

## 2-6 配置后端验证功能

---

140.143.132.225:8000/project-1/doc-18

对于客户端提交表单或者Ajax中的数据，后端的Controller必须先要做验证，然后才能使用这些数据。既然要验证数据，那么不妨我们来使用一下 **Validation** 库。

### 一、添加依赖

---

在 **pom.xml** 文件中添加依赖，然后让Maven加载依赖库。

```
1. <dependency>
2.     <groupId>org.springframework.boot</groupId>
3.     <artifactId>spring-boot-starter-validation</artifactId>
4. </dependency>
```

### 二、创建Form类

---

**validation** 库在做后端验证的时候，要求必须用封装类（Form类）来保存客户端提交的数据，然后在封装类中，我们可以定义验证的规则，**validation** 会执行这些规则，帮我们验证客户端提交的数据。

我们为之前的 **TestController** 里面的 **sayHello()** 方法设置一个Form类，接受客户端提交的 **name** 数据。我们在 **com.example.emos.wx.controller.form** 包里面创建 **TestSayHelloForm** 类。

```
1. package com.example.emos.wx.controller.form;
2. import io.swagger.annotations.ApiModel;
3. import lombok.Data;
4. import javax.validation.constraints.NotBlank;
5. import javax.validation.constraints.Pattern;
6. @ApiModel
7. @Data
8. public class TestSayHelloForm {
9.     @NotBlank
10.    @Pattern(regexp = "^[\\u4e00-\\u9fa5]{2,15}$")
11.    @ApiModelProperty("姓名")
12.    private String name;
13. }
```

### 三、修改sayHello()方法

---

```

1. package com.example.emos.wx.controller;
2. import com.example.emos.wx.common.util.R;
3. import com.example.emos.wx.controller.form.TestSayHelloForm;
4. import io.swagger.annotations.Api;
5. import io.swagger.annotations.ApiOperation;
6. import org.springframework.web.bind.annotation.*;
7. import javax.validation.Valid;
8. import javax.validation.Validation;
9. @RestController
10. @RequestMapping("/test")
11. @Api("测试Web接口")
12. public class TestController {
13.     @PostMapping("/sayHello")
14.     @ApiOperation("最简单的测试方法")
15.     public R sayHello(@Valid @RequestBody TestSayHelloForm form){
16.         return R.ok().put("message", "Hello, "+form.getName());
17.     }
18. }

```

## 四、执行测试

**test-controller** Test Controller

POST

/test/sayHello 最简单的测试方法

Parameters

Name

Description

form required

form

(body)

Example Value

Model

```
{
  "name": "张无忌"
}
```

Cancel