**servlet的一些总结**

1.SUN公司定义了两个servlet接口的默认实现类：GenericServlet和HttpServlet.

HttpServlet指能够处理HTTP请求的servlet，它在原有的Servlet接口上添加了一些与HTTP协议相关的处理方法，它比Servlet接口的功能更强大。

GenericServlet业务方法为service。不仅仅可以处理HTTP请求，还可以处理其他请求，但javaweb是浏览器开发，使用的基本上是HTTP请求，所以绝大多数情况下应该继承HttpServlet。

2.servlet是一个供其他Java程序（Servlet引擎）调用的Java类。它不能独立运行，完全由Servlet引擎（容器）控制和调度。

针对客户端的多次Servlet请求，通常服务器只会创建一个Servlet实例对象，即Servlet实例对象一旦创建，它就会驻留在内存中，为后续的其他请求服务，直至web容器退出，Servlet实例对象才会销毁。

3.应用服务器启动后，Servlet不会马上被创建（优化原则），当第一次Http请求时，web服务器收到客户端的Servlet访问请求后：

①web服务器首先检查是否已经装载并创建了该Servlet的实例对象。如果是，则直接执行第④步，否则，执行第②步；

②装载并创建该Servlet的一个实例对象；

③调用Servlet实例对象的init方法；

④创建一个用于封装HTTP请求消息的HttpServletRequest对象和一个代表HTTP相应消息的HttpServletResponse对象，然后调用Servlet的service方法将请求和响应对象作为参数传递过去；

⑤web应用程序被停止或重新启动之前，Servlet引擎将卸载servlet，并在卸载之前调用destroy方法。

4.servlet和JSP的区别

①JSP经编译就变成了Servlet。JSP的本质为Servlet，JVM只能识别Java的类，不能识别JSP的代码，web容器将JSP的代码编译成JVM能够识别的Java类。

②Servlet中的内置对象都必须通过HttpServletRequest对象、HttpServletResponse对象以及HttpServlet对象得到。Jsp是Servlet的一种简化，使用JSP只需要完成程序员需要输出到客户端的内容。

③JSP中的Java脚本如何镶嵌到一个类中，由JSP容器完成，而Servlet则是一个完整的Java类，这个类的service方法用于生成对客户端的相应。

5.不同的用户同时对同一个业务（如注册）发出请求，那这个时候容器里产生的有几个Servlet实例呢？

答：只有一个Servlet实例。一个Servlet是在第一次被访问时加载到内存并实例化的。同样的业务请求共享一个Servlet实例。不同的业务请求一般对应不同的Servlet。

JSP中存在的多线程问题：当客户端第一次请求某一个JSP文件时，服务端把该JSP编译成一个CLASS文件，并创建一个该类的实例，然后创建一个线程处理客户端的请求。如果有多个客户端同时请求该JSP文件，则服务端会创建多个线程。每个客户端请求对应一个线程。以多线程方式执行可大大降低对系统的资源请求，提高系统的并发量和响应时间。

6.Servlet的线程安全问题

让Servlet去实现一个SingleThreadModel接口，如果某个Servlet实现了SingleThreadModel接口，那么Servlet引擎将以单线程模式来调用其service方法。

查看Servlet的API可以看到，SingleThreadModel接口中没有定义任何方法和变量，在JAVA中，把没有定义任何方法和常量的接口称之为标记接口，经常看到的一个最典型的标机接口就是“Serializable”，这个接口也没有定义任何方法和变量，标机接口在JAVA里的作用：给某个对象打上一个标志，告诉JVM，这个对象可以做什么，比如实现了“Serializable”接口的类的对象就可以被序列化。

对于实现了SingleThreadModel接口的Servlet，Servlet引擎仍可支持对该Servlet的多线程并发访问，其采用的方式是产生多个Servlet实例对象，并发的每个线程分别调用一个独立的Servlet实例对象。

实现SingleThreadModel接口并不能真正解决Servlet的线程安全问题，因为Servlet引擎会创建多个Servlet实例对象，而真正意义上解决多线程安全问题是指一个Servlet实例对象被多个线程同时调用的问题。事实上，在Servlet API 2.4中，已将SingleThreadModel标记为过时的。