2 is paramHandleDemo

post请求入参处理

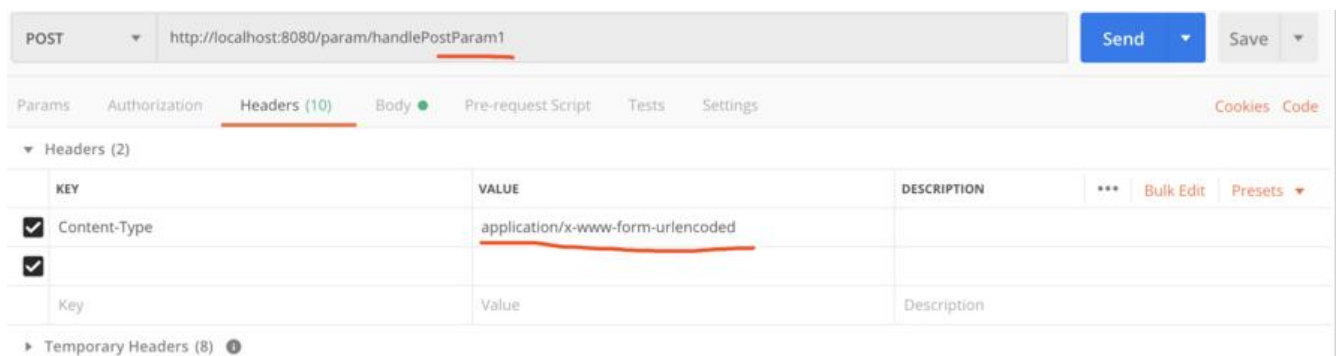
```
/**
 * Post 请求入参也可以这样平铺使用
 * 一般适用于前端Header中Content-Type为application/x-www-form-urlencoded的场景
 * @param param1
 * @param param2
 * @return
 */
@PostMapping("param/handlePostParam1")
public String handlePostParam1(String param1, String param2){
    String result = String.format("in handle1 param1 is %s, param2 is %s", param1, param2);
    return result;
}

/**
 * 入参为一个实体的形式,没有任何注解
 * 一般适用于参数个数较多,在Header中Content-Type为application/x-www-form-urlencoded
场景
 * @param demo
 * @return
 */
@PostMapping("param/handlePostParam2")
public String handlePostParam2(ParamDemo demo){
    String result = String.format("in handle2 param1 is %s, param2 is %s", demo.param1, d
mo.param2);
    return result;
}

/**
 * 请求入参是一个实体,并且加上了 @RequestBody
 * 一般适用于前端Header中Content-Type 为 application/json的场景
 * 注意入参要是json格式
 * @param demo
 * @return
 */
@PostMapping("param/handlePostParam3")
public String handlePostParam3(@RequestBody ParamDemo demo){
    String result = String.format("in handle2 param1 is %s, param2 is %s", demo.param1, d
mo.param2);
    return result;
}
```

PostMan模拟请求结果如下:

模拟时注意修改Header中Content-Type的值



POST http://localhost:8080/param/handlePostParam1

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> param1	javaDemo	
<input checked="" type="checkbox"/> param2	paramPostDemo	
Key	Value	Description

Body Cookies Headers (3) Test Results Status: 200 OK Time: 13ms Size: 170 B Save Response

Pretty Raw Preview Visualize BETA Text

```
1 in handle1 param1 is javaDemo, param2 is paramPostDemo
```

模拟时注意修改Header中Content-Type的值

POST http://localhost:8080/param/handlePostParam3

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies Code

Headers (2)

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Content-Type	application/json	
Key	Value	Description

Temporary Headers (8)

POST http://localhost:8080/param/handlePostParam3

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA JSON

```
1 {
2   "param1": "javaDemo",
3   "param2": "paramPostDemo"
4 }
```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 44ms Size: 170 B Save Response

Pretty Raw Preview Visualize BETA Text

```
1 in handle2 param1 is javaDemo, param2 is paramPostDemo
```

DEMO总评

入参处理这块不是很复杂,主要是要和前端的Header中的Content-Type对应,本次DEMO只讲了String类型的入参处理,后面会分享文件,数组等复杂类型的入参处理. 还有一种入参是@PathVariable的形式,种在实际入开发中使用的少之又少,可以学习,不推荐实际开发中使用.加油吧!

DEMO代码下载

[demospringbootparam.zip](#)