# 一、SpringBoot注解(中台作业使用过得注解)

**@PathVariabl 与@RequestParam 区别：**

**浏览器上请求路径不同。一个直接/拼接 一个使用？拼接**

@RequestMapping(value = "/user/{username}/blog/{blogId}")  
public String getUserBlog(@PathVariable String username, @PathVariable int blogId) {  
 return "user:" + username + "blog->" + blogId;  
}

@RequestMapping(value="/user")  
public String getUserBlog(@RequestParam(value="id") int blogId) {  
 return "blogId="+blogId;  
}

@SpringBootApplication 这个注解是Spring Boot最核心的注解，用在 Spring Boot的主类上，标识这是一个 Spring Boot 应用，用来开启 Spring Boot 的各项能力。实际上这个注解是@Configuration,@EnableAutoConfiguration,@ComponentScan三个注解的组合。由于这些注解一般都是一起使用，所以Spring Boot提供了一个统一的注解@SpringBootApplication。

@RestController 用于标注控制层组件(如struts中的action)，表示这是个控制器bean,并且是将函数的返回值直 接填入HTTP响应体中,是REST风格的控制器；它是@Cont roller和@ResponseBody的合集。

@RequestMapping(value = "/grade") RequestMapping是一个用来处理请求地址映射的注解；提供路由信息，负责URL到Controller中的具体函数的映射，可用于类或方法上。用于类上，表示类中的所有响应请求的方法都是以该地址作为父路径。

@Resource 用于标注数据访问组件，即DAO组件。使用@Repository注解可以确保DAO或者repositories提供异常转译，这个注解修饰的DAO或者repositories类会被 ComponetScan发现并配置，同时也不需要为它们提供XML配置项。

@RequestParam 用在方法的参数前面

@GetMapping(value = "param/get")

@PostMapping(value = "param/post")

@Mapper 一般用于修饰mapper层的

@Service 一般用于修饰service层的组件

# 二、增删改查的restful风格：

**查询---GET**

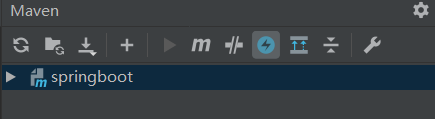
**增加---POST**

**修改---PUT**

**删除---Delete**

**遇到的BUG以及解决：**

## [IDEA出现Please refer to dump files (if any exist) [date].dump, [date]-jvmRun[N].dump and [date].dumpstream的解决方法](https://www.cnblogs.com/li1234567980/p/13388168.html)



点击闪电图标即可。

post和get 请求：

在浏览器上可以采用localhost:8080/hello?str=abc进行get请求 但是采用post请求采用同样的路径却不行。需采用postman或者在浏览器中装插件进行访问。

使用@RespondMapping时，可以指定value(访问路径)也可以指定method （post 还是get）请求方式。

# 三、Springboot 与Mybaits 的整合

新创建spring Initialize 项目 勾选 web sql mybaits

1. @RestController注解，然后直接return json数据即可；

2. @Controller注解放类之前，然后若类中某个方法需要返回json数据，则需在该方法前添加@ResponseBody注解；

## 1 问题

Mybaits中得一对一关联查询方式有两种（新加一个实体类和在原实体类加一个属性）

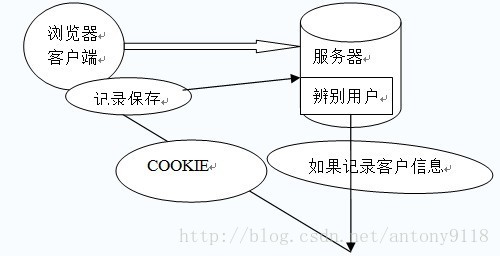
在分数表中加用户表属性时：

1、**如本demo中在分数实体类加入User属性 一定一定要生成get 和set 方法！！！**

2、**报错：INT变成LONG类型 时，**要在所有实体类中加入无参构造方法！！！

/META-INF/resources>resources>static>public

# 四、Cookie和Session的区别



1. 会话（Session）跟踪是Web程序中常用的技术，用来跟踪用户的整个会话。常用的会话跟踪技术是Cookie与Session。Cookie通过在客户端记录信息确定用户身份，Session通过在服务器端记录信息确定用户身份。
2. **Session对象是在客户端第一次请求服务器的时候创建的**
3. 客户端请求服务器，如果服务器需要记录该用户状态，就使用response向客户端浏览器颁发一个Cookie。**客户端浏览器会把Cookie保存起来**。当浏览器再请求该网站时，浏览器把请求的网址连同该Cookie一同提交给服务器。

# 五、Git总结

git clone 地址;将远端仓库拷贝到本地

git remote -v ;获取当前clone 的地址

git remote set-url origin <https://github.com/go2myself/k12.git> ;切换远端地址。

以下三步是最常用的，用于将本地代码提交到远端

git add .;

git commit -m 'update .gitignore';

git push origin master;

git rm -r --cached .    #新增的忽略文件没有生效，是因为git是有缓存的，而之前的文件在缓存中，并不会清除掉，还会继续提交，所以更新.gitignore文件，要清除缓存文件

关于分支：

Git branch master01;新建分支

git checkout master01;切换到分支

Git pull;把分支代码pull下来

Git checkout master;切换到主分支

Git merge master01;把分支的代码merge到主分支

Git push;git push推上去

下面这个插件是可以将不相关的代码去除掉上传到github上。

<https://plugins.jetbrains.com/plugin/7495--ignore>