

记录时间：2019.10.14

公司名称：火鹰科技有限公司

学生姓名：曾伟涛

记录类型：学习笔记

记录阶段：2019.10.14-2019.10.20

前言：本周主要实现springboot MVC技术，实现简单的MVC，下周将结合thymeleaf以及结合freemark实现简单的功能，直观的对比thymeleaf 与 freemark 两个模板之间的不同！

springboot MVC 技术

1.什么是MVC?

之前写过的 springboot 整合 mybatis+swagger 实现数据库的增删改查，可能没发现已经不知不觉中我们运用了MVC。MVC的全称是（Model view Controller）。

Model(模型)，在上个小的项目中，有个很明显的model包，包下有xxxmodel,没错，这就是MVC中的model，一般moel类是写相关数据，常见的model中的写法如下：

```
1 @Data
2 @EqualsAndHashCode(callSuper = false)
3 @Table(name = "buying")
4 public class BuyingModel extends Paging {
5
6     @Column(id = true)
7     private Integer id;
8
9     /**
10      * 用户id
11      */
12     @Column(name = "user_id")
13     private Integer userId;
14     .....
15 }
```

看到这样的写法，其实很明显发现，并不难，写得是与数据库相关的，为什么这样说？剖析如下：

@Data 并不是spring springboot jpa mybatis的注解，它是**import** lombok.Data;什么是lombok，他是一个plugins(插件)，这个的作用是封装了setter/getter 等方法。

@EqualsAndHashCode(callSuper = false) 也是应用Lombok插件，作用是自动的给model bean实现equals方法和hashCode方法。

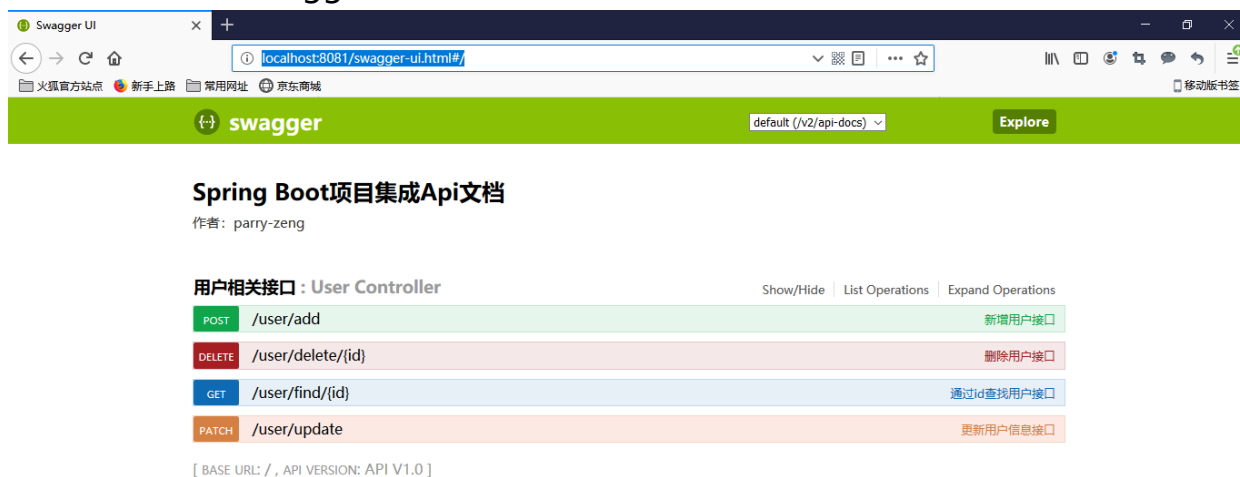
@Table (name = "buying") 对应的是数据库表名

@Column(id = true) 说的是数据库表里id这个字段是主键，而@Column(name = "xxx")说的是数据库表里的字段名

@Column(name = "user_id")与private Integer userId; 意思是 userId 的数据将传到数据库表里的user_id这个字段

看到这里，你是不是对springboot很有兴趣了呢？别急，继续往下看。

view(视图)：既然是视图的意思，很显然就是眼睛看得到的，像天猫 京东 可以看到我们输入想要的“图”，在上次的项目中，我们在浏览器地址栏输入localhost:8081/swagger-ui.html,看到了这样一张图。



除了在pom.xml添加了它相应的依赖，重点是下面你这个依赖，要实现视图，得加入这个依赖

```
1 <dependency>
2 <groupid>org.springframework.boot</groupid>
3 <artifactid>spring-boot-starter-web</artifactid>
4 </dependency>
```

单纯的model以及view是实现不了的，需要用控制器来控制model与view之间的关系，那就需要用到Controller。

Controller(控制): 一般项目的写法是前后端分离, 这里的Controller将写得是接口, 与前端交互。那么, 在controller中常见的API接口写法如下:

```
1 @Api(tags = "求购信息列表-API")
2 @RestController
3 @RequestMapping("Buying:buying")
4 public class BuyingController {
5
6     @Autowired
7     private BuyingService buyingService;
8
9     @ApiOperation(httpMethod = "POST", value = "创建", produces = MediaType.APPLICATION_JSON_UTF8_VALUE, consumes = MediaType.APPLICATION_JSON_UTF8_VALUE)
10    @PostMapping(value = "", produces = MediaType.APPLICATION_JSON_UTF8_VALUE, consumes = MediaType.APPLICATION_JSON_UTF8_VALUE)
11    CloudApiResponse<BuyingProjection> create(@Validated @RequestBody BuyingApiPost api) {
12        CloudApiResponse<BuyingProjection> apiResponse = new CloudApiResponse<>();
13        BuyingModel model = buyingService.save(map(api, BuyingModel.class));
14        apiResponse.setData(model, BuyingProjection.class);
15        return apiResponse;
16    }
```

这里的代码写得是提交数据的接口, 也就是常说的前端提交数据, 后端接收, 看到这里是不是发现写法很简单, 代码量很少, 就能实现一个接口。

2.走进springboot MVC 之门

超级无敌简单的入门案例: hello world!

1.创建springboot项目, 不知道怎么用idea创建请看第一周周报

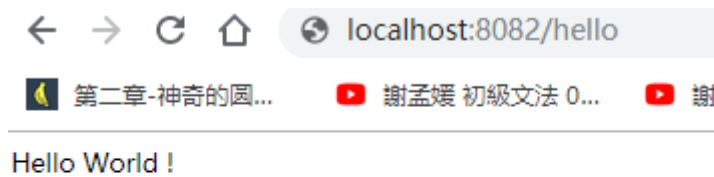
2.在主包下创建子包controller, 在这个子包下新建java类 HelloWorldController,

代码如下:

```
1 @RestController
2 public String sayHello(){
3     @RequestMapping("/hello")
4     return "Hello World !"
```

```
5 }
```

由于我配置的端口是8082，在地址栏输入是localhost:8082/hello



进阶：页面跳转

我们在resources文件夹下新建个目录叫static，在static下新建login.html,user.html
代码如下：

login.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>登录</title>
6 </head>
7 <body>
8   <a href="user.html">请先登录</a>
9 </body>
10 </html>
```

user.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6 </head>
7 <body>
8   <h1 style="text-align: center">用户登录</h1>
9   username: <input type="text"><br>
10  password: <input type="password"><br>
11  <input type="submit" value="登录">
12 </body>
```

```
13 </html>
```

在地址栏输入：localhost:8082/static/login.html

← → ↻ 🏠 ⓘ localhost:8082/springBoot/static/login.html

🌙 第二章-神奇的圆... 📺 謝孟媛 初級文法 0... 📺 謝孟媛 中級文法 0... 📺 易

[请先登录](#)

点击“请先登录”，跳转到user.html

← → ↻ 🏠 ⓘ localhost:8082/springBoot/static/user.html

🌙 第二章-神奇的圆... 📺 謝孟媛 初級文法 0... 📺 謝孟媛 中級文法 0... 📺 哈佛大学公开课：... 📺 MySQL :: MySQL...

用户登录

username:
password:

3.再进阶：将表单输入的用户名与密码与controller交互。代码如下：

user.html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Title</title>
6 </head>
7 <body>
8 <h1 style="text-align: center">用户登录</h1>
9 <form action ="goods/addUser5" method="post">
10 username: <input type="text" name="userName"><br>
11 password: <input type="password" name="password"><br>
12 <input type="submit" value="登录">
13 </form>
```

```
14 </body>
15 </html>
```

HelloWorldController:

```
1 package test.controller;
2
3 import org.springframework.web.bind.annotation.*;
4 import test.model.UserModel;
5
6
7 /**
8  * @author : parry
9  * @Create : 2019-10-14 16:18
10  */
11 @RestController
12 @RequestMapping("/goods")
13 public class HelloWorldController {
14     @RequestMapping(value="/addUser5")
15     public String addUser5(UserModel user) {
16         System.out.println("username is:"+user.getUserName());
17         System.out.println("password is:"+user.getPassword());
18         return "success";
19     }
20 }
```

UserModel:

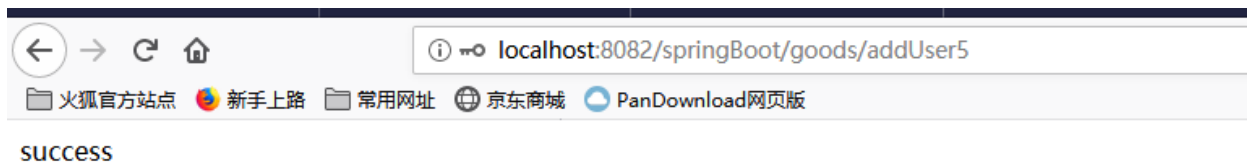
```
1 package test.model; /*
2 import java.awt.Component;
3
4 @author : parry
5 @Create : 2019-09-28 15:12
6
7 import boat.data.model.Paging;
8 import boat.mybatis.annotation.Column;
9 import boat.mybatis.annotation.Table;
```

```

10 import io.swagger.annotations.ApiModel;
11 import io.swagger.annotations.ApiModelProperty;
12 import lombok.Data;
13 import lombok.EqualsAndHashCode;
14
15
16 @Data
17 @EqualsAndHashCode(callSuper = false)
18 @Table(name = "user")
19
20 public class UserModel extends Paging {
21     /**
22      * ID
23      */
24     @Column(id = true)
25     private Integer id;
26
27     /**
28      * 用户名称
29      */
30     @Column(name = "userName")
31     private String userName;
32
33     @Column(name = "password")
34     private String password;
35 }

```

在浏览器的地址栏输入：localhost:8082/user.html,username: parry,
password:132454zlx; 点击提交按钮，会跳转到**controller**类中，会返回**success**:



若不成功会返回此下类似的报错：

org.springframework.web.bind.MissingServletRequestParameterException:
Required String parameter 'userName'.....

控制台输出一下信息：

```
username is:parry  
password is:132454zlx
```

证明成功了。

看完此周周报后，请留下一个问题：没有用thymeleaf、freemarker跟用了它们会有什么不同的效果，下周敬请期待！