

昵称: [zeroingToOne](#)

园龄: 1年10个月

粉丝: 12

关注: 4

[+加关注](#)

最新随笔

1. java中线程安全和非线程安全的集合
2. hashCode和equal
3. MySQL中char、varchar和nvarchar的区别
4. MySQL存储引擎
5. String在内存中如何存储 (Java)
6. String、StringBuffer、StringBuilder区别
7. MySQL中的事务
8. gbk、utf-8、utf8mb4区别
9. ArrayList的扩容机制
10. 如何在Anaconda中实现多版本python共存

随笔分类

- [Angular\(4\)](#)
- [C#\(1\)](#)
- [DL\(7\)](#)
- [Java\(31\)](#)
- [javaweb\(23\)](#)
- [LaTeX\(2\)](#)
- [LeetCode\(36\)](#)
- [MySQL\(5\)](#)
- [Python\(4\)](#)
- [面试\(9\)](#)
- [小程序\(2\)](#)

随笔档案

- [2018年11月 \(10\)](#)
- [2018年7月 \(2\)](#)
- [2018年6月 \(8\)](#)
- [2018年5月 \(20\)](#)
- [2018年4月 \(19\)](#)
- [2018年3月 \(16\)](#)
- [2018年2月 \(5\)](#)
- [2018年1月 \(10\)](#)
- [2017年12月 \(10\)](#)
- [2017年11月 \(15\)](#)
- [2017年4月 \(1\)](#)

积分与排名

积分 - 54333
排名 - 9668

阅读排行榜

1. @RequestParam、@RequestBody和@ModelAttribute区别(41725)
2. 微信小程序与java后台交互(11855)
3. python中安装pandas(8200)
4. Spring中Model、ModelMap及ModelAttribute之间的区别(6760)
5. 将javaweb项目部署到阿里云服务器(4447)

评论排行榜

1. InvalidArgumentError: You must feed a value for placeholder tensor 'Placeholder_1' with dtype float and shape [?,10] (2)

@RequestParam、@RequestBody和@ModelAttribute区别

一、@RequestParam

GET和POST请求传的参数会自动转换赋值到@RequestParam 所注解的变量上

1. @RequestParam (org.springframework.web.bind.annotation.RequestParam) 用于将指定的请求参数赋值给方法中的形参。

例:

(1) get请求:

url请求: <http://localhost:8080/WxProgram/findAllBookByTag?tagId=1&pageIndex=3>

userTest.jsp

```
<form action="/WxProgram/json/requestParamTest" method="get">
    requestParam Test<br>
    用户名: <input type="text" name="username"><br>
    用户昵称: <input type="text" name="usernick"><br>
    <input type="submit" value="提交">
</form>
```

UserController.java

```
@RequestMapping(value="/requestParamTest", method = RequestMethod.GET)
public String requestParamTest(@RequestParam(value="username") String userName,
    @RequestParam(value="usernick") String userNick){
    System.out.println("requestParam Test");
    System.out.println("username: " + userName);
    System.out.println("usernick: " + userNick);
    return "hello";
}
```

上述代码会将请求中的username参数的值赋给username变量。

等价于:

```
@RequestMapping(value="/requestParamTest", method = RequestMethod.GET)
public String requestParamTest(String username, HttpServletRequest request){
    System.out.println("requestParam Test");
    System.out.println("username: " + username);
    String usernick = request.getParameter("usernick");
    System.out.println("usernick: " + usernick);
    return "hello";
}
```

也可以不使用@RequestParam, 直接接收, 此时要求controller方法中的参数名称要跟form中name名称一致

```
@RequestMapping(value="/requestParamTest", method = RequestMethod.GET)
public String requestParamTest(String username, String usernick){
    System.out.println("requestParam Test");
    System.out.println("username: " + username);
    System.out.println("usernick: " + usernick);
    return "hello";
}
```

总结:

接收请求参数的方式:

3. python中安装pandas(2)
4. C#解析数组形式的json数据(1)
5. @RequestParam、@RequestBody和@ModelAttribute区别(1)

推荐排行榜

1. @RequestParam、@RequestBody和@ModelAttribute区别(5)
2. Win10系统 安装Anaconda+TensorFlow+Keras(2)
3. elasticsearch模板 去掉Preprint submitted to(1)
4. mybatis中@Param的使用(1)
5. Spring、MyBatis和SpringMVC整合的jar包下载(1)

```
String username, String usernick// 此时要参数名称一致
HttpServletRequest request //request.getParameter("usernick")
```

(2) post请求:

跟get请求格式一样, 只是把方法中的get换成post

@RequestParam

用来处理Content-Type: 为 application/x-www-form-urlencoded编码的内容。提交方式为get或post。
(Http协议中, 如果不指定Content-Type, 则默认传递的参数就是application/x-www-form-urlencoded类型)

RequestParam实质是将Request.getParameter() 中的Key-Value参数Map利用Spring的转化机制ConversionService配置, 转化成参数接收对象或字段。

get方式中query String的值, 和post方式中body data的值都会被Servlet接受到并转化到Request.getParameter()参数集中, 所以@RequestParam可以获取的到。

二. @RequestBody

@RequestBody注解可以接收json格式的数据, 并将其转换成对应的数据类型。

1. @RequestBody接收一个对象

url请求: <http://localhost:8080/WxProgram/findBookByName>

```
@RequestMapping(value="/findBookByName", method = RequestMethod.POST)
@ResponseBody
public DbBook findBookByName(@RequestBody DbBook book){
    System.out.println("book: " + book.toString());
    System.out.println("book name: " + book.getTitle());
    String bookName = book.getTitle();
    DbBook book = wxService.findBookByName(bookName);
    return book;
}
```

2. @RequestBody接收不同的字符串

(1)前台界面, 这里以小程序为例

```
wx.request({
  url: host.host + `/WxProgram/deleteBookById`,
  method: 'POST',
  data: {
    nick: this.data.userInfo.nickName,
    bookIds: bookIds
  },
  success: (res) => {
    console.log(res);
    this.getCollectionListFn();
  },
  fail: (err) => {
    console.log(err);
  }
})
```

(2)controller

```
@RequestMapping(value="/deleteBookById",method=RequestMethod.POST)
@ResponseBody
public void deleteBookById(@RequestBody Map<String, String> map){
    String bookIds = map.get("bookIds");
    String nick = map.get("nick");
    String[] idArray = bookIds.split(",");
    Integer userId = wxService.findByIdByNick(nick);
    for(String id : idArray){
```

```
        wxService.removeBookById(bookid, userId);
    }
}
```

@RequestBody

处理HttpEntity传递过来的数据，一般用来处理非Content-Type: application/x-www-form-urlencoded编码格式的数据。

GET请求中，因为没有HttpEntity，所以@RequestBody并不适用。

POST请求中，通过HttpEntity传递的参数，必须要在请求头中声明数据的类型Content-Type，SpringMVC通过使用HandlerAdapter 配置的HttpMessageConverters来解析HttpEntity中的数据，然后绑定到相应的bean上。

@RequestBody用于post请求，不能用于get请求

这里涉及到使用@RequestBody接收不同的对象

1. 创建一个新的entity，将两个entity都进去。这是最简单的，但是不够“优雅”。
2. 用Map<String, Object>接受request body，自己反序列化到各个entity中。
3. 类似方法2，不过更为generic，实现自己的HandlerMethodArgumentResolver。参考[这里](#)

三、@ModelAttribute

@ModelAttribute注解类型将参数绑定到Model对象

1. userTest.jsp

```
<form action="/WxProgram/json/modelAttributeTest" method="post">
    modelAttribute Test<br>
    用户id: <input type="text" name="userId"><br>
    用户名: <input type="text" name="userName"><br>
    用户密码: <input type="password" name="userPwd"><br>
    <input type="submit" value="提交"><br>
</form>
```

name的属性值要跟User的属性相对应。

2. UserController.java

```
@RequestMapping(value="/modelAttributeTest", method = RequestMethod.POST)
public String modelAttributeTest(@ModelAttribute User user) {
    System.out.println("modelAttribute Test");
    System.out.println("userid: " + user.getUserId());
    System.out.println("username: " + user.getUserName());
    System.out.println("userpwd: " + user.getUserPwd());
    return "hello";
}
```

3. User.java

```
public class User {
    private Integer userId;
    private String userName;
    private String userPwd;

    public User() {
        super();
    }

    //setter and getter
}
```

1. application/x-www-form-urlencoded，这种情况的数据@RequestParam、@ModelAttribute可以处理，@RequestBody也可以处理。
2. multipart/form-data，@RequestBody不能处理这种格式的数据。（form表单里面有文件上传时，必须要指定enctype属性值为multipart/form-data，意思是以二进制流的形式传输文件。）
3. application/json、application/xml等格式的数据，必须使用@RequestBody来处理。

参考：

[@RequestBody和@RequestParam区别](#)

<https://segmentfault.com/q/1010000009017635>

分类: [javaweb](#)



[zeroingToOne](#)

[关注 - 4](#)

[粉丝 - 12](#)

[+加关注](#)

5

0

« 上一篇: [java.lang.IllegalArgumentException: Invalid character found in the request target. The valid characters are defined in RFC 7230 and RFC 3986](#)

» 下一篇: [通过域名访问部署在服务器上的javaweb项目](#)

posted @ 2018-05-07 16:52 zeroingToOne 阅读(41725) 评论(1) 编辑 收藏

评论列表

#1楼 2018-09-21 10:44 面向老婆编程

很详细

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【幸运】99%的人不知道我们有可以帮你薪资翻倍的秘笈！

【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码

【推荐】百度云“猪”你开年行大运，红包疯狂拿

【推荐】55K刚面完Java架构师岗，这些技术你必须掌握

相关博文：

- [@RequestParam和@RequestBody区别](#)
- [@RequestParam,@PathVariable,@ResponseBody,@RequestBody,@ModelAttribute学习](#)
- [@RequestBody和@RequestParam区别](#)
- [@RequestParam和@RequestBody的区别？](#)
- [springMVC中的@RequestBody和@ResponseBody以及@RequestParam](#)

最新新闻：

- [惹怒程序员的下场！阿里达摩院大神难忍骚扰电话，打造“二哈”AI](#)

- [量子实验暗示客观现实不存在](#)
- [百度发布NLP模型ERNIE，在多个中文NLP任务中表现超越BERT](#)
- » [更多新闻...](#)