



Tomcat源码解读

T H A N K Y O U F O R W A T C H I N G

主讲老师 : King QQ: 2962938812



Tomcat8源码下载地址: <http://mirrors.shu.edu.cn/apache/tomcat/tomcat-8/v8.0.53/src/apache-tomcat-8.0.53-src.zip>

ant下载: <http://ant.apache.org/>

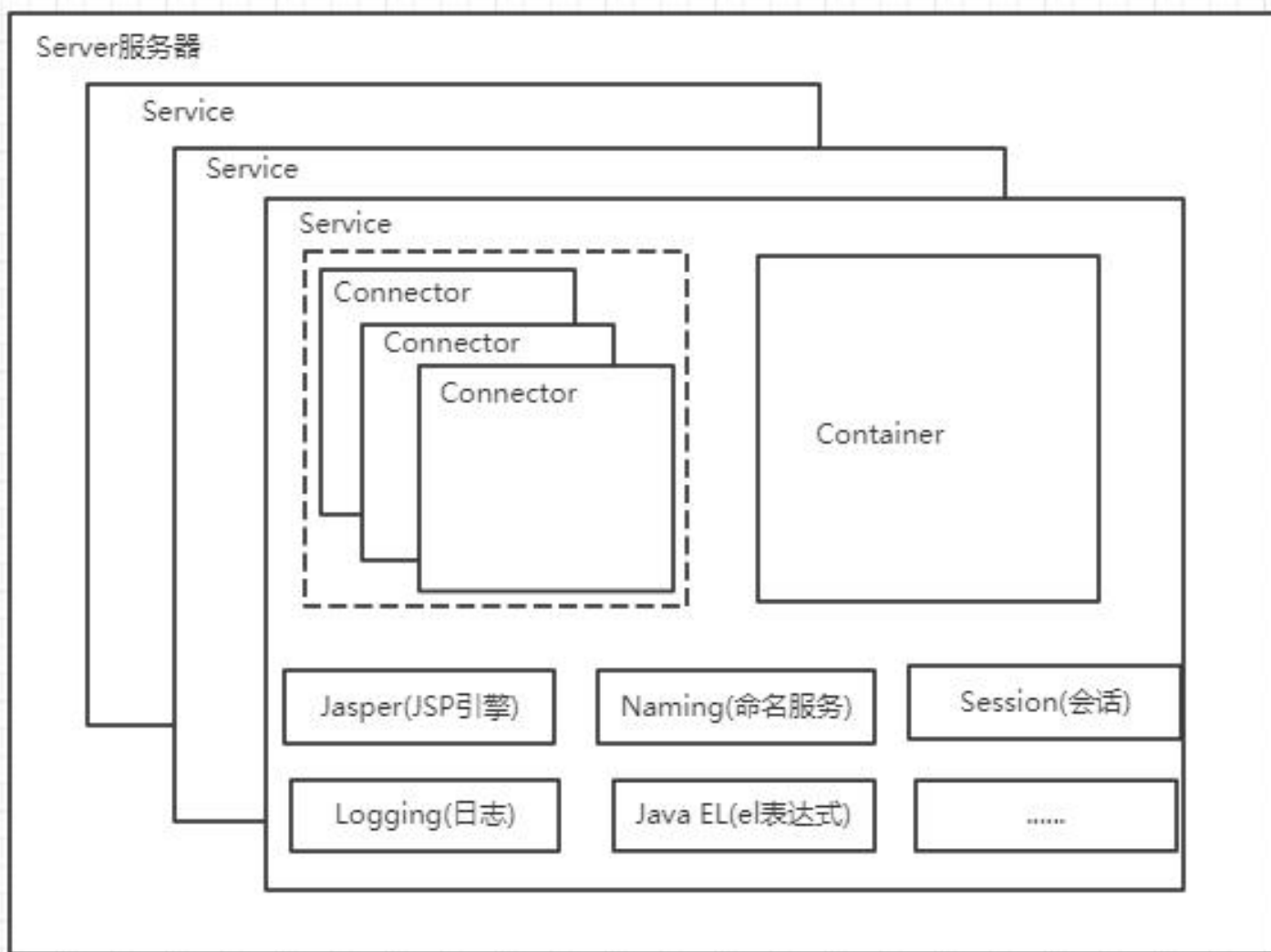
配置环境变量:

ANT_HOME	D:\ant\apache-ant-1.9.8
Path	%ANT_HOME%\bin
ClassPath	%ANT_HOME%\lib

源码目录搭建: 源码中的java包作为代码、
调试或运行时需要加VM参数 -Dcatalina.home="源码路径\output\build"

king老师的Tomcat源码链接: <https://pan.baidu.com/s/1EcB7UQLTh3zJh10Lvco8DQ> 密码: 1f08

02 Tomcat顶层结构



Server: 服务器的意思，代表整个tomcat服务器，一个tomcat只有一个Server；

Service: Server中的一个逻辑功能层，一个Server可以包含多个Service；

Connector: 称作连接器，是Service的核心组件之一，一个Service可以有多个Connector，主要是连接客户端请求；

Container: Service的另一个核心组件，按照层级有Engine, Host, Context, Wrapper四种，一个Service只有一个Engine，其主要作用是执行业务逻辑；

Jasper: JSP引擎；

Session: 会话管理；

...



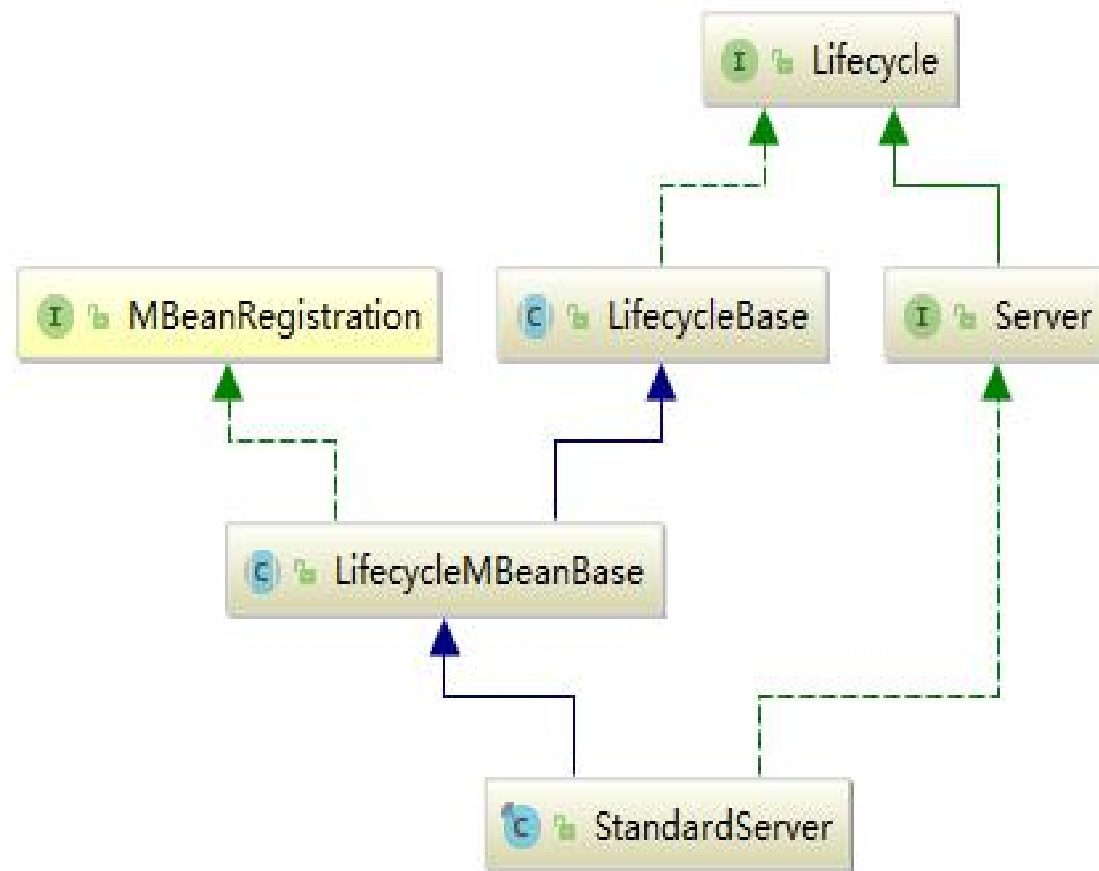
Server是Tomcat最顶层的容器，代表着整个服务器

Tomcat中其标准实现：

org.apache.catalina.core.**StandardServer**类

StandardServer继承结构类图如右图：
除了实现**Server**接口还需要继承**Lifecycle**

好处：生命周期统一接口Lifecycle把所有的启动、停止、关闭等都放在一起统一管理





Service : 一个Tomcat包含多个Service

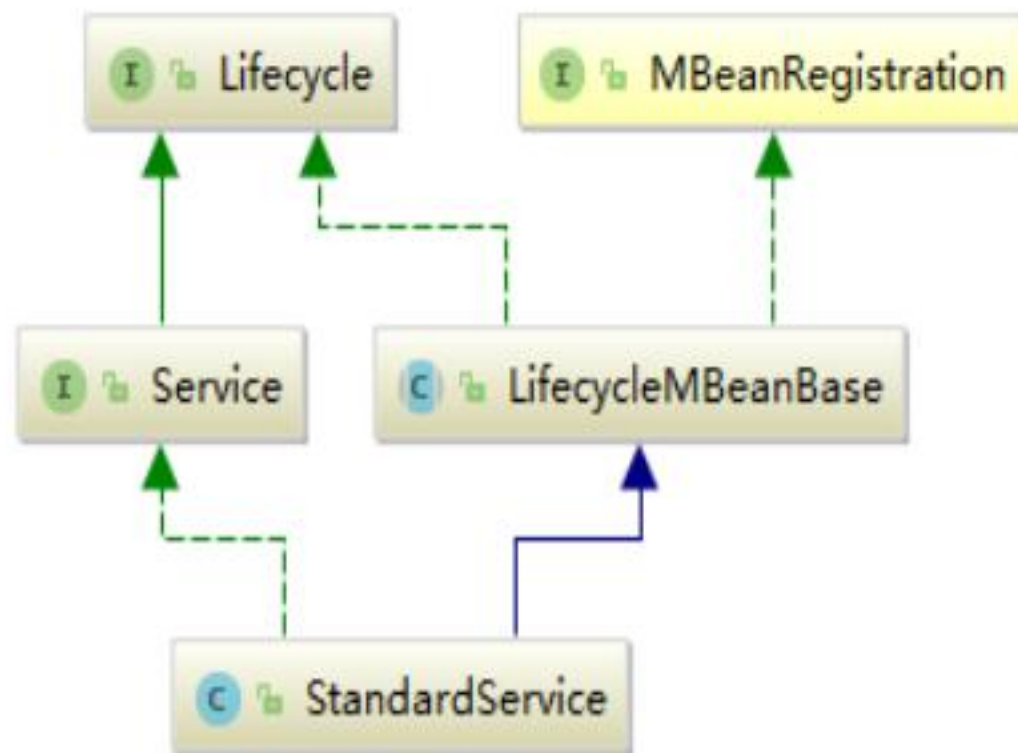
Tomcat中其标准实现:

org.apache.catalina.core.**StandardService**类

StandardService继承结构类图如右图:

除了实现**Service**接口还需要继承**Lifecycle**

好处: 生命周期统一接口Lifecycle把所有的启动、停止、关闭等都放在一起统一管理



05 Service拓展出Tomcat原型



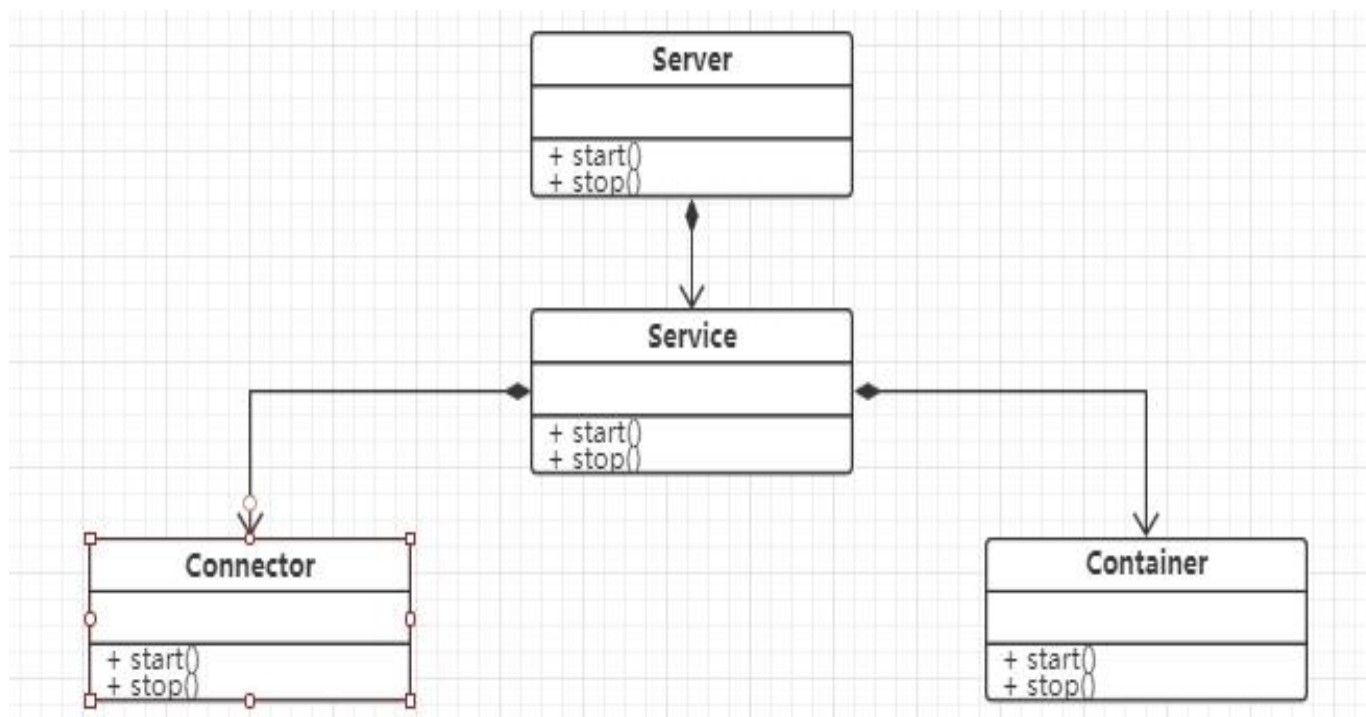
Service中请求监听和请求处理分开为两个模块：

Connector负责处理请求监听；

Container负责处理请求处理；

一个Service可以有多个Connector，但只能有一个Container

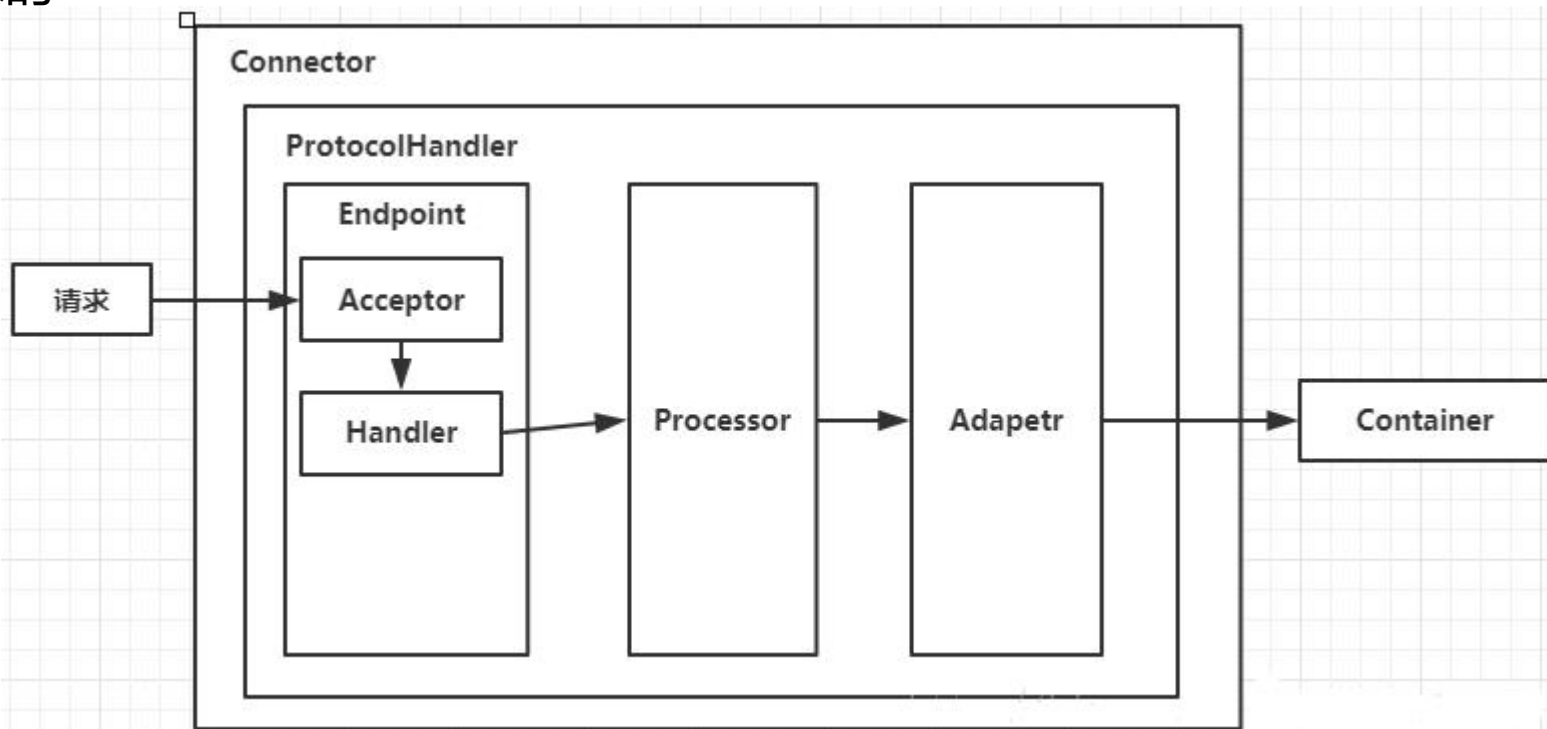
任何容器都有启动start()和关闭stop()方法



Connector使用ProtocolHandler来处理请求的

ProtocolHandler由包含了三个部件：
Endpoint、Processor、Adapter

Endpoint用来处理底层Socket的网络连接。
Processor用于将Endpoint接收到的Socket封装成Request。
Adapter充当适配器，用于将Request转换为ServletRequest交给Container进行具体的处理

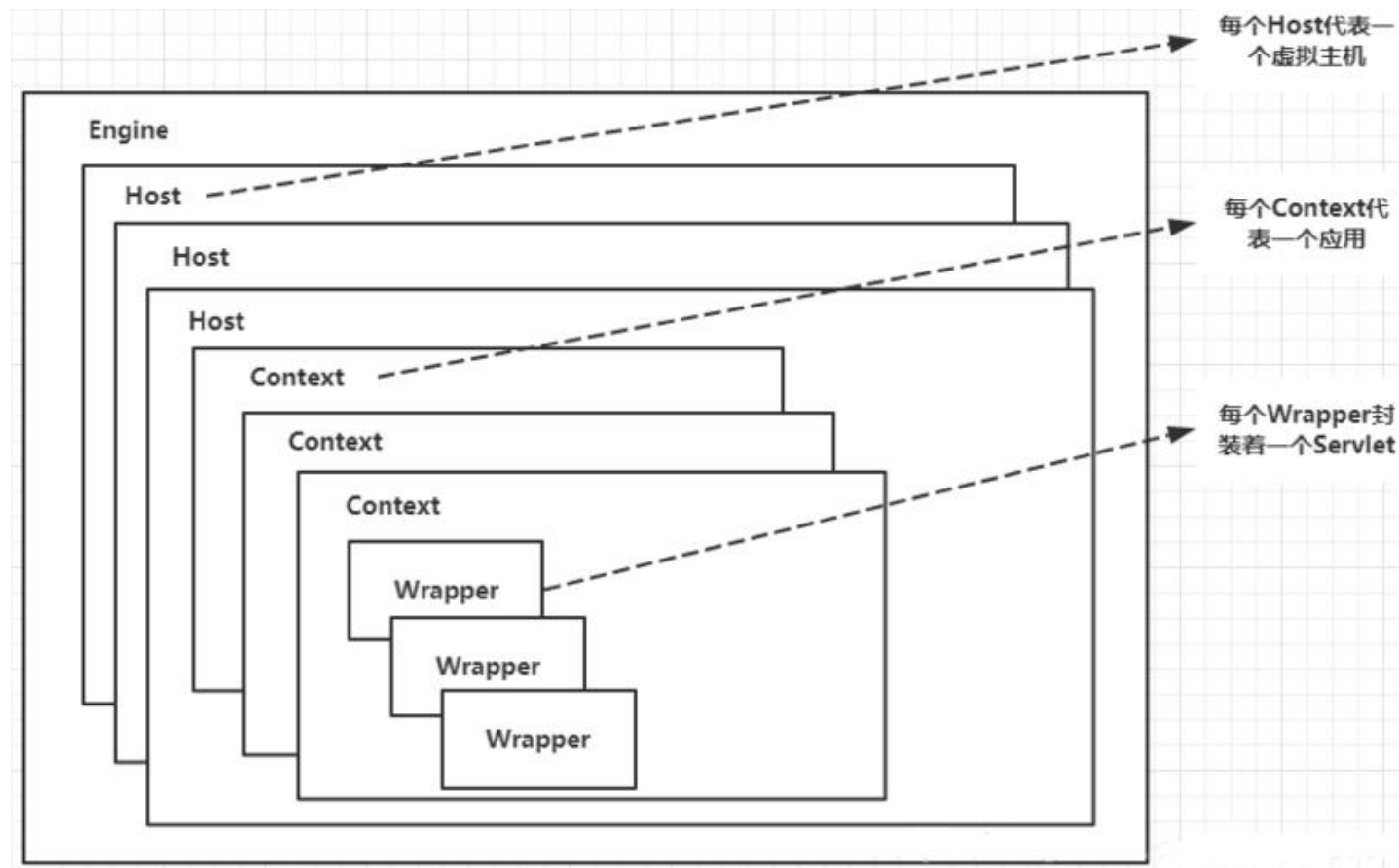


Engine:引擎、只有一个
定义了一个名为Catalina的Engine

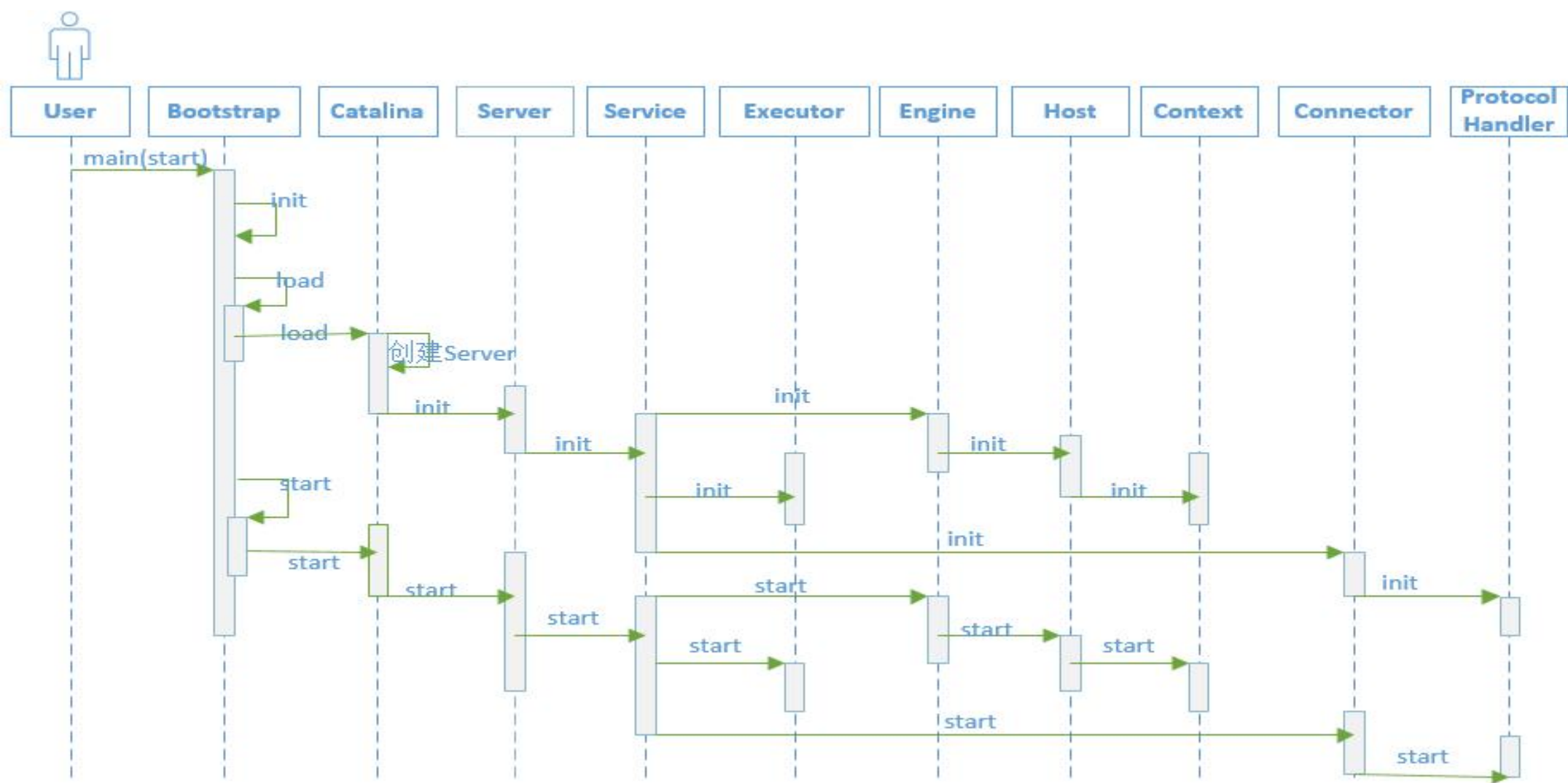
Host:站点、虚拟主机
一个Engine包含多个Host的设计，使得一个服务器实例可以承担多个域名的服务，是很灵活的设计

Context:一个应用
默认配置下webapps下的每个目录都是一个应用

Wrapper:一个Servlet



08 Tomcat启动流程





Tomcat启动两大流程:`init`流程和`start`流程

`init`流程:分别在Bootstrap、Catalina、StandardServer、StandardService 的`init`方法加入断点和输出日志

`start`流程:分别在Bootstrap、Catalina、StandardServer、StandardService 的`start`方法加入断点和输出日志



模板方法就是为多种类似业务提供一个算法执行的统一框架，把这些业务中共同的部分**抽取出来进行具体实现**，而某些业务中特定的需求**推迟到子类中进行重写实现**。

案例：

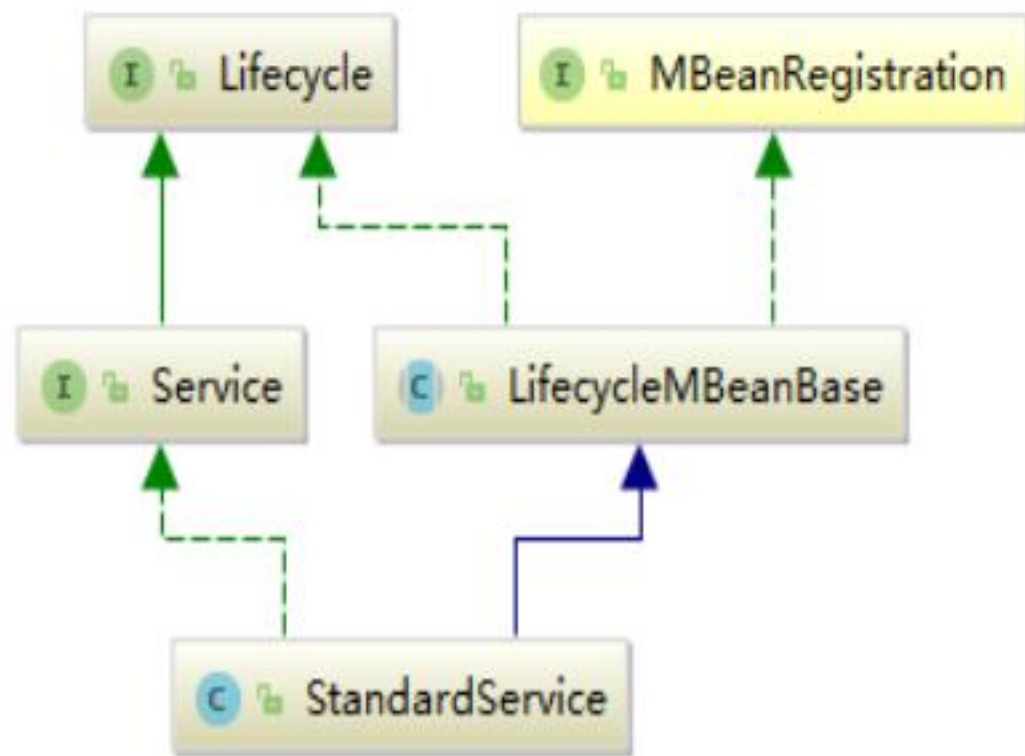
Tomcat的启动过程中Catalina调用StandardService中的start()方法，**但是StandardService自身没有start()方法！**

分析：

原来StandardService继承了抽象类LifecycleBase，它有start()并且它在此方法中调用了一个未实现的抽象方法startInternal()，Catalina调用StandardService中的start()最后会调用至startInternal()

优点：

这种模式使得StandardService等这些类抽出一个共同的start()在LifecycleBase中进行实现（方便统一生命周期管理）。如果它需进行特殊的业务处理的话可以在startInternal()中处理





完 毕！

享学课堂：King老师