什么是servlet?

含义：本质就是一个java类，继承HttpServlet，只需要覆盖doGet和doPost方法

作用：接收页面的请求数据，以及负责响应页面给客户端。

Servlet 的生命周期，并说出 Servlet 和 CGI 的区别？

1.Servlet 生命周期1、加载 2、实例化 3、初始化 4、处理请求 5、销毁

2.Servlet与cgi的区别：

Servlet处于服务器进程中，它通过多线程方式运行其service方法，一个实例可以服务于多个请求，并且其实例一般不会销毁，

而CGI对每个请求都产生新的进程，服务完成后就销毁，所以效率上低于servlet。

3.Servlet与JSP的比较：

有许多相似之处，都可以生成动态网页。

JSP的优点是擅长于网页制作，生成动态页面比较直观，缺点是不容易跟踪与排错。

Servlet是纯Java语言，擅长于处理流程和业务逻辑，缺点是生成动态网页不直观。

Servlet 接口中有哪些方法？

Servlet接口定义了5种方法：

•init() 初始化 一次

•service() 根据请求方式决定调用doGet还是doPost 根据请求有多少次执行多少次

•destroy() 服务器关闭 销毁

•getServletConfig() 该方法返回容器调用init()方法时传递给servlet对象的servletConfig对象，servletConfig对象包含了Servlet的初始化参数。

•getServletInfo() 返回一个String类型的字符串，其中包括了关于Servlet的信息。该方法返回的应该是纯文本字符串，而不是任何类型的标记。

get 和 post 请求的区别？

get提交方法 明文提交 不安全 数据量较小

1. a链接
2. 表单method不指定 默认get提交

Post提交方式 密文提交 安全 数据量大

1. form表单 method指定post

转发（Forward）和重定向（Redirect）的区别？

响应重定向 请求两次 url地址会发生改变 不会携带请求参数到下一个组件

请求转发 请求一次 url地址栏不会发生改变 会携带请求参数到下一个组件

Servlet 与线程安全

Servlet不是线程安全的。

当Tomcat接收到Client的HTTP请求时，Tomcat从线程池中取出一个线程，之后找到该请求对应的Servlet对象并进行初始化，之后调用service()方法。要注意的是每一个Servlet对象再Tomcat容器中只有一个实例对象，即是单例模式。如果多个HTTP请求请求的是同一个Servlet，那么着两个HTTP请求对应的线程将并发调用Servlet的service()方法。

JSP 和 Servlet 有哪些相同点和不同点，他们之间的联系是什么？

1. jsp经编译后就变成了Servlet.(JSP的本质就是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)  
   2.jsp更擅长表现于页面显示,servlet更擅长于逻辑控制.  
   3.Servlet中没有内置对象，Jsp中的内置对象都是必须通过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，Jsp中的Java脚本如何镶嵌到一个类中，由Jsp容器完成。而Servlet则是个完整的Java类，这个类的Service方法用于生成对客户端的响应。  
   联系： JSP是Servlet技术的扩展，本质上就是Servlet的简易方式。JSP编译后是“类servlet”。Servlet和JSP最主要的不同点在于，Servlet的应用逻辑是在Java文件中，并且完全从表示层中的HTML里分离开来。而JSP的情况是Java和HTML可以组合成一个扩展名为.jsp的文件。JSP侧重于视图，Servlet主要用于控制逻辑

JSP 工作原理：

JSP处理

以下步骤表明了Web服务器是如何使用JSP来创建网页的：

1.就像其他普通的网页一样，您的浏览器发送一个HTTP请求给服务器。

2.Web服务器识别出这是一个对JSP网页的请求，并且将该请求传递给JSP引擎。通过使用URL或者.jsp文件来完成。

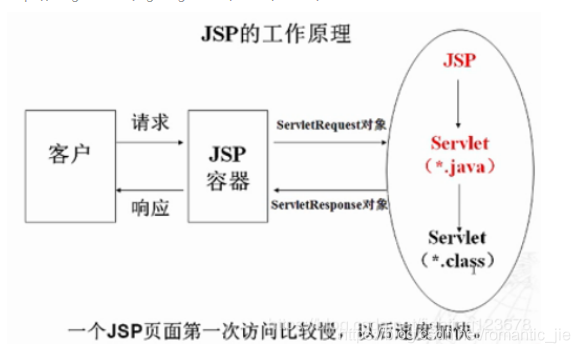
3.JSP引擎从磁盘中载入JSP文件，然后将它们转化为servlet。这种转化只是简单地将所有模板文本改用println()语句，并且将所有的JSP元素转化成Java代码。

4.JSP引擎将servlet编译成可执行类，并且将原始请求传递给servlet引擎。

5.Web服务器的某组件将会调用servlet引擎，然后载入并执行servlet类。在执行过程中，servlet产生HTML格式的输出并将其内嵌于HTTP response中上交给Web服务器。

6.Web服务器以静态HTML网页的形式将HTTP response返回到您的浏览器中。

7.最终，Web浏览器处理HTTP response中动态产生的HTML网页，就好像在处理静态网页一样。



JSP 中动态 include 和静态 include 的区别？

1. 动态INCLUDE用jsp:include动作实现<jsp:include page = included.jsp flush = "true">它总会检查所含文件中的变化，适用于动态页面并且可以带参数。
2. 静态INCLUDE用include伪码实现，定不会检查所含文件的变化，适用于包含静态页面<%@include file=included.html%>

JSP 有哪些内置对象？作用分别是什么？

1.HttpServletRequet类的Request对象：代表请求对象，主要用于接受客户端通过HTTP协议连接传输服务器端的数据。

2.HttpSevletResponse类的Response对象：代表响应对象，主要用于向客户端发送数据。

3.JspWriter类的out对象：主要用于向客户端输出数据，out的基类是jspWriter

4.HttpSession类的session对象：主要用来分别保存每个月的信息与请求关联的会话；会话状态的维持是web应用开发者必须面对的问题。

5.ServletContext类的application对象：主要用于保存用户信息，代码片段的运行环境；它是一个共享的内置对象，即一个容器中的多个用户共享一个application，故其保存的信息被所有用户所共享。

6.PageContext类的PageContext对象：管理网页属性，为jsp页面包装页面的上下文，管理对属于jsp的特殊可见部分中已经命名对象的访问，它的创建和初始化都是由容器来完成的。

7.ServletConfig类的Config对象：代码片段配置对象，标识Servlet的配置。

8.Object类的Page对象，处理jsp页面，是object类的一个实例，指的是jsp实现类的实例

9.Exception对象：处理jsp文件执行时发生的错误和异常，只有在错误页面里才使用，前提是在页面指令里要有isErrorPage=true。

JSP 常用指令有哪些？

Jsp一共包含三个编译指令和七个动作指令。

三个编译指令为：page、include、taglib。

七个动作指令为：jsp:forward、jsp:param、jsp:include、jsp:plugin、jsp:useBean、jsp:setProperty、jsp:getProperty。

讲解 JSP 中的四种作用域

JSP中的四种作用域包括page、request、session和application，具体来说：

page代表与一个页面相关的对象和属性。

request代表与Web客户机发出的一个请求相关的对象和属性。一个请求可能跨越多个页面，涉及多个Web组件；需要在页面显示的临时数据可以置于此作用域。

session代表与某个用户与服务器建立的一次会话相关的对象和属性。跟某个用户相关的数据应该放在用户自己的session中。

application代表与整个Web应用程序相关的对象和属性，它实质上是跨越整个Web应用程序，包括多个页面、请求和会话的一个全局作用域。

JSP 乱码如何解决？

Cookie 和 Session 的的区别？

Session是在服务器端保存用户信息，Cookie是在客户端保存用户信息。

Session中保存的是任意对象，Cookie保存的是字符串

Session随会话结束而关闭，Cookie可以长期保存在客户端硬盘上，也可以临时保存在浏览器内存中。

Cookie通常用于保存不重要的用户信息，重要的信息使用session保存。

什么是 Tomcat？

Tomcat是由Apache开发的一个Servlet容器，实现了对Servlet和JSP的支持，并提供了作为Web服务器的一些特有功能，如Tomcat管理和控制平台、安全域管理和Tomcat阀等。Tomcat包含了一个配置管理工具，也可以通过编辑XML格式的配置文件来进行配置。

详细描述 MVC

M：model模型层 javabean负责数据传递

V：view视图 页面jsp,html

C:controller控制层 负责主要逻辑调用Servlet

过程：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 发送请求 | |  |  |  |  | |  |  |  |  |
| |  | | --- | | 页面 | |  | Servlet |  | service | |  | | --- | |  | | |  |  | dao层 |  |
|  |  |  |  |  |  | |  |  |  |  |
| 响应控制层 | |  | 业务逻辑处理层 | | |  |  | 数据库操作 |  |