## 讲课内容

#### Tomcat下载和安装

#### Tomcat的顶层架构

Server和Service是Tomcat核心组件中最外层的两个组件.

我们看源码要能够把快速入手时就必须做到抓主放次，我们要抓主主要的、核心的内容，把次要的东西先放一边。

#### Server

这里要强调一点，整个Tomcat中的设计方式，我们讲的都是一个抽象，抽象在源码中都是接口，具体的实现一般都是StandardXXXX之类的

Lifecycle这个概念重点讲下：像Tomcat这么大的系统，必要要对生命周期进行统一的管理，所以基本上大部分的组件都会去继承这个接口，Lifecycle把所有的启动、停止、关闭等都组织在了一起。

**补充点：**MBeanRegistration这个类是完成JMX的功能，就是我们俗称的监控管理功能，之前我们讲的使用jconsole查看Tomcat也就是通过JMX玩的。

#### Serivce

跟Server差不多，不过老师要强调Service是我们看源码的黄金分割线，看他之前的比较简单，看它后面的处理就比较深入了（它启动的分支也比较多）。

#### Serivce拓展出Tomcat原型

Tomcat的包虽然不大，但是他的非常的模块化，模块非常多，所以我们在解读的时候要注意只需要抓住主要的部分，把那些非主线的模块放弃掉。

#### Connector解析

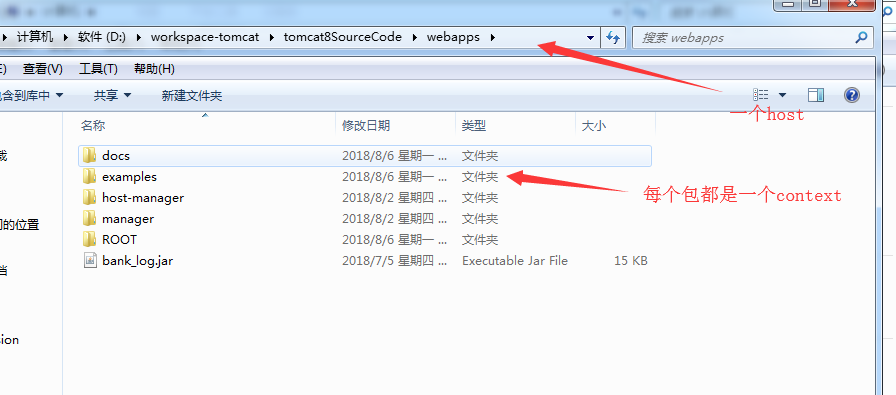
这个部分可以结合server.xml一起来看，会理解得比较透彻

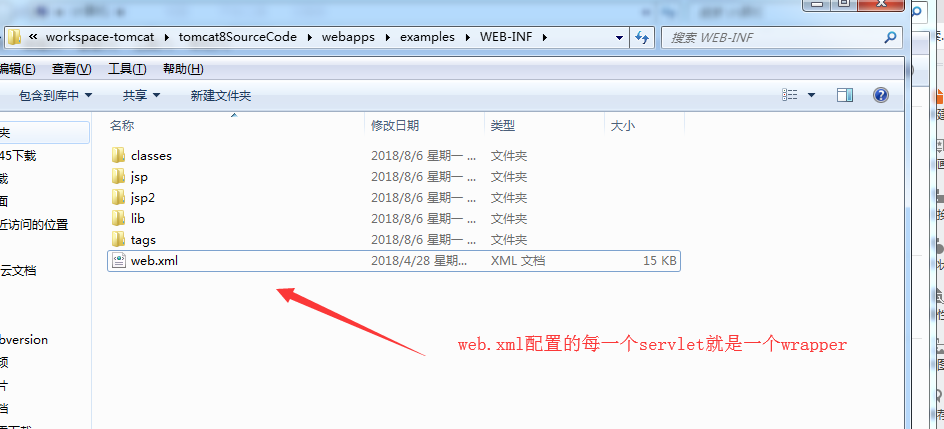
#### Container解析

service中的name是catalina,engine中的name也是catalina，这个就证明一个service就只有一个engine, 一个engine可以有多个host主机

host，站点，虚拟主机

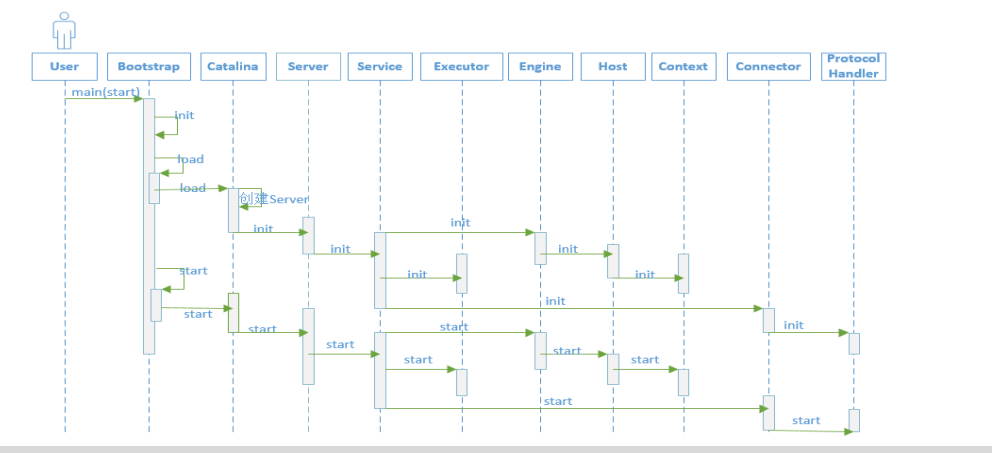
在理解下面的4个容器之间的关系





#### Tomcat的启动流程

Tomcat的启动大家一看好像比我们之前多了两个东西出来了，一个是bootstrap，一个是catalina。首先只有bootstrap中有main方法。我整体去看启动过程可以分为两个阶段，这样看的话就比较容易了。我们使用切割话，找到黄金切割线，这个service这里就是一个黄金切割线。左边的，就包含前面讲的server和service，后面的就是具体的很细的组件了。都可以看出，左边都是为了加载service,右边的serivce会引发三个动作，一个是启动engine，一个是启动connector，最后一个是启动executor也就是执行现场。这样看的话，整体的启动流程就非常清爽了。



#### Tomcat源码预览1——启动流程

找Bootstrap启动类 -> catalina -> StandardServer -> StandardService

#### 调试Tomcat源码2——Lifecycle与模板方法模式

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,所有的用到的方法前加入断点，加入日志输出（要注意由于LifecycleBase中很多东西都继承了它，包括容器等，所以这里不建议加断点）