快手内部文档请勿外传

Tomcat是一个Web应用服务器，可以作为Servlet容器。它的作用是，解析客户端client发起的request，并组装出HttpRequest、创建HttpResponse，将二者交于内部的HttpServlet处理和填充，如图所示...

SpringMVC在tomcat中执行

WEB容器在启动时，它会为每个WEB应用程序都创建一个对应的ServletContext对象（每个web应用程序唯一），它代表当前web应用web容器提供其一个全局的上下文环境，其为后面的spring IoC容器提供宿主环境

Tomcat映射处理请求的Servlet是通过web.xml做的。

SpringMVC使用一个DispatcherServlet来接收所有的请求，并把它们分发到不同的controller中来做进一步处理。

SpringMVC = Spring + Web框架，Spring这部分主要是AOP/IOC容器。

SpringBoot是Spring的扩展，简化了Spring的配置，通过starter的方式简化了常用组件依赖的引入，使其更加易用。

SpringBoot内置了tomcat。

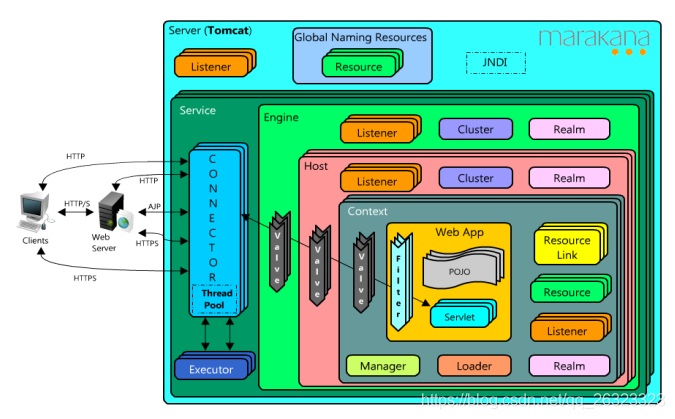
Tomcat是一个Web应用服务器，可以作为Servlet容器。它的作用是，解析客户端client发起的request，并组装出HttpRequest、创建HttpResponse，将二者交于内部的HttpServlet处理和填充，如图所示

Tomcat 是Web应用服务器,是一个Servlet/JSP容器. Tomcat 作为Servlet容器,负责处理客户请求,把请求传送给Servlet,并将Servlet的响应传送回给客户.而Servlet是一种运行在支持Java语言的服务器上的组件. Servlet最常见的用途是扩展Java Web服务器功能,提供非常安全的,可移植的,易于使用的CGI替代品.

# Tomcat和Servlet

Tomcat是一个Servlet容器，它将前后端交互过程中不变的东西（网络通信、协议解析等）封装了起来。而Servlet是一个逻辑处理器，它可以被Tomcat创建、调用和销毁。所以我们的web程序是基于Servlet的，一个Web项目中有一个或者多个Servlet，Web程序依靠Tomcat启动。另外，Tomcat是多线程的，每一个请求都是一个线程，而Servlet是单个的实例，所以，【敲黑板】我们必须要注意在自己的Servlet的中，尽量不要用全局变量，如果要用全局变量，一定要有同步操作。当然，如果全局变量是只读的，那么没有关系。

实际上，**Tomcat通过web.xml文件获取所有Servlet类的路径**，然后通过反射的方法实例化Servlet、调用Servlet的service方法。然后，service方法中会对HTTP的请求类型进行分类，调用doPost、doGet、doDelete等方法。这些方法都是抽象的。我们开发web程序的程序员通过继承HttpServlet可以覆写这些方法。另外，Tomcat会传过来两个参数：（HttpServletRequest request, HttpServletResponse response）。我们通过request对象获取请求信息，通过response对象向Tomcat返回数据。



# Spring是一种利用了注解、反射和模板等技术的Servlet

spring是基于Servlet的，它的核心类是DispatchServlet（分发Servlet）。它继承了HttpServlet，所以它还是一个Servlet。只不过它覆写了HttpServlet的init、doPost、doGet等方法。它在init中通过类加载器检索所有的@Controller、@Service等类，然后通过反射实例化类，通过反射向类中注入实例依赖。著名顶顶的IOC是什么？其实，它就是一个HashMap！！！key=类长名，value=实例对象。在init函数中，它还会通过反射获取所有带有@RequestMapping的类和方法。然后维护一个handlerMapping的对象，key=URL，value=method。

现在假设有一个http请求来了，Tomcat首先收到了请求，并将请求的信息封装成HttpServletRequest 传给了 DispatchServlet，DispatchServlet通过URL是不是可以获取到对应的方法。然后，DispatchServlet中的doPost和doGet方法只需要对HttpServletRequest中的参数进行解析，创建一个参数列表。然后通过"method.invoke(方法所在的实例对象，通过IOC容器获取, 参数列表);"即可触发我们自己写的方法。

# JSP也是一个Servlet

我们浏览器输入URL，Tomcat收到URL，找到相应的文件。如果是.jsp文件的话，先将.jsp解析翻译成.java文件，且这个.java文件是一个Servlet类。然后调用javac将.java编译成.class文件。最后，通过JVM加载执行。所以，JSP是Servlet没毛病吧？

可以看出，JSP在第一次需要解析、编译、加载、运行。而且JSP生成的Servlet是通过后台服务器运行的。这给服务器造成了巨大的压力，因为，每一个页面都需要一个Servlet。

最重要的是，每次请求，后台都会返回一整个页面给浏览器（页面是Servlet动态生成的）。就算只是变了一个字符，都TMD的全发一遍，多么的不合理啊！！！

Ajax本质上是一个浏览器端的技术.

其实质是通过js内置的ajax引擎对象(XMLHttpRequest)去后台获取数据,后台servlet进行处理,这样从后台获取来的数据就只是传递给了js中的方法,并不是整个界面;

快手内部文档请勿外传