# JAVA注解原理

# SpringMVC常用注解

## 控制器相关注解

### @Controller

1. 标识该类是一个Spring MVC中的二级控制器类

### @RequestMapping(value, method)

1. 配置url映射
2. 标注类时，就是类内所有映射的父路径

### @ ResponseBody

1. 注解方法，该方法的返回值不再经过view模块，而是直接以指定的格式写入Http response body中返回给用户
2. @RestController注解类：@Controller + @ResponseBody

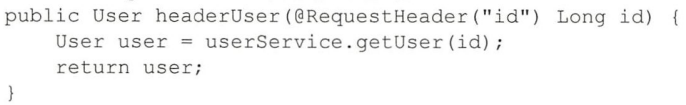
### @RequestParam

1. 用于将请求中的指定参数赋值给方法中的形参
2. 样例
   1. method(@RequestParam(value="username") String userName)
3. 形参名与请求参数名相同时不需要使用value，甚至可以不使用注解就能获取到参数

### @RequestBody

1. 只能用于post请求，接收的是一个json数据，然后转换成指定类型的对象
2. 一个请求只能有一个@RequestBody，可以有多个@RequestParam
3. 样例
   1. method(@RequestBody Map<String, String> map)
   2. method(@RequestBody User user)

### @RequestHeader

1. 获取请求头中的字段
2. 

### @ModelAttribute

1. 与@RequestBody比较类似，将请求参数绑定到一个对象上，对象中的成员变量名要与参数中的字段名相同
2. 样例
   1. method(@ModelAttribute User user)

### @PathVariable

1. 用于获取请求路径URL中的占位符
2. 样例
   1. @RequestMapping(value="/users/{id}/{name}")
   2. method(@PathVariable(value="id") Long id, @PathVariable(value="name") String name)

### @DateTimeFormat和@NumberFormat

1. 用于传递特殊格式的日期和数值数据



## 数据验证相关注解

### @Valid

1. 将HTTP请求的参数注入到自定义类型时开启数据验证
2. SpringMVC验证后会自动填充Errors对象
3. 

## Session相关注解



1. @SessionAttribute：从请求Session中取出相应数据，填充控制器方法参数
2. @SessionAttributes：将模型model中的数据保存至Session中，可以指定保存数据的名称或者数据类型

## 容器相关注解

### @Bean

### @Component @Controller @Service @Repository

1. 都是用来注解类，该类会被自动扫描注册到容器中，成为一个Bean
2. 功能大致相同，只是用来区分被注解类所属层次
   1. @Controller 控制层，提供映射服务
   2. @Service业务层，具体请求处理
   3. @Repository 组件层，DAO组件，也就是具体的数据访问的包装类
   4. @Component 泛指组件，当组件不好归类的时候，我们可以使用这个注解进行标注

### @Autowired

1. 自动装配，用于注解成员变量、方法及构造函数
2. 交由容器去提供具体的对象实例



