Atitit webserver web服务器的艺术

目录

[1.1. 2.2 使用处理器处理请求 1](#_Toc22352)

[2. 2.5 处理器的作用域 : 2](#_Toc19471)

[2.1. 在Jetty中，很多标准的服务器会继承HandlerWrappers，用来进行链式处理，比如从请求从 ContextHandler 到 SessionHandler ，再到 SecurityHandler 最后到ServletHandler 2](#_Toc14857)

[2.2. 2.6 嵌入 ResourceHandle 来处理当前工作路径下的静态资源 2](#_Toc24808)

[2.3. 2.8 ServletHandle 2](#_Toc22933)

[2.4. 2.9 嵌入 Context(ContextHandler)只用来响应配匹配指定URI前缀的请求， 3](#_Toc10396)

[2.5. 2.10 嵌入 ServletContextHandler ServletContextHandler是一种特殊的ContextHandler，它可以支持标准的sessions 和Servlets。 3](#_Toc19855)

[2.6. 2.11 嵌入 web 应用(WebAppContext): 3](#_Toc12560)

[http://aiden-dong.github.io/2018/12/25/Jetty%E5%B5%8C%E5%85%A5%E5%BC%8F%E6%9C%8D%E5%8A%A1%E5%99%A8/](http://aiden-dong.github.io/2018/12/25/Jetty%E5%B5%8C%E5%85%A5%E5%BC%8F%E6%9C%8D%E5%8A%A1%E5%99%A8/" \t "https://keep.google.com/" \l "NOTE/_blank)  
  
2.2 使用处理器处理请求  
为了针对请求产生一个相应，Jetty要求用户为服务创建一个处理请求的处理器，一个处理器可以做的工作有：  
  
检测或修改一个请求  
一个完整的HTTP响应  
处理转发（详见：HandlerWrapper）  
处理转发到多个其它处理器上（详见：HandlerCollection）

## 2.2 使用处理器处理请求

为了针对请求产生一个相应，Jetty要求用户为服务创建一个处理请求的处理器，一个处理器可以做的工作有：  
  
检测或修改一个请求  
一个完整的HTTP响应  
处理转发（详见：HandlerWrapper）  
处理转发到多个其它处理器上（详见：HandlerCollection）  
  
2.4 处理器的集合以及封装 :  
复杂的请求可以由多个处理器来完成，你可以通过不同的方式把它们结合起来，Jetty有几个 HandlerContainer 接口的实现：

# 2.5 处理器的作用域 :

## 在Jetty中，很多标准的服务器会继承HandlerWrappers，用来进行链式处理，比如从请求从 ContextHandler 到 SessionHandler ，再到 SecurityHandler 最后到ServletHandler

。然而，因为servlet规范的性质，外部处理器不能在没有调用内部处理器的时候得知内部处理器的信息，例如：ContextHandler调用应用监听请求的context时，它必须已经知道ServletHandler将要转发请求到哪个servlet，以确保servletPath方法返回正确的值。

## 2.6 嵌入 ResourceHandle 来处理当前工作路径下的静态资源

你可以使用 ResourceHandler 来处理当前工作路径下的静态资源。  
  
ScopedHandler是HandlerWrapper 一个抽象实现类，用来提供链式调用时作用域的支持。例如：一个ServletHandler内嵌在一个ContextHandler中，方法嵌套调用的顺序为：  
  
Server.handle(...)  
ContextHandler.doScope(...)  
ServletHandler.doScope(...)  
ContextHandler.doHandle(...)  
ServletHandler.doHandle(...)  
SomeServlet.service(...)

## 2.8 ServletHandle

Servlets 是处理逻辑和HTTP请求的标准方式。Servlets 类似于Jetty的处理器，request对象不可变且不能被修改。在Jetty中servlet将有ServletHandler进行负责调用。它使用标准的路径匹配一个请求到servlet，设置请求的路径和请求内容，将请求传递到servlet，或者通过过滤器产生一个响应。

## 2.9 嵌入 Context(ContextHandler)只用来响应配匹配指定URI前缀的请求，

ContextHandler是一种ScopedHandler，只用来响应配匹配指定URI前缀的请求，  
一个Classloader 当在一个请求作用域里的时候处理当前线程的请求  
一个ServletContext有小属性的集合  
通过ServletContext获得的初始化参数的集合  
通过ServletContext 获得的基础资源的集合  
一个虚拟主机名称的集合

## 2.10 嵌入 ServletContextHandler ServletContextHandler是一种特殊的ContextHandler，它可以支持标准的sessions 和Servlets。

下面例子的OneServletContext 实例化了一个 DefaultServlet为/tmp/ 和DumpServlet 提供静态资源服务，DumpServlet 创建session并且应答请求信息

## 2.11 嵌入 web 应用(WebAppContext):

WebAppContext是ServletContextHandler 的扩展，使用标准的web应用组件和web.xml，通过web.xml和注解配置servlet，filter和其它特性。下面这个OneWebApp例子部署了一个简单的web应用。Web应用程序可以使用容器提供的资源，在这种情况下需要一个LoginService并配置：  
  
2.8 ServletHandle  
Servlets 是处理逻辑和HTTP请求的标准方式。Servlets 类似于Jetty的处理器，request对象不可变且不能被修改。在Jetty中servlet将有ServletHandler进行负责调用。它使用标准的路径匹配一个请求到servlet，设置请求的路径和请求内容，将请求传递到servlet，或者通过过滤器产生一个响应。  
  
imp  
  
2.8 ServletHandle  
Servlets 是处理逻辑和HTTP请求的标准方式。Servlets 类似于Jetty的处理器，request对象不可变且不能被修改。在Jetty中servlet将有ServletHandler进行负责调用。它使用标准的路径匹配一个请求到servlet，设置请求的路径和请求内容，将请求传递到servlet，或者通过过滤器产生一个响应  
  
  
1.Jetty简介  
1.1 什么是Jetty  
1.2 概述  
2. 嵌入式Jetty 使用案例:  
2.1 创建一个 server 实例  
2.2 使用处理器处理请求  
2.4 处理器的集合以及封装 :  
2.5 处理器的作用域 :  
2.6 嵌入 ResourceHandle  
2.7 多个连接的例子  
2.8 ServletHandle  
2.9 嵌入 Context(ContextHandler)  
2.10 嵌入 ServletContextHandler  
2.11 嵌入 web 应用(WebAp