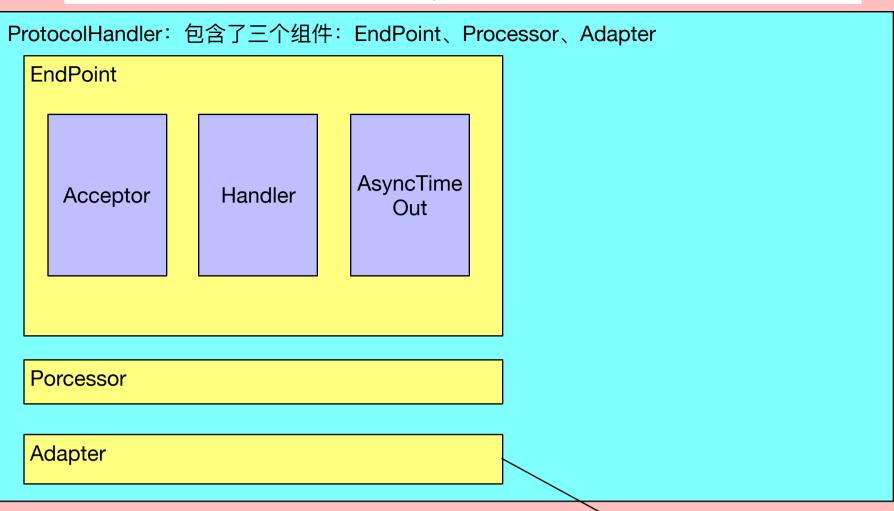
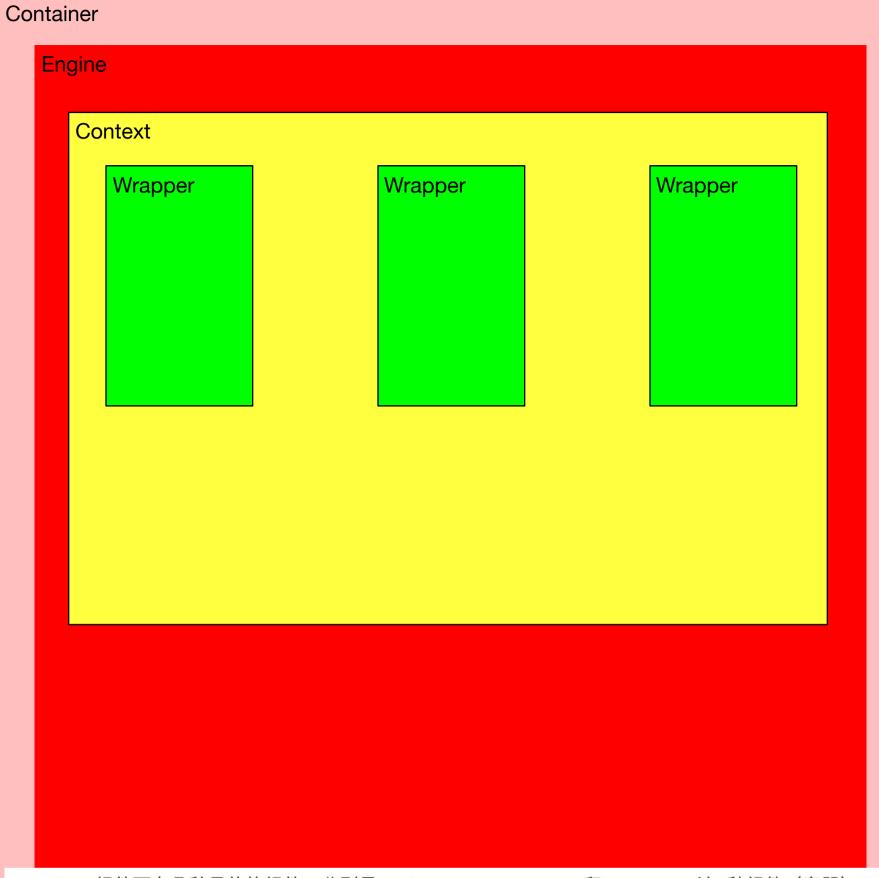
## Service: 一个server可以有多个service, 一个service只能有一个container, 但是可以有多个connector;

- 1、Connector用于处理连接相关的事情,并提供Socket与Request和Response相关的转化;
- 2、Container用于封装和管理Servlet,以及具体处理Request请求;

## Connector:就是使用ProtocolHandler来处理请求的,不同的ProtocolHandler代表不同的连接类型



组件	作用描述
EndPoint	EndPoint 是 Coyote 通信端点,即通信监听的接口,是具体Socket接收和发 送处理器,是对传输层的抽象,因此EndPoint用来实现TCP/IP协议的
Processor	Processor 是Coyote 协议处理接口,如果说EndPoint是用来实现TCP/IP协议的,那么Processor用来实现HTTP协议,Processor接收来自EndPoint的Socket,读取字节流解析成Tomcat Request和Response对象,并通过Adapter将其提交到容器处理,Processor是对应用层协议的抽象
ProtocolHandler	Coyote 协议接口, 通过Endpoint 和 Processor , 实现针对具体协议的处理能力。Tomcat 按照协议和I/O 提供了6个实现类: AjpNioProtocol , AjpAprProtocol , AjpAprProtocol , Http11NioProtocol , Http11NioProtocol ,
Adapter	由于协议不同,客户端发过来的请求信息也不尽相同,Tomcat定义了自己的Request类来封装这些请求信息。ProtocolHandler接口负责解析请求并生成Tomcat Request类。但是这个Request对象不是标准的ServletRequest,不能用Tomcat Request作为参数来调用容器。Tomcat设计者的解决方案是引入CoyoteAdapter,这是适配器模式的经典运用,连接器调用CoyoteAdapter的Sevice方法,传入的是Tomcat Request对象,CoyoteAdapter负责将Tomcat Request转成ServletRequest,再调用容器



Container组件下有几种具体的组件,分别是Engine、Host、Context和Wrapper。这4种组件(容器) 是父子关系。Tomcat通过一种分层的架构,使得Servlet容器具有很好的灵活性。

Engine

表示整个Catalina的Servlet引擎,用来管理多个虚拟站点,一个Service最多只能有一个Engine,但是一个引擎可包含多个Host

Host

代表一个虚拟主机,或者说一个站点,可以给Tomcat配置多个虚拟主机地址,而一个虚拟主机下可包含多个Context

Context

表示一个Web应用程序, 一个Web应用可包含多个Wrapper

Wrapper

表示一个Servlet, Wrapper 作为容器中的最底层,不能包含子容器