

03 | 你应该知道的Servlet规范和Servlet容器

2019-05-16 李号双

深入拆解Tomcat & Jetty

[进入课程 >](#)



讲述：李号双

时长 10:39 大小 9.77M



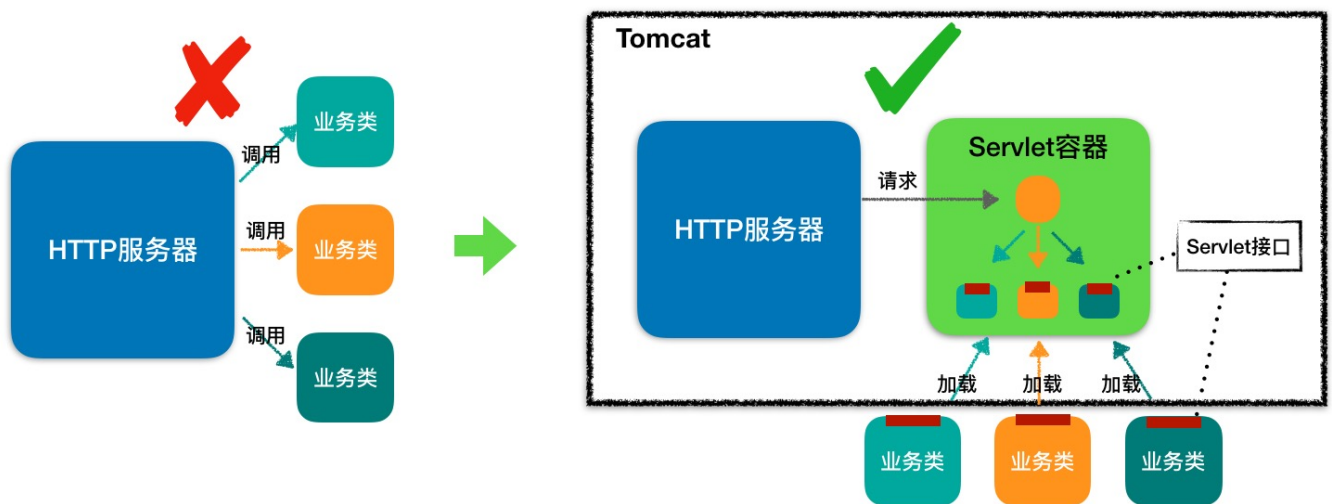
通过专栏上一期的学习我们知道，浏览器发给服务端的是一个 HTTP 格式的请求，HTTP 服务器收到这个请求后，需要调用服务端程序来处理，所谓的服务端程序就是你写的 Java 类，一般来说不同的请求需要由不同的 Java 类来处理。

那么问题来了，HTTP 服务器怎么知道要调用哪个 Java 类的哪个方法呢。最直接的做法是在 HTTP 服务器代码里写一大堆 if else 逻辑判断：如果是 A 请求就调 X 类的 M1 方法，如果是 B 请求就调 Y 类的 M2 方法。但这样做明显有问题，因为 HTTP 服务器的代码跟业务逻辑耦合在一起了，如果新加一个业务方法还要改 HTTP 服务器的代码。

那该怎么解决这个问题呢？我们知道，面向接口编程是解决耦合问题的法宝，于是有一伙人就定义了一个接口，各种业务类都必须实现这个接口，这个接口就叫 Servlet 接口，有时我们也把实现了 Servlet 接口的业务类叫作 Servlet。

但是这里还有一个问题，对于特定的请求，HTTP 服务器如何知道由哪个 Servlet 来处理呢？Servlet 又是由谁来实例化呢？显然 HTTP 服务器不适合做这个工作，否则又和业务类耦合了。

于是，还是那伙人又发明了 Servlet 容器，Servlet 容器用来加载和管理业务类。HTTP 服务器不直接跟业务类打交道，而是把请求交给 Servlet 容器去处理，Servlet 容器会将请求转发到具体的 Servlet，如果这个 Servlet 还没创建，就加载并实例化这个 Servlet，然后调用这个 Servlet 的接口方法。因此 Servlet 接口其实是**Servlet 容器跟具体业务类之间的接口**。下面我们通过一张图来加深理解。



图的左边表示 HTTP 服务器直接调用具体业务类，它们是紧耦合的。再看图的右边，HTTP 服务器不直接调用业务类，而是把请求交给容器来处理，容器通过 Servlet 接口调用业务类。因此 Servlet 接口和 Servlet 容器的出现，达到了 HTTP 服务器与业务类解耦的目的。

而 Servlet 接口和 Servlet 容器这一整套规范叫作 Servlet 规范。Tomcat 和 Jetty 都按照 Servlet 规范的要求实现了 Servlet 容器，同时它们也具有 HTTP 服务器的功能。作为 Java 程序员，如果我们要实现新的业务功能，只需要实现一个 Servlet，并把它注册到 Tomcat (Servlet 容器) 中，剩下的事情就由 Tomcat 帮我们处理了。

接下来我们来看看 Servlet 接口具体是怎么定义的，以及 Servlet 规范又有哪些要重点关注的地方呢？

Servlet 接口

Servlet 接口定义了下面五个方法：

```
1 public interface Servlet {  
2     void init(ServletConfig config) throws ServletException;  
3  
4     ServletConfig getServletConfig();  
5  
6     void service(ServletRequest req, ServletResponse res) throws ServletException, IOException;  
7  
8     String getServletInfo();  
9  
10    void destroy();  
11 }
```

其中最要的是 `service` 方法，具体业务类在这个方法里实现处理逻辑。这个方法有两个参数：`ServletRequest` 和 `ServletResponse`。`ServletRequest` 用来封装请求信息，`ServletResponse` 用来封装响应信息，因此**本质上这两个类是对通信协议的封装**。

比如 HTTP 协议中的请求和响应就是对应了 `HttpServletRequest` 和 `HttpServletResponse` 这两个类。你可以通过 `HttpServletRequest` 来获取所有请求相关的信息，包括请求路径、Cookie、HTTP 头、请求参数等。此外，我在专栏上一期提到过，我们还可以通过 `HttpServletRequest` 来创建和获取 Session。而 `HttpServletResponse` 是用来封装 HTTP 响应的。

你可以看到接口中还有两个跟生命周期有关的方法 `init` 和 `destroy`，这是一个比较贴心的设计，Servlet 容器在加载 Servlet 类的时候会调用 `init` 方法，在卸载的时候会调用 `destroy` 方法。我们可能会在 `init` 方法里初始化一些资源，并在 `destroy` 方法里释放这些资源，比如 Spring MVC 中的 `DispatcherServlet`，就是在 `init` 方法里创建了自己的 Spring 容器。

你还会注意到 `ServletConfig` 这个类，`ServletConfig` 的作用就是封装 Servlet 的初始化参数。你可以在 `web.xml` 给 Servlet 配置参数，并在程序里通过 `getServletConfig` 方法拿到这些参数。

我们知道，有接口一般就有抽象类，抽象类用来实现接口和封装通用的逻辑，因此 Servlet 规范提供了 `GenericServlet` 抽象类，我们可以通过扩展它来实现 Servlet。虽然 Servlet 规范并不在乎通信协议是什么，但是大多数的 Servlet 都是在 HTTP 环境中处理的，因此 Servlet 规范还提供了 `HttpServlet` 来继承 `GenericServlet`，并且加入了 HTTP 特性。这样

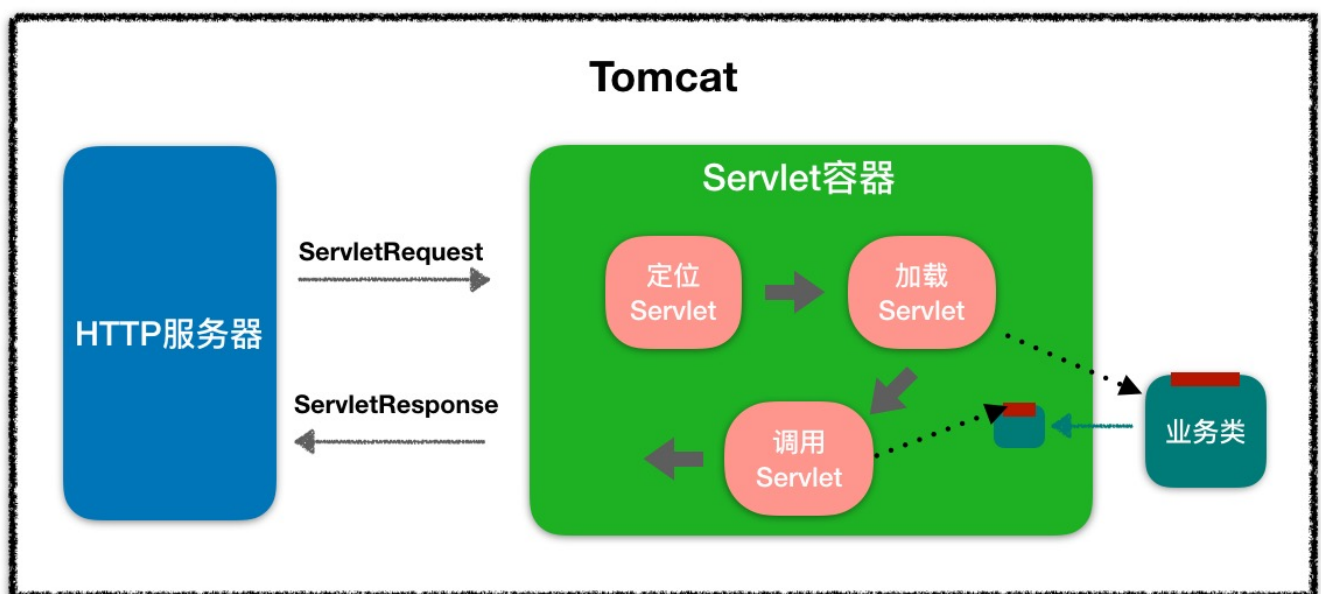
我们通过继承 `HttpServlet` 类来实现自己的 `Servlet`，只需要重写两个方法：`doGet` 和 `doPost`。

Servlet 容器

我在前面提到，为了解耦，HTTP 服务器不直接调用 `Servlet`，而是把请求交给 `Servlet` 容器来处理，那 `Servlet` 容器又是如何工作的呢？接下来我会介绍 `Servlet` 容器大体的工作流程，一起来聊聊我们非常关心的两个话题：**Web 应用的目录格式是什么样的，以及我该如何扩展和定制化 `Servlet` 容器的功能。**

工作流程


当客户请求某个资源时，HTTP 服务器会用一个 `ServletRequest` 对象把客户的请求信息封装起来，然后调用 `Servlet` 容器的 `service` 方法，`Servlet` 容器拿到请求后，根据请求的 URL 和 `Servlet` 的映射关系，找到相应的 `Servlet`，如果 `Servlet` 还没有被加载，就用反射机制创建这个 `Servlet`，并调用 `Servlet` 的 `init` 方法来完成初始化，接着调用 `Servlet` 的 `service` 方法来处理请求，把 `ServletResponse` 对象返回给 HTTP 服务器，HTTP 服务器会把响应发送给客户端。同样我通过一张图来帮助你理解。



Web 应用

`Servlet` 容器会实例化和调用 `Servlet`，那 `Servlet` 是怎么注册到 `Servlet` 容器中的呢？一般来说，我们是以 Web 应用程序的方式来部署 `Servlet` 的，而根据 `Servlet` 规范，Web 应

用程序有一定的目录结构，在这个目录下分别放置了 Servlet 的类文件、配置文件以及静态资源，Servlet 容器通过读取配置文件，就能找到并加载 Servlet。Web 应用的目录结构大概是下面这样的：

 复制代码

```
1 | - MyWebApp
2   | - WEB-INF/web.xml      -- 配置文件，用来配置 Servlet 等
3   | - WEB-INF/lib/         -- 存放 Web 应用所需各种 JAR 包
4   | - WEB-INF/classes/     -- 存放你的应用类，比如 Servlet 类
5   | - META-INF/            -- 目录存放工程的一些信息
```

Servlet 规范里定义了**ServletContext**这个接口来对应一个 Web 应用。Web 应用部署好后，Servlet 容器在启动时会加载 Web 应用，并为每个 Web 应用创建唯一的 ServletContext 对象。你可以把 ServletContext 看成是一个全局对象，一个 Web 应用可能有多个 Servlet，这些 Servlet 可以通过全局的 ServletContext 来共享数据，这些数据包括 Web 应用的初始化参数、Web 应用目录下的文件资源等。由于 ServletContext 持所有 Servlet 实例，你还可以通过它来实现 Servlet 请求的转发。

扩展机制

不知道你有没有发现，引入了 Servlet 规范后，你不需要关心 Socket 网络通信、不需要关心 HTTP 协议，也不需要关心你的业务类是如何被实例化和调用的，因为这些都被 Servlet 规范标准化了，你只要关心怎么实现的你的业务逻辑。这对于程序员来说是件好事，但也有不方便的一面。所谓规范就是说大家都要遵守，就会千篇一律，但是如果这个规范不能满足你的业务的个性化需求，就有问题了，因此设计一个规范或者一个中间件，要充分考虑到可扩展性。Servlet 规范提供了两种扩展机制：**Filter**和**Listener**。

Filter是过滤器，这个接口允许你对请求和响应做一些统一的定制化处理，比如你可以根据请求的频率来限制访问，或者根据国家地区的不同来修改响应内容。过滤器的工作原理是这样的：Web 应用部署完成后，Servlet 容器需要实例化 Filter 并把 Filter 链接成一个 FilterChain。当请求进来时，获取第一个 Filter 并调用 doFilter 方法，doFilter 方法负责调用这个 FilterChain 中的下一个 Filter。

Listener是监听器，这是另一种扩展机制。当 Web 应用在 Servlet 容器中运行时，Servlet 容器内部会不断的发生各种事件，如 Web 应用的启动和停止、用户请求到达等。Servlet

容器提供了一些默认的监听器来监听这些事件，当事件发生时，Servlet 容器会负责调用监听器的方法。当然，你可以定义自己的监听器去监听你感兴趣的事件，将监听器配置在 web.xml 中。比如 Spring 就实现了自己的监听器，来监听 ServletContext 的启动事件，目的是当 Servlet 容器启动时，创建并初始化全局的 Spring 容器。

到这里相信你对 Servlet 容器的工作原理有了深入的了解，只有理解了这些原理，我们才能更好的理解 Tomcat 和 Jetty，因为它们都是 Servlet 容器的具体实现。后面我还会详细谈到 Tomcat 和 Jetty 是如何设计和实现 Servlet 容器的，虽然它们的实现方法各有特点，但是都遵守了 Servlet 规范，因此你的 Web 应用可以在这两个 Servlet 容器中方便的切换。

本期精华

今天我们学习了什么是 Servlet，回顾一下，Servlet 本质上是一个接口，实现了 Servlet 接口的业务类也叫 Servlet。Servlet 接口其实是 Servlet 容器跟具体 Servlet 业务类之间的接口。Servlet 接口跟 Servlet 容器这一整套规范叫作 Servlet 规范，而 Servlet 规范使得程序员可以专注业务逻辑的开发，同时 Servlet 规范也给开发者提供了扩展的机制 Filter 和 Listener。

最后我给你总结一下 Filter 和 Listener 的本质区别：

Filter 是干预过程的，它是过程的一部分，是基于过程行为的。

Listener 是基于状态的，任何行为改变同一个状态，触发的事件是一致的。

课后思考

Servlet 容器与 Spring 容器有什么关系？

不知道今天的内容你消化得如何？如果还有疑问，请大胆的在留言区提问，也欢迎你把你的课后思考和心得记录下来，与我和其他同学一起讨论。如果你觉得今天有所收获，欢迎你把它分享给你的朋友。

深入拆解 Tomcat & Jetty

从源码角度深度探索 Java 中间件

李号双

eBay 技术主管



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 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。

上一篇 02 | HTTP协议必知必会

下一篇 04 | 实战：纯手工打造和运行一个Servlet

精选留言 (69)

写留言



天娜 置顶

2019-05-16

65

文章中提到

- 1.SpringMVC 容器实在DispatcherServlet中init方法里创建的。
- 2.Spring 容器是通过Listener创建的
 - a、就是说SpringMVC容器和Spring容器还不一样，那么他们是什么关系？
 - b、他们和Servlet容器又是啥关系？

展开

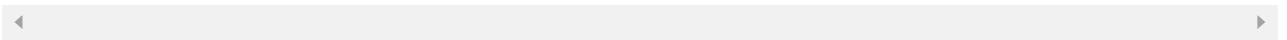
作者回复: Tomcat&Jetty在启动时给每个Web应用创建一个全局的上下文环境，这个上下文就是ServletContext，其为后面的Spring容器提供宿主环境。

Tomcat&Jetty在启动过程中触发容器初始化事件，Spring的ContextLoaderListener会监听到这

个事件，它的contextInitialized方法会被调用，在这个方法中，Spring会初始化全局的Spring根容器，这个就是Spring的IoC容器，IoC容器初始化完毕后，Spring将其存储到ServletContext中，便于以后来获取。

Tomcat&Jetty在启动过程中还会扫描Servlet，一个Web应用中的Servlet可以有多个，以SpringMVC中的DispatcherServlet为例，这个Servlet实际上是一个标准的前端控制器，用以转发、匹配、处理每个Servlet请求。

Servlet一般会延迟加载，当第一个请求达到时，Tomcat&Jetty发现DispatcherServlet还没有被实例化，就调用DispatcherServlet的init方法，DispatcherServlet在初始化的时候会建立自己的容器，叫做SpringMVC容器，用来持有Spring MVC相关的Bean。同时，Spring MVC还会通过ServletContext拿到Spring根容器，并将Spring根容器设为SpringMVC容器的父容器，请注意，Spring MVC容器可以访问父容器中的Bean，但是父容器不能访问子容器的Bean，也就是说Spring根容器不能访问SpringMVC容器里的Bean。说的通俗点就是，在Controller里可以访问Service对象，但是在Service里不可以访问Controller对象。



0.618

2019-05-16

15

笔记总结：

- 1.Servlet规范：Servlet和Servlet容器的一整套规则
- 2.Servlet和Servlet的出现是为了解耦http服务器和业务逻辑
- 3.HttpServletRequest和HttpServletResponse对象是对通信协议的封装
- 4.Servlet接口有5个方法，其中包括生命周期函数两个：init和destroy；获取信息的函数...

展开



一路远行

2019-05-16

15

spring容器只是servlet容器上下文(ServletContext)的一个属性，web容器启动时通过ServletContextListener机制构建出来



石头狮子

2019-05-16

6

servlet 容器抽象了网络处理，请求封装等事情，同样提供了可以处理其他非 http 协议的能力。

spring 容器是依赖注入设计模式的体现，其主要抽象了类初始化，注入，依赖解决，动态

代理等功能。
两者主要解决的问题不同。

展开 ▾

作者回复: 我理解你说的Servlet容器应该是说HTTP服务器+Servlet容器, Servlet容器本身只管Servlet的事, 不管HTTP协议的解析



Monday

2019-05-17

👍 5

基于思考题, 我在梦中醒来, 觉得servlet容器管理的是servlet (把controller也理解成了servlet), spring容器则是管理服务, DAO这类bean。这样理解的话springMVC不就是多余的吗? 但是我们项目中都有使用springMVC, 存在即合理, 所以我的理解是有误的。于是想老师帮忙给出以下三张图。非常感谢,

1, 恳求老师能给出servlet, spring, springMVC三个容器的关系图。...

展开 ▾

作者回复: 本来想自己画一张, 但是在网上找了一下, 找到这张图, 蛮清楚的:

<https://blog.csdn.net/zhanglf02/article/details/89791797>



QQ怪

2019-05-16

👍 4

Spring容器是管理服务和服务dao的,

SpringMVC容器是管理controller对象的,

...

展开 ▾



刘三通

2019-05-19

👍 4

servlet容器初始化成功后被spring监听, 创建spring容器放入servlet容器中, 访问到达, 初始化dispatcher servlet时创建springmvc容器, 通过servletContext拿到spring容器,

并将其作为自己的父容器，spring mvc容器会定义controller相关的bean,spring会定义业务逻辑相关的bean



neohope

2019-05-19

👍 3

Servlet容器，是用于管理Servlet生命周期的。
Spring容器，是用于管理Spring Bean生命周期的。
SpringMVC容器，适用于管理SpringMVC Bean生命周期的。

Tomcat/Jetty启动，对于每个WebApp，依次进行初始化工作：...

展开 ∨

作者回复: DispatcherServlet中的成员变量都是初始化好后就不会被改变了，所以是线程安全的，那“可见性”怎么保证呢？

这是由Web容器比如Tomcat来做到的，Tomcat在调用Servlet的init方法时，用了synchronized。

```
private synchronized void initServlet(Servlet servlet)
{...}
```



李青

2019-05-21

👍 2

老师给你个大大的赞，讲课方式是很喜欢的风格，先抛出问题，再讲思路和解决方案。以后讲课希望一直用这个风格

作者回复: 😊，谢谢



-W.LI-

2019-05-18

👍 2

老师好!我看留言里有同学说，spring上下文负责创建service和dao的bean，MVC负责创建controller的bean。我们平时说的IOC容器是指哪个啊?还有就是controller注解是一个组合注解，我在controller上用service注解一样能注册成功，spring和MVC容器又是怎么区分这个bean是controller还是service,或者是dao，bean的?还是我完全理解错了。

展开 ▾

作者回复: SpringBoot中只有一个Spring上下文:

`org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplication`

你可以在`FrameworkServlet.initWebApplicationContext`方法中下个断点。



allean

2019-05-16

👍 2

有收获，跟着专栏一步步深入

展开 ▾



仙道

2019-05-20

👍 1

servlet容器的Filter跟 spring 的interceptor 有啥区别

展开 ▾

作者回复: Filter是Servlet规范的一部分，是Servlet容器Tomcat实现的。Interceptor是Spring发明的。它们的执行顺序是：

`Filter.doFilter();`

`HandlerInterceptor.preHandle();`

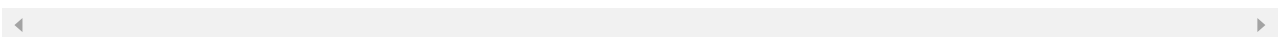
Controller

`HandlerInterceptor.postHandle();`

DispatcherServlet 渲染视图

`HandlerInterceptor.afterCompletion();`

`Filter.doFilter();` Servlet方法返回





非洲铜

2019-05-18

👍 1

能感受到李老师写得很用心👍

展开 ▾

作者回复: 谢谢!



Geek_ebda9...

2019-05-17

👍 1

老师, spring容器指的是spring本身的ioc容器吧, 是用来管理所有的bean, servlet本身会把spring的容器设置到上下文中, 而spring mvc的容器dispatch servlet相当于是一个具体的servlet的实现, 然后会创建一个全局的上下文application context spring的ioc容器会注入到这个上下文中, 后面通过上下文getbean, 其实是先找到上下文中的ioc容器, 然后再从这个容器拿到具体的bean, 这是不是对的?

展开 ▾

作者回复: 不太准确哦, 首先我们明确一点, Spring和SpringMVC分别有自己的IOC容器或者说上下文。

为什么要分成两个容器呢? 为了职责划分清晰。

SpringMVC的容器直接管理跟DispatcherServlet相关的Bean, 也就是Controller, ViewResolver等, 并且SpringMVC容器是在DispatcherServlet的init方法里创建的。而Spring容器管理其他的Bean比如Service和DAO。

并且SpringMVC容器是Spring容器的子容器, 所谓的父子关系意味着什么呢, 就是你通过子容器去拿某个Bean时, 子容器先在自己管理的Bean中去找这个Bean, 如果找不到再到父容器中找。但是父容器不能到子容器中去寻找某个Bean。

其实这个套路跟JVM的类加载器设计有点像, 不同的类加载器也为了隔离, 不过加载顺序是反的, 子加载器总是先委托父加载器去加载某个类, 加载不到再自己来加载。



DFighting

2019-05-17

👍 1

提个建议吧, 这些知识以前也都看过, 博客, 书本等等, 老师讲的比较系统, 看起来也有

不少收获，但是没机会实践验证和加深自己的理解是最难受的，上课的时候也实现过简单是Servlet，但都不成体系，也很难和实际结合起来，希望老师在后面中高级模块解析的时候能不能在一开始准备一个经典的项目，一边实践一遍学习我觉得会好很多。

展开 ∨

作者回复: 完全理解，你可以把Tomcat和Jetty本身当成一个项目，源码跑起来，再结合专栏的讲解，理解Java技术该怎么用



王鹏飞Tbb

2019-05-17

👍 1

老师好，请问下http服务器指的是什么？跟tomcat什么关系？

展开 ∨

作者回复: 你可以把HTTP服务器理解为Tomcat的一部分。



蔡伶

2019-05-17

👍 1

打卡

给老师提个建议：介绍下servlet规范和spring的历史起源演变过程，这样大家对于技术的发展脉络就会有更准确的认识。

本课程理解：

http服务，处理http协议网络连接的事情。...

展开 ∨

作者回复: 谢谢建议，前面的基础安排了三篇，由于篇幅有限，我尽量用简单的语言来解释为什么需要Servlet，Servlet是如何设计的，为什么要这样设计，也就是思考方式把，但是Servlet的很多细节还需要课后学习一下的。



nsn_huang

2019-05-16

👍 1

单独使用servlet时，servlet容器是根据url等信息，选择将请求交给哪个servlet来处理。使用springMVC时，servlet容器将请求交给springMVC的DispatchServlet来处理。我认为二者是合作关系。

展开 ▾



李海明

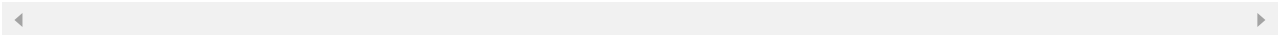
2019-05-16

👍 1

spring容器中还包含许多的子容器，其中springmvc容器就是其中常用的一个，文中的DispatcherServlet就是springmvc容器中的servlet接口，也是springmvc容器的核心类。spring容器主要用于整个Web应用程序需要共享的一些组件，比如DAO、数据库的ConnectionFactory等,springmvc的容器主要用于和该Servlet相关的一些组件,比如Controller、ViewResovler等。至此就清楚了spring容器内部的关系，那servlet容器跟...

展开 ▾

作者回复: 对的，说的很好



小胖狗

2019-05-16

👍 1

业务类也是Servlet,Servlet容器是用来加载和运行Servlet的。Spring容器就是提供业务类Servlet中使用的各种bean的加载，依赖关系，以及注入。