

3、Rest

笔记本: spring
创建时间: 2022/4/4 17:24
作者: 雷丰阳

Rest: 系统希望以非常简洁的URL地址来发请求;
怎样表示对一个资源的增删改查用请求方式来区分

/getBook?id=1 : 查询图书
/deleteBook?id=1: 删除1号图书
/updateBook?id=1:更新1号图书
/addBook : 添加图书

Rest推荐;
url地址这么起名; /资源名/资源标识符
/book/1 : GET-----查询1号图书
/book/1 :PUT-----更新1号图书
/book/1 :DELETE-----删除1号图书
/book :POST-----添加图书
系统的URL地址就这么来设计即可;
简洁的URL提交请求, 以请求方式区分对资源操作;
问题: 从页面上只能发起两种请求, GET、POST;
其他的请求方式没法使用;

使用Rest来构建一个增删改查系统;
页面地址:

<!-- 发起图书的增删改查请求; 使用Rest风格的URL地址;
请求url 请求方式 表示含义
/book/1 GET: 查询1号图书
/book/1 DELETE: 删除1号图书
/book/1 PUT: 更新1号图书
/book POST: 添加1号图书
从页面发起PUT、DELETE形式的请求
-->

处理程序:

```
package com.atguigu.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

@Controller
public class BookController {
    /**
     * 处理查询图书请求
     * @param id
     * @return
     */
    @RequestMapping(value="/book/{bid}",method=RequestMethod.GET)
    public String getBook(@PathVariable("bid")Integer id) {
        System.out.println("查询到了"+id+"号图书");
        return "success";
    }

    /**
     * 图书删除
     * @param id
     * @return
     */
    @RequestMapping(value="/book/{bid}",method=RequestMethod.DELETE)
    public String deleteBook(@PathVariable("bid")Integer id) {
        System.out.println("删除了"+id+"号图书");
        return "success";
    }
}
```

```

/**
 * 图书更新
 * @return
 */
@RequestMapping(value="/book/{bid}",method=RequestMethod.PUT)
public String updateBook(@PathVariable("bid")Integer id) {
    System.out.println("更新了"+id+"号图书");
    return "success";
}

@RequestMapping(value="/book",method=RequestMethod.POST)
public String addBook() {
    System.out.println("添加了新的图书");
    return "success";
}
}

```

```

@Override
protected void doFilterInternal(HttpServletRequest request, HttpServletResponse response,
FilterChain filterChain)
    throws ServletException, IOException {
    //获取表单上_method带来的值 (delete\put)
    String paramValue = request.getParameter(this.methodParam);

    //判断如过表单是一个post而且_method有值
    if ("POST".equals(request.getMethod()) && StringUtils.hasLength(paramValue)) {
        //转为PUT、DELETE
        String method = paramValue.toUpperCase(Locale.ENGLISH);
        //重写了request.getMethod();
        HttpServletRequest wrapper = new HttpMethodRequestWrapper(request, method);

        //wrapper.getMethod()===PUT;
        filterChain.doFilter(wrapper, response);
    }
    else {
        //直接放行
        filterChain.doFilter(request, response);
    }
}
}

```

从页面发起PUT、DELETE形式的请求?Spring提供了对Rest风格的支持

1)、SpringMVC中有一个Filter; 他可以把普通的请求转化为规定形式的请求; 配置这个filter;

<filter>

<filter-name>HiddenHttpMethodFilter</filter-name>

<filter-

class>org.springframework.web.filter.HiddenHttpMethodFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>HiddenHttpMethodFilter</filter-name>

<url-pattern>/*</url-pattern>

</filter-mapping>

2)、如何发其他形式请求?

按照以下要求; 1、创建一个post类型的表单 2、表单项中携带一个_method的参数, 3、这个_method的值就是DELETE、PUT

-->

查询图书

<form action="book" method="post">

<input type="submit" value="添加1号图书"/>

```

</form><br/>
<!-- 发送DELETE请求 -->
<form action="book/1" method="post">
    <input name="_method" value="delete"/>
    <input type="submit" value="删除1号图书"/>
</form><br/>
<!-- 发送PUT请求 -->
<form action="book/1" method="post">
    <input name="_method" value="put"/>
    <input type="submit" value="更新1号图书"/>
</form><br/>

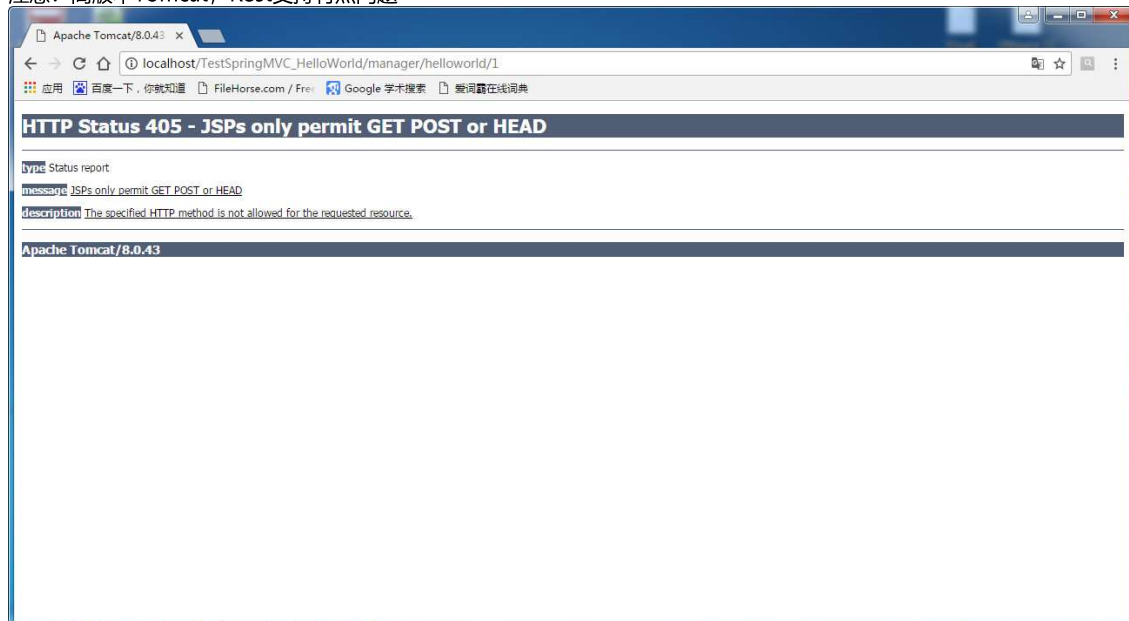
```

```

<filter>
    <filter-name>HiddenHttpMethodFilter</filter-name>
    <filter-class>org.springframework.web.filter.HiddenHttpMethodFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>HiddenHttpMethodFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

```

注意：高版本Tomcat; Rest支持有点问题



解决方案：

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" isErrorPage="true"%>

```