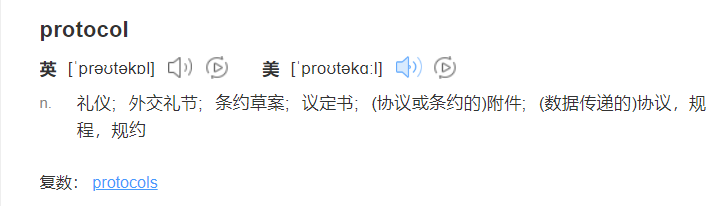
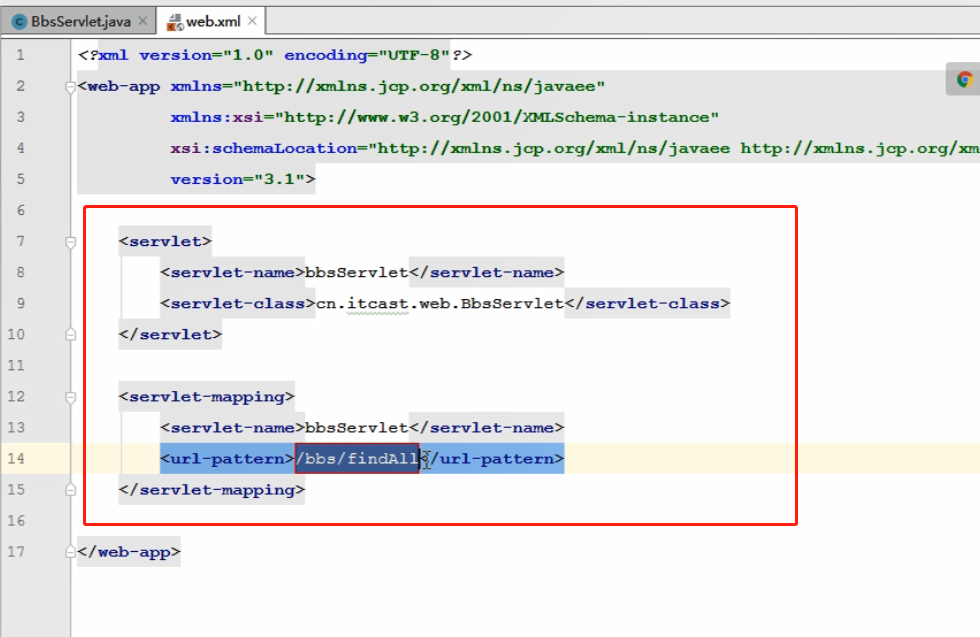
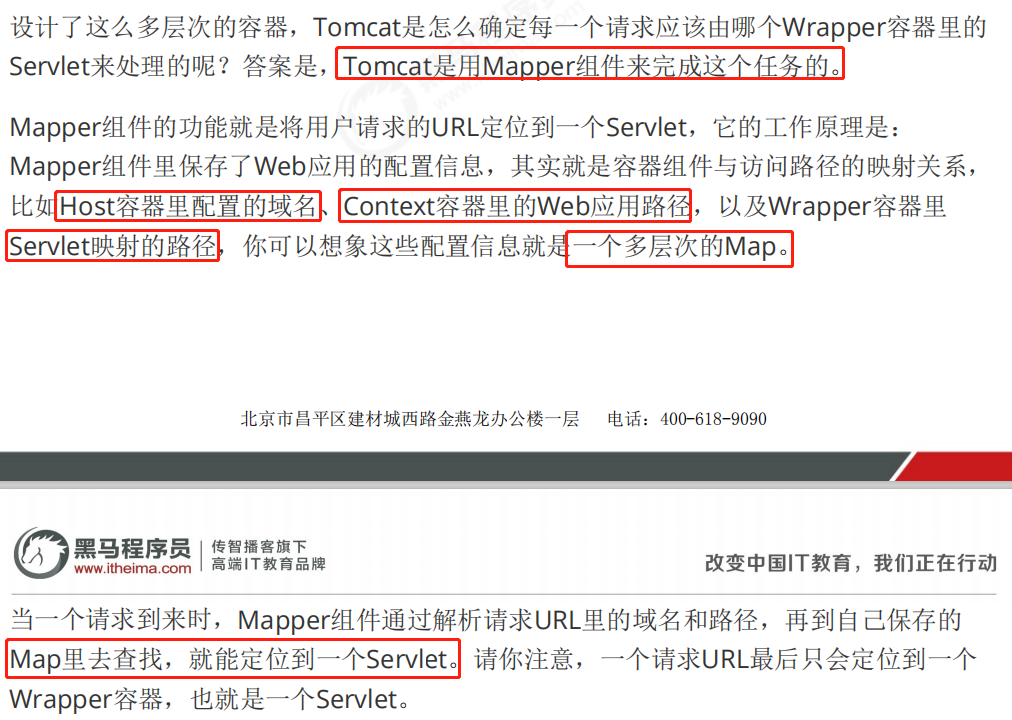
模板设计模式，把公有的方法抽取到父类中，子类去实现自己的逻辑（lifecycle接口，lifecycleBase抽象类）

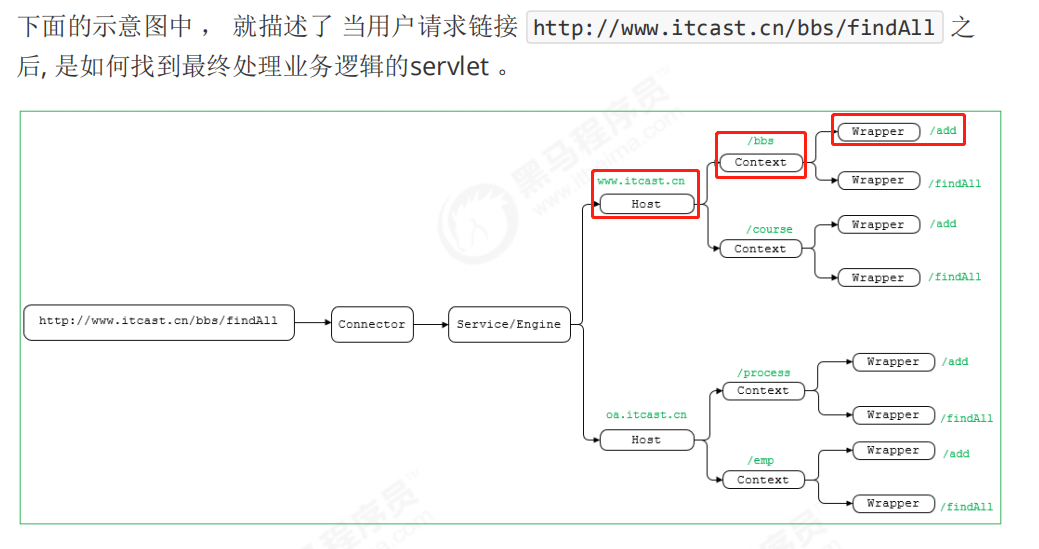


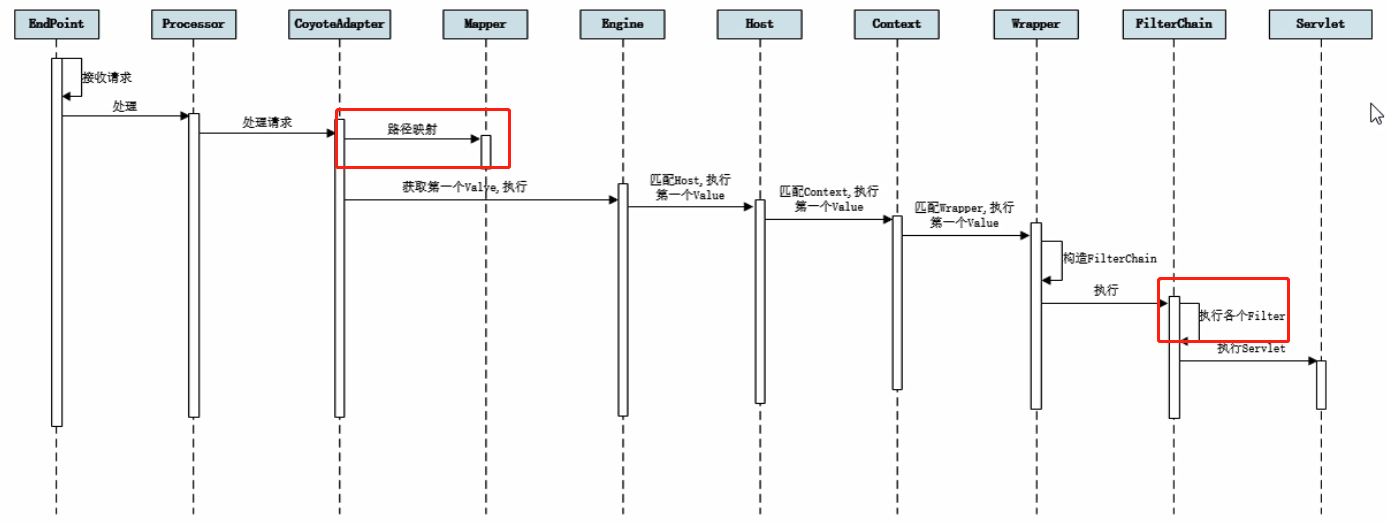
ProtocolHandle



传统的servlet

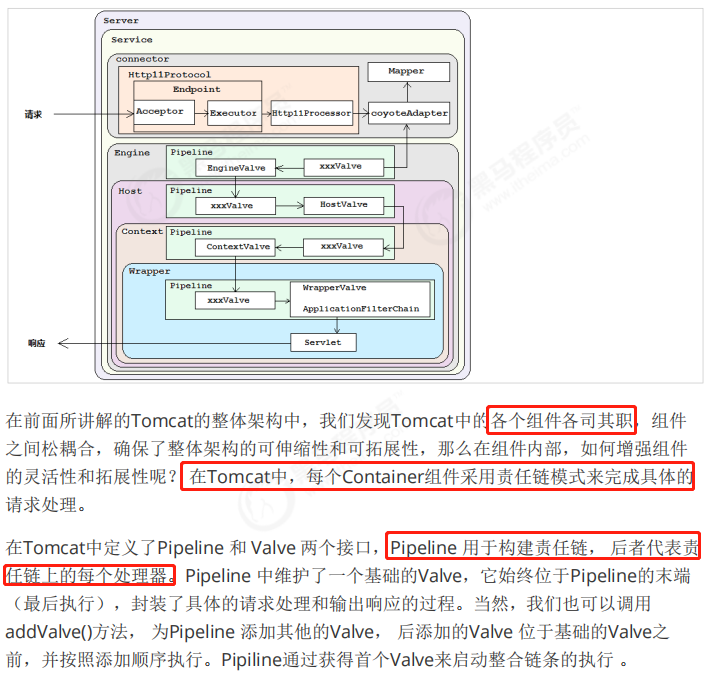




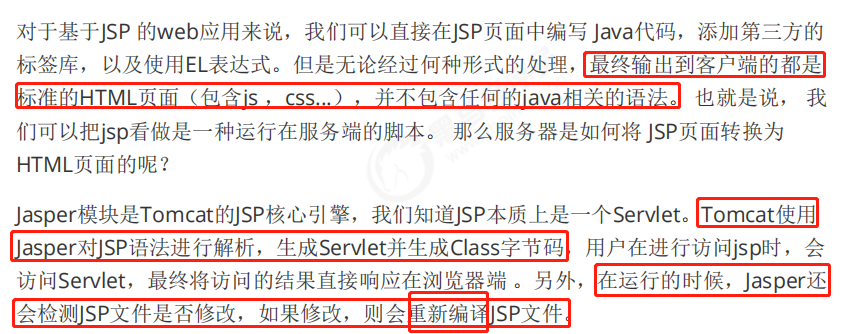


适配器到Mapper解析多层级路径然后解析到wrapper(servlet的包装对象)执行filter再执行对应的servlet

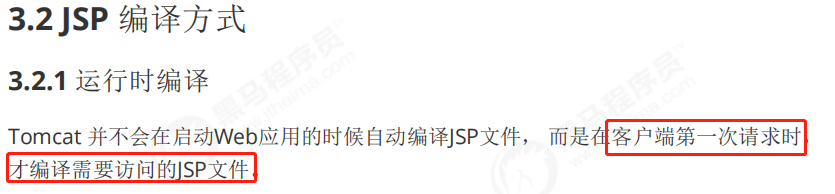
Mapper组件根据请求行的URL值和请求头的Host值匹配由哪个Host容器、Context容器、Wrapper容器处理请求。



责任链，调用第一个处理完会自动去处理链条的下一个对象

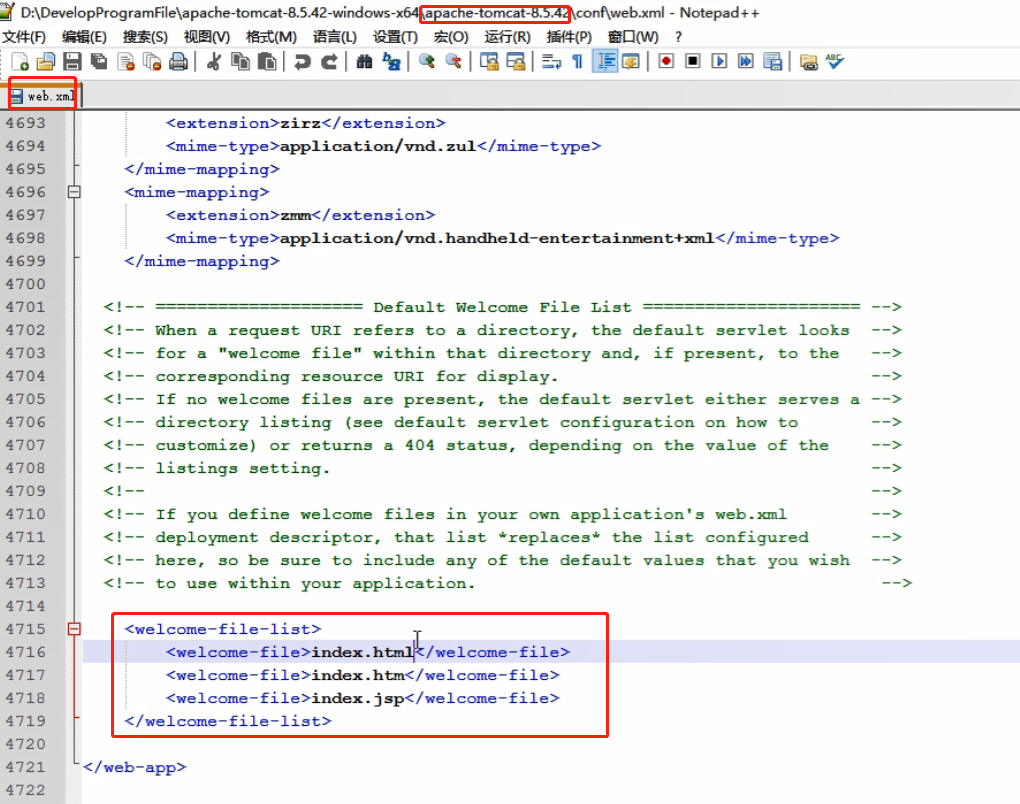


Jsp修改会重新编译（jsp文件都会生成一个java文件和一个class文件，根据class文件执行servlet返回response结果html页面，tomcat的jsp引擎的作用）





编译完放在work目录



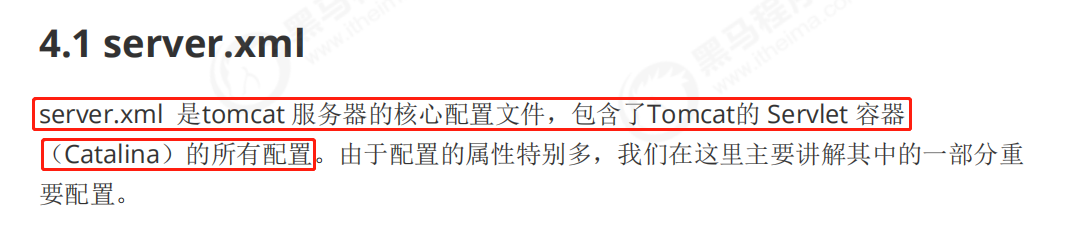


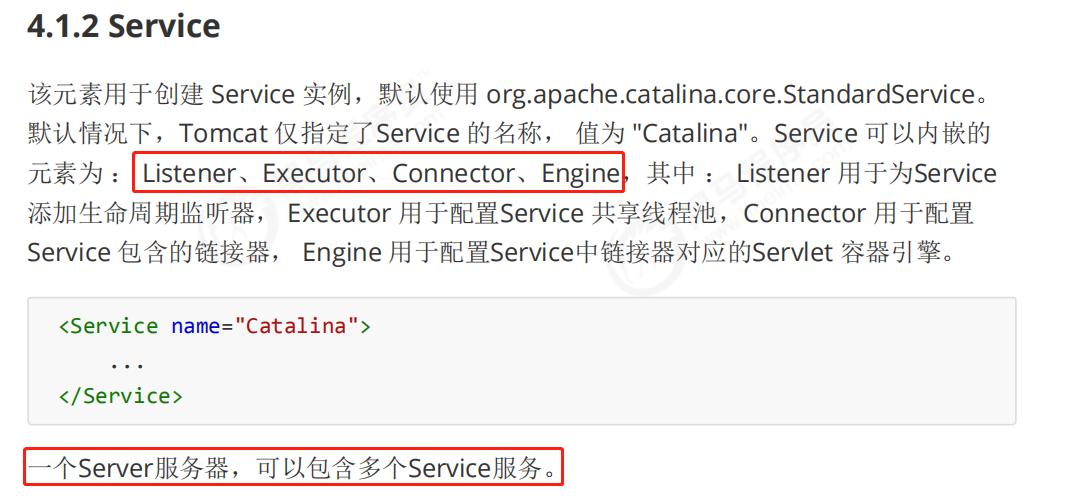
所以无需输入index.jsp自动会去jsp页面





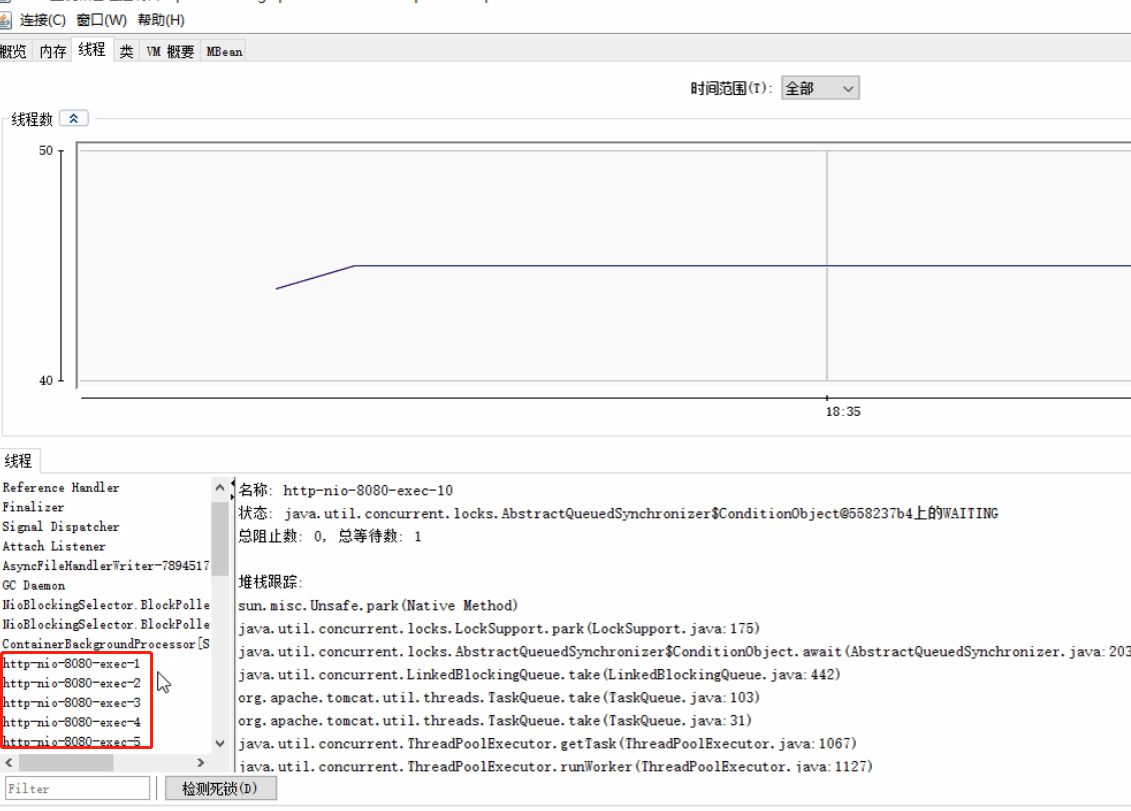
Jsp默认运行时编译所以修改不需要重启服务器





如果不配置共享线程池，那么Catalina 各组件在用到线程池时会独立创建。

Service里面可以配置共享线程池，不然里面配置的connector都会自己创建线程池（默认是自己创建的线程池10个活跃线程）



8080http协议的线程池

核心线程是不会销毁的，会一直存在

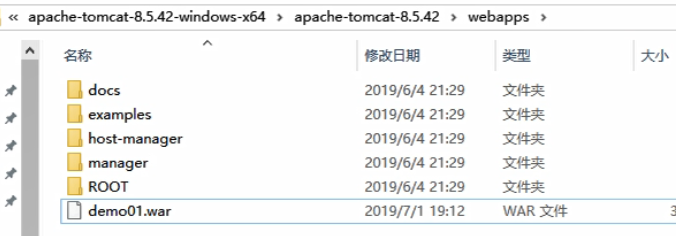








unpackWARs会解压war包,所以把war包丢到webapps就行了，不需要特定解压完再丢进去



Host可以通过域名和主机来访问不同的配置

name配置域名,appBase配置文件位置，就可以实现不同域名访问不同文件目录



