



第二版：SpringMVC 35 道

目录

第二版：SpringMVC 35 道	1
什么是 Spring MVC? 简单介绍下你对 Spring MVC 的理解?	2
Spring MVC 的优点	2
核心组件	2
Spring MVC 的主要组件?	2
什么是 DispatcherServlet	3
什么是 Spring MVC 框架的控制器?	3
Spring MVC 的控制器是不是单例模式,如果是,有什么问题,怎么解决?	3
工作原理	4
请描述 Spring MVC 的工作流程? 描述一下 DispatcherServlet 的工作流程?	4
MVC 框架	4
MVC 是什么? MVC 设计模式的好处有哪些	4
常用注解	5
注解原理是什么	5
Spring MVC 常用的注解有哪些?	5
SpringMvc 中的控制器的注解一般用哪个,有没有别的注解可以替代?	6
@Controller 注解的作用	6
@RequestMapping 注解的作用	7
@ResponseBody 注解的作用	7
@PathVariable 和@RequestParam 的区别	7
其他	8
Spring MVC 与 Struts2 区别	8
Spring MVC 怎么样设定重定向和转发的?	8
Spring MVC 怎么和 AJAX 相互调用的?	9
如何解决 POST 请求中文乱码问题, GET 的又如何处理呢?	9
Spring MVC 的异常处理?	10
如果在拦截请求中,我想拦截 get 方式提交的方法,怎么配置	10
怎样在方法里面得到 Request,或者 Session?	10



如果想在拦截的方法里面得到从前台传入的参数,怎么得到?	11
如果前台有很多个参数传入,并且这些参数都是一个对象的,那么怎么样快速得到这个对象?	11
Spring MVC 中函数的返回值是什么?	11
Spring MVC 用什么对象从后台向前台传递数据的?	11
怎么样把 ModelMap 里面的数据放入 Session 里面?	11
Spring MVC 里面拦截器是怎么写的.....	12
介绍一下 WebApplicationContext.....	12

我们的网站: <https://tech.souyunku.com>

关注我们的公众号: 搜云库技术团队, 回复以下关键字

回复: **【进群】** 邀请您进「技术架构分享群」

回复: **【内推】** 即可进: 北京, 上海, 广周, 深圳, 杭州, 成都, 武汉, 南京, 郑州, 西安, 长沙「程序员工作内推群」

回复 **【1024】** 送 4000G 最新架构师视频

回复 **【PPT】** 即可无套路获取, 以下最新整理调优 PPT!

46 页《JVM 深度调优, 演讲 PPT》



53 页《Elasticsearch 调优演讲 PPT》



微信扫一扫

<https://tech.souyunku.com>

技术、架构、资料、工作、内推

专注于分享最有价值的互联网技术干货文章

什么是 Spring MVC? 简单介绍下你对 Spring MVC 的理解?

- Spring MVC 是一个基于 Java 的实现了 MVC 设计模式的请求驱动类型的轻量级 Web 框架, 通过把模型-视图-控制器分离, 将 web 层进行职责解耦, 把复杂的 web 应用分成逻辑清晰的几部分, 简化开发, 减少出错, 方便组内开发人员之间的配合。

Spring MVC 的优点

- (1) 可以支持各种视图技术,而不仅仅局限于 JSP ;
- (2) 与 Spring 框架集成 (如 IoC 容器、AOP 等) ;
- (3) 清晰的角色分配: 前端控制器(dispatcherServlet), 请求到处理器映射(handlerMapping), 处理器适配器 (HandlerAdapter), 视图解析器(ViewResolver) 。
- (4) 支持各种请求资源的映射策略。



核心组件

Spring MVC 的主要组件?

- (1) 前端控制器 DispatcherServlet (不需要程序员开发)
- 作用: 接收请求、响应结果, 相当于转发器, 有了 DispatcherServlet 就减少了其它组件之间的耦合度。
- (2) 处理器映射器 HandlerMapping (不需要程序员开发)
- 作用: 根据请求的 URL 来查找 Handler
- (3) 处理器适配器 HandlerAdapter
- 注意: 在编写 Handler 的时候要按照 HandlerAdapter 要求的规则去编写, 这样适配器 HandlerAdapter 才可以正确的去执行 Handler。
- (4) 处理器 Handler (需要程序员开发)
- (5) 视图解析器 ViewResolver (不需要程序员开发)
- 作用: 进行视图的解析, 根据视图逻辑名解析成真正的视图 (view)
- (6) 视图 View (需要程序员开发 jsp)
- View 是一个接口, 它的实现类支持不同的视图类型 (jsp, freemarker, pdf 等等)

什么是 DispatcherServlet

- Spring 的 MVC 框架是围绕 DispatcherServlet 来设计的, 它用来处理所有的 HTTP 请求和响应。

什么是 Spring MVC 框架的控制器?



- 控制器提供一个访问应用程序的行为，此行为通常通过服务接口实现。控制器解析用户输入并将其转换为一个由视图呈现给用户的模型。Spring 用一个非常抽象的方式实现了一个控制层，允许用户创建多种用途的控制器。

Spring MVC 的控制器是不是单例模式,如果是,有什么问题,怎么解决?

- 答：是单例模式,所以在多线程访问的时候有线程安全问题,不要用同步,会影响性能的,解决方案是在控制器里面不能写字段。

工作原理

请描述 Spring MVC 的工作流程？描述一下

DispatcherServlet 的工作流程？

- (1) 用户发送请求至前端控制器 DispatcherServlet;
- (2) DispatcherServlet 收到请求后，调用 HandlerMapping 处理器映射器，请求获取 Handle；
- (3) 处理器映射器根据请求 url 找到具体的处理器，生成处理器对象及处理器拦截器(如果有则生成)一并返回给 DispatcherServlet；
- (4) DispatcherServlet 调用 HandlerAdapter 处理器适配器；



- (5) HandlerAdapter 经过适配调用 具体处理器(Handler, 也叫后端控制器);
- (6) Handler 执行完成返回 ModelAndView;
- (7) HandlerAdapter 将 Handler 执行结果 ModelAndView 返回给 DispatcherServlet;
- (8) DispatcherServlet 将 ModelAndView 传给 ViewResolver 视图解析器进行解析;
- (9) ViewResolver 解析后返回具体 View;
- (10) DispatcherServlet 对 View 进行渲染视图 (即将模型数据填充至视图中)
- (11) DispatcherServlet 响应用户。

MVC 框架

MVC 是什么? MVC 设计模式的好处有哪些

- mvc 是一种设计模式 (设计模式就是日常开发中编写代码的一种好的方法和经验的总结)。模型 (model) -视图 (view) -控制器 (controller), 三层架构的设计模式。用于实现前端页面的展现与后端业务数据处理的分离。
- mvc 设计模式的好处
 1. 分层设计, 实现了业务系统各个组件之间的解耦, 有利于业务系统的可扩展性, 可维护性。
 2. 有利于系统的并行开发, 提升开发效率。

常用注解



注解原理是什么

- 注解本质是一个继承了 Annotation 的特殊接口，其具体实现类是 Java 运行时生成的动态代理类。我们通过反射获取注解时，返回的是 Java 运行时生成的动态代理对象。通过代理对象调用自定义注解的方法，会最终调用 AnnotationInvocationHandler 的 invoke 方法。该方法会从 memberValues 这个 Map 中索引出对应的值。而 memberValues 的来源是 Java 常量池。

Spring MVC 常用的注解有哪些？

- @RequestMapping：用于处理请求 url 映射的注解，可用于类或方法上。用于类上，则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。
- @RequestBody：注解实现接收 http 请求的 json 数据，将 json 转换为 java 对象。
- @ResponseBody：注解实现将 controller 方法返回对象转化为 json 对象响应给客户。
- @Controller：控制器的注解，表示是表现层,不能用别的注解代替

SpringMvc 中的控制器的注解一般用哪个,有没有别的注解可以替代？

- 答：一般用@Controller 注解,也可以使用@RestController,@RestController 注解相当于@ResponseBody + @Controller,表示是表现层,除此之外，一般不用别的注解代替。



@Controller 注解的作用

- 在 Spring MVC 中，控制器 Controller 负责处理由 DispatcherServlet 分发的请求，它把用户请求的数据经过业务处理层处理之后封装成一个 Model，然后再把该 Model 返回给对应的 View 进行展示。在 Spring MVC 中提供了一个非常简便的定义 Controller 的方法，你无需继承特定的类或实现特定的接口，只需使用 @Controller 标记一个类是 Controller，然后使用 @RequestMapping 和 @RequestParam 等一些注解用以定义 URL 请求和 Controller 方法之间的映射，这样的 Controller 就能被外界访问到。此外 Controller 不会直接依赖于 HttpServletRequest 和 HttpServletResponse 等 HttpServletRequest 对象，它们可以通过 Controller 的方法参数灵活的获取到。
- @Controller 用于标记在一个类上，使用它标记的类就是一个 Spring MVC Controller 对象。分发处理器将会扫描使用了该注解的类的方法，并检测该方法是否使用了 @RequestMapping 注解。@Controller 只是定义了一个控制器类，而使用 @RequestMapping 注解的方法才是真正处理请求的处理器。单单使用 @Controller 标记在一个类上还不能真正意义上的说它就是 Spring MVC 的一个控制器类，因为这个时候 Spring 还不认识它。那么要如何做 Spring 才能认识它呢？这个时候就需要我们把这个控制器类交给 Spring 来管理。有两种方式：
 - 在 Spring MVC 的配置文件中定义 MyController 的 bean 对象。
 - 在 Spring MVC 的配置文件中告诉 Spring 该到哪里去找标记为 @Controller 的 Controller 控制器。

@RequestMapping 注解的作用



- RequestMapping 是一个用来处理请求地址映射的注解，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。
- RequestMapping 注解有六个属性，下面我们把她分成三类进行说明（下面有相应示例）。
 - **value, method**
 - value: 指定请求的实际地址，指定的地址可以是 URI Template 模式（后面将会说明）；
 - method: 指定请求的 method 类型， GET、POST、PUT、DELETE 等；
 - **consumes, produces**
 - consumes: 指定处理请求的提交内容类型（Content-Type），例如 application/json, text/html;
 - produces: 指定返回的内容类型，仅当 request 请求头中的(Accept)类型中包含该指定类型才返回；
 - **params, headers**
 - params: 指定 request 中必须包含某些参数值是，才让该方法处理。
 - headers: 指定 request 中必须包含某些指定的 header 值，才能让该方法处理请求。

@ResponseBody 注解的作用

- 作用: 该注解用于将 Controller 的方法返回的对象，通过适当的 HttpMessageConverter 转换为指定格式后，写入到 Response 对象的 body 数据区。
- 使用时机: 返回的数据不是 html 标签的页面，而是其他某种格式的数据时（如 json、xml 等）使用；

@PathVariable 和@RequestParam 的区别



- 请求路径上有个 id 的变量值，可以通过@PathVariable 来获取
@RequestMapping(value = "/page/{id}" , method = RequestMethod.GET)
- @RequestParam用来获得静态的 URL 请求入参 spring 注解时 action 里用到。

其他

Spring MVC 与 Struts2 区别

- 相同点
 - 都是基于 mvc 的表现层框架，都用于 web 项目的开发。
 - 不同点
1. 前端控制器不一样。Spring MVC 的前端控制器是 servlet: DispatcherServlet。struts2 的前端控制器是 filter: StrutsPreparedAndExcutorFilter。
 2. 请求参数的接收方式不一样。Spring MVC 是使用方法的形参接收请求的参数，基于方法的开发，线程安全，可以设计为单例或者多例的开发，推荐使用单例模式的开发（执行效率更高），默认就是单例开发模式。struts2 是通过类的成员变量接收请求的参数，是基于类的开发，线程不安全，只能设计为多例的开发。
 3. Struts 采用值栈存储请求和响应的数据，通过 OGNL 存取数据，Spring MVC 通过参数解析器是将 request 请求内容解析，并给方法形参赋值，将数据和视图封装成 ModelAndView 对象，最后又将 ModelAndView 中的模型数据通过 reques 域传输到页面。Jsp 视图解析器默认使用 jstl。
 4. 与 spring 整合不一样。Spring MVC 是 spring 框架的一部分，不需要整合。在企业项目中，Spring MVC 使用更多一些。



Spring MVC 怎么样设定重定向和转发的?

- (1) 转发: 在返回值前面加"forward:", 譬如
"forward:user.do?name=method4"
- (2) 重定向: 在返回值前面加"redirect:", 譬如"redirect:www.baidu.com"

Spring MVC 怎么和 AJAX 相互调用的?

- 通过 Jackson 框架就可以把 Java 里面的对象直接转化成 Js 可以识别的 Json 对象。具体步骤如下 :
- (1) 加入 Jackson.jar
- (2) 在配置文件中配置 json 的映射
- (3) 在接受 Ajax 方法里面可以直接返回 Object,List 等,但方法前面要加上 @ResponseBody 注解。

如何解决 POST 请求中文乱码问题, GET 的又如何处理呢?

- (1) 解决 post 请求乱码问题:
- 在 web.xml 中配置一个 CharacterEncodingFilter 过滤器, 设置成 utf-8;

```
<filter>
    <filter-name>CharacterEncodingFilter</filter-name>
```



```
<filter-class>org.springframework.web.filter.CharacterEncodingFilter</fi
lter-class>
<init-param>
    <param-name>encoding</param-name>
    <param-value>utf-8</param-value>
</init-param>
</filter>

<filter-mapping>
    <filter-name>CharacterEncodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

- (2) get 请求中文参数出现乱码解决方法有两个：
- 1、修改 tomcat 配置文件添加编码与工程编码一致，如下：

```
<ConnectorURIEncoding="utf-8" connectionTimeout="20000"
port="8080" protocol="HTTP/1.1" redirectPort="8443"/>
```

- 2、另外一种方法对参数进行重新编码：
String userName = new
String(request.getParamter("userName").getBytes("ISO8859-1"),"utf-8")
ISO8859-1 是 tomcat 默认编码，需要将 tomcat 编码后的内容按 utf-8 编码。

Spring MVC 的异常处理？



- 答：可以将异常抛给 Spring 框架，由 Spring 框架来处理；我们只需要配置简单的异常处理器，在异常处理器中添视图页面即可。

如果在拦截请求中，我想拦截 get 方式提交的方法,怎么配置

- 答：可以在 @RequestMapping 注解里面加上 method=RequestMethod.GET。

怎样在方法里面得到 Request,或者 Session?

- 答：直接在方法的形参中声明 request, Spring MVC 就自动把 request 对象传入。

如果想在拦截的方法里面得到从前台传入的参数,怎么得到?

- 答：直接在形参里面声明这个参数就可以,但必须名字和传过来的参数一样。

如果前台有很多个参数传入,并且这些参数都是一个对象的,那么

怎么样快速得到这个对象?

- 答：直接在方法中声明这个对象, Spring MVC 就自动会把属性赋值到这个对象里面。



Spring MVC 中函数的返回值是什么？

- 答：返回值可以有很多类型,有 String, ModelAndView。ModelAndView 类把视图和数据都合并的一起的，但一般用 String 比较好。

Spring MVC 用什么对象从后台向前台传递数据的？

- 答：通过 ModelMap 对象,可以在这个对象里面调用 put 方法,把对象加到里面,前台就可以通过 el 表达式拿到。

怎么样把 ModelMap 里面的数据放入 Session 里面？

- 答：可以在类上面加上 @SessionAttributes 注解,里面包含的字符串就是要放入 session 里面的 key。

Spring MVC 里面拦截器是怎么写的

- 有两种写法,一种是实现 HandlerInterceptor 接口，另外一种继承适配器类，接着在接口方法当中，实现处理逻辑；然后在 Spring MVC 的配置文件中配置拦截器即可：



```
<!-- 配置 Spring MVC 的拦截器 -->
<mvc:interceptors>
    <!-- 配置一个拦截器的 Bean 就可以了 默认是对所有请求都拦截 -->
    <bean id="myInterceptor"
class="com.zwp.action.MyHandlerInterceptor"></bean>
    <!-- 只针对部分请求拦截 -->
    <mvc:interceptor>
        <mvc:mapping path="/modelMap.do" />
        <bean class="com.zwp.action.MyHandlerInterceptorAdapter" />
    </mvc:interceptor>
</mvc:interceptors>
```

介绍一下 **WebApplicationContext**

- **WebApplicationContext** 继承了 **ApplicationContext** 并增加了一些 WEB 应用必备的特有功能, 它不同于一般的 **ApplicationContext** , 因为它能处理主题, 并找到被关联的 **servlet**。