

12 | 再回首：如何实现Spring MVC?

2023-04-07 郭屹 来自北京

《手把手带你写一个MiniSpring》



你好，我是郭屹。

恭喜你学完 MiniSpring 的第二部分——MVC 了。你是不是跟着我的脚步自己也实现了一个 MVC 呢？如果已经完成了，欢迎你把你的实现代码放到评论区，我们一起交流讨论。

为了让你更好地掌握这部分内容，我们来对这一整章做一个重点回顾。

MVC 重点回顾

MVC 是 Spring 框架的核心组成部分之一，负责实现一个 Web 应用程序的模型视图控制器模式。Spring 框架提供了丰富的组件和工具，而 MVC 负责处理一个 Web 应用程序中的核心过程，例如请求处理、数据转换、模板渲染和视图管理等。而 MVC 和 Spring 的结合，就像是车与引擎的结合一样，给 Web 应用程序提供了强大而且可靠的性能和灵活性，让我们能够快速、方便地搭建高性能、可靠的 Web 应用程序。

因为 Spring 是 Java 实现的，也因为发明人 Rod Johnson 先生自己是 Java 团队 Servlet 规范的专家组成员，所以很自然，他的 MVC 的实现是基于 Servlet 的。利用 Servlet 的机制，把这种功效发挥到了极致，快速构造了完整的 Web 程序结构，让我们大开眼界。

那我们在课程中是怎么实现 MVC 的呢？

首先我们利用 Servlet 机制，用一个单一的 Servlet 拦截所有请求，由它来分派任务，这样实现了原始的 MVC 结构。然后呢，我们把 MVC 和 IoC 结合在一起，在 Servlet 容器启动的时候，给上下文环境里注入 IoC 容器，使得在 Servlet 里可以访问到 IoC 容器里的 Bean。

之后我们进一步解耦 MVC 结构，独立出请求处理器，还用了一个简洁统一的注解方式，把 Web 请求方便地定位到后台处理的类和方法里，实现 Spring 的 RequestHandler。

在前后台打通的时候，实现数据参数的自动转换，也就是说先把 Web 请求的传入参数，自动地从文本转换成对象，实现数据绑定功能。对于返回数据，也自动根据用户需求进行格式化转换，这样实现了 Spring 里面的 data binder 和 data conversion。最后回到前端 View，如果有前端引擎，在 Spring 中引用，把数据自动渲染到前端。

我们可以利用 Servlet 机制、MVC 结构、IoC 容器、RequestHandler 和数据绑定等功能，确保前后台的有效沟通和良好的交互体验，实现一个高效可靠的 Web 应用程序。你学会了吗？

最后，我们再来看一下这一章的几道思考题，我在文稿里都给出了参考答案。我建议你也可以把你的思考分享在留言区，看看自己的思路是不是对的。题目我也就不一道一道地说了，你在做完思考题之后，自己来对比。

07 | 原始 MVC：如何通过单一的 Servlet 拦截请求分派任务？

思考题

我们在 MVC 中也使用了 Bean 这个概念，它跟我们以前章节中的 Bean 是什么关系？

参考答案

MVC (Model-View-Controller) 和 IoC (Inversion of Control) 是两个不同的设计模式，它们都使用 “Bean” 这个概念，但是在不同的层级和实现方式上有所不同。

在 MVC 中，“Bean” 通常指代模型对象。模型对象是业务逻辑层的核心，用于实现数据访问和业务逻辑处理等功能。在 MVC 中，模型对象通常是由控制器 (Controller) 创建并向视图 (View) 传递的。

在 IoC 中，“Bean” 指代由 IoC 容器管理的对象。IoC 容器负责创建及管理应用程序中的所有 Bean 对象。通过 IoC 容器，应用程序能够实现“控制反转”，即由 IoC 容器统一管理和调度应用程序中的各个组件。在 IoC 中，Bean 是由 IoC 容器创建、初始化、配置和装配的。

因此，尽管 MVC 和 IoC 都使用了 “Bean” 这个概念，但它们的含义及在系统中的作用是不同的。MVC 中的 Bean 一般是 Web 相关的业务逻辑，IoC 中的 Bean 可能是一些更加基础性的逻辑。从 MVC 中可以访问到 IoC 容器中的 Bean。

08 | 整合 IoC 和 MVC：如何在 Web 环境中启动 IoC 容器？

思考题

我们看到从 Dispatcher 里可以访问 WebApplicationContext 里管理的 Bean，那通过 WebApplicationContext 可以访问 Dispatcher 内管理的 Bean 吗？

参考答案

不可以。

Servlet 容器启动的时候，按照时序，是先启动 Listener，在 Listener 的初始化过程中创建 IoC 容器，放到 ServletContext 里，这就是 WAC。这之后再初始化的 Servlet。所以 Dispatcher 可以访问到 WAC，但是 WAC 访问不到 DispatcherServlet，这个是单向的。

09 | 分解 Dispatcher：如何把专门的事情交给专门的部件去做？

思考题

目前，我们只支持了 GET 方法，你能不能尝试自己增加 POST 方法。想一想，需要改变现有的程序结构吗？

参考答案

增加 POST 方法支持不需要改变现有程序结构。因为我们的 DispatcherServlet 现在统一用 service() 方法处理所有请求，之后调用 doDispatch() 方法，最后通过 this.handlerMapping.getHandler() 找到需要调用的方法。无论对于 GET 还是 POST，都是同样的流程，统一由 handlerMapping 来区分不同的调用方法。

所以如果要区分 GET 和 POST，则可以在 RequestMapping 注解上增加 METHOD 属性，表示 GET 还是 POST，然后 handlerMapping.getHandler() 中根据 GET 和 POST 匹配实际的调用方法。

10 | 数据绑定：如何自动转换传入的参数？

思考题

我们现在的实现是把 request 里面的参数值，按照内部的次序隐含地自动转成后台调用方法参数对象中的某个属性值，那么可不可以使用一个手段，让程序员手动指定某个调用方法的参数跟哪个 request 参数进行绑定呢？

参考答案

参数绑定的处理，是在 RequestMappingHandlerAdatper 的 invokeHandlerMethod() 方法中处理的，它拿到调用方法的所有参数，一个参数一个参数进行绑定：`WebDataBinder wdb = binderFactory.createBinder(request, methodParamObj, methodParameter.getName());`。所以我们在这里可以考虑给参数增加一个注解 `@RequestParam`。对于带有这个注解的参数，就不是隐含地按照参数名去匹配，而是按照指定的名字去 request 中匹配。

11 | ModelAndView：如何将处理结果返回到前端？

现在返回的数据只支持 Date、Number 和 String 三种类型，如何扩展到更多的数据类型？现在也只支持 JSP，如何扩展到别的前端？

参考答案

返回数据的格式处理是通过 `ObjectMapper` 来实现的。我们有一个默认实现 `DefaultObjectMapper`，只要在它的 `writeValuesAsString()` 里判断数据类型的时候，增加别的类型就可以了。

对于 JSP 之外的 View，我们现在的结构是可扩展的。只要自己另外实现一个 View 和一个 View resolver 即可。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 (4)



Geek_320730

2023-04-11 来自北京

<https://github.com/kuifir/practice/tree/master/miniSpring> 跟着老师每门课程跟着学完都很有收获，完成一个课程的内容并且测试成功后也很有成就感。

作者回复：动手是诀窍，事情要经过手才行。



欧阳利

2023-04-11 来自广东

怎么感觉越到后面，内容越简单了

作者回复：这就是我希望的效果，跟一遍之后就懂了Spring框架。



peter

2023-04-08 来自北京

请教老师几个问题：

Q1：SpringMVC与Spring是什么关系？

SpringMVC是Spring的一个子模块或子组件吗？或者两者不是包含关系，是平等、相互独立的

两个东西？

Q2: SpringMVC是基于servlet，可以基于其他的吗？

Q3: SpringMVC需要依赖于一个容器，比如tomcat，但spring不需要依赖tomcat，是这样吗？

Q4: servlet具体是一个进程还是一个线程？ Redis速度快，因为是单线程，现在servlet处理全部请求，是不是类似于redis的单线程？

Q5: 前后盾分离开发，前端用vue，应该会有一个“vue引擎”一类的东西。前端开发完成后，页面会打包放到后端idea项目中，那么，此时，后端会有一个“vue引擎”吗？

Q6: wac访问不到servlet，是不需要吗？还是技术上不可行？可以把wac放到ServletContext中，为什么不可以把DispatchServlet放到wac中？

作者回复：一并回答如下。spring mvc是spring的一个模块。它的实现是用的servlet，也可以使用filter来实现的。Spring mvc是web框架，所以要在web容器具体来讲是servlet容器如Tomcat中运行。Spring是一个大体系，狭义的核心Spring IoC是不依赖于Tomcat的。servlet是一个线程，不是单线程的。前后端分离了，后端不需要vue引擎的。wac和dispatcherservlet的关系是上下游的关系或者说是上下层的关系，软件体系结构是一层层的，一般是上层访问下层。Visual Basic的发明人说过：软件是一个地质学堆积过程。



Jay

2023-04-07 来自美国

请教老师，有个问题一直没想明白，RequestMappingHandlerMapping和RequestMappingHandlerAdapter都是配置在applicationContext.xml里面由IOC容器管理的，DispatcherServlet中通过wac.getBean()方法获取，这一步没有问题。但是，RequestMappingHandlerMapping依赖的WebApplicationContext必须是MVC容器（initMapping方法中需要获取MVC容器中维护的controllerNames），我一直没有看到RequestMappingHandlerMapping依赖的这个wac是如何注入的。master代码里面实现了ApplicationContextAware接口，但是我想即便实现了这里也只能得到IOC容器对应的ApplicationContext，好像依然不能解决这个问题？不知道到我理解的有没有问题？

作者回复：refresh的时候初始化这两个东西：

```
protected void Refresh() {  
    initHandlerMappings(this.webApplicationContext);  
    initHandlerAdapters(this.webApplicationContext);  
    initViewResolvers(this.webApplicationContext);  
}
```

而这个wac是在dispatcherservlet init的时候创建的：

```
public void init(ServletConfig config) throws ServletException {
```

```
super.init(config);
```

```
this.parentApplicationContext =  
    (WebApplicationContext) this.getServletContext().getAttribute(WebApplicationCont  
ext.ROOT_WEB_APPLICATION_CONTEXT_ATTRIBUTE);
```

```
this.sContextConfigLocation = config.getInitParameter("contextConfigLocation");
```

```
this.webApplicationContext = new AnnotationConfigWebApplicationContext(this.sContextConfi  
gLocation,this.parentApplicationContext);
```

