HiddenHttpMehodFilter过滤器

作用：把post请求转化为delete或put请求

1. **配置HiddenHttpMehodFilter过滤器**

<!-- 配置hiddenHttoMethodFilter，作用：把post请求转化为delete或者put请求 -->

<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>

1. **发送各种请求**

<form action=*"hello/testRest/1"* method=*"post"*>

**用隐藏域的方式，告诉filter你要发送的请求是什么，name值固定为\_method,value值为需要发出的请求**

<input type=*"hidden"* name=*"\_method"* value=*"****PUT****"* /> <input type=*"submit"* value=*"test put"* />

</form><br><br>

<form action=*"hello/testRest/1"* method=*"post"*>

<input type=*"hidden"* name=*"\_method"* value=*"****DELETE****"* /> <input type=*"submit"* value=*"test delete"* />

</form><br><br>

<form action=*"hello/testRest"* method=*"****post****"*>

<input type=*"submit"* value=*"test post"* />

</form><br><br>

<a href=*"hello/testRest/1"*>test ***get***</a>

1. **写各种响应函数**

指定接受的请求方式，默认为get

@RequestMapping(value = "/testRest/{id}", **method = RequestMethod.*PUT***)

**public** String testRestPut(@PathVariable Integer id) {

System.***out***.println("Put:" + id);

**return** ***SUCCESS***;

}

@RequestMapping(value = "/testRest/{id}", method = RequestMethod.***DELETE***)

**public** String testRestDelete(@PathVariable Integer id) {

System.***out***.println("Delete:" + id);

**return** ***SUCCESS***;

}

@RequestMapping(value = "/testRest", method = RequestMethod.***POST***)

**public** String testRestPost() {

System.***out***.println("Post");

**return** ***SUCCESS***;

}

@RequestMapping(value = "/testRest/{id}")

**public** String testRestGet(@PathVariable Integer id) {

System.***out***.println("Get:" + id);

**return** ***SUCCESS***;

}

使用@PathVariable注解，映射url占位符到参数中的方式，Rest风格的URL

新增：/order/ Post

删除：/order/1 Delete

更新：/order/1 Put

获取：/order/1 Get

简洁明了