

### 1、解释一下什么是 servlet;

Servlet 是用来处理客户端请求并产生动态网页内容的 Java 类。Servlet 主要是用来处理或者是存储 HTML 表单提交的数据，产生动态内容，在无状态的 HTTP 协议下管理状态信息。

### 2、说一说 Servlet 的生命周期?

答:servlet 有良好的生存期的定义，包括加载和实例化、初始化、处理请求以及服务结束。

这个生存期由 javax.servlet.Servlet 接口的 init, service 和 destroy 方法表达。

Servlet 在第一次被访问的时候，加载、实例化、初始化，调用 init 方法，之后的每次访问仅仅只做请求的处理。

请求到达时运行其 service 方法，service 方法自动调用与请求对应的 doGet 或 doPost 方法。

当服务器正常关闭的时候，会对所有的 Servlet 进行销毁，调用 destroy 方法。

服务器在运行的过程中，他只会维护一个 Servlet 的实例。用这个实例来处理所有用户的请求。

### 3、Servlet 的基本架构

```
public class ServletName extends HttpServlet {  
    public void doPost(HttpServletRequest request,HttpServletResponse response) throws  
        ServletException, IOException {  
    }  
    public void doGet(HttpServletRequest request,HttpServletResponse response) throws  
        ServletException, IOException {  
    }  
}
```

### 4、response.sendRedirect 与 request.getRequestDispatcher.forward 的区别:

1、请求转发是在容器内部实现的同一个 Web 应用程序的重定向，所以 forward 方法只能重定向到同一个 Web 应用程序中的一个资源，重定向后浏览器地址栏 URL 不变，而 sendRedirect 方法可以重定向到任何 URL，因为这种方法是修改 http 头来实现的，URL 没什么限制，重定向后浏览器地址栏 URL 改变。

2、请求转发前后页面共享一个 request，response.sendRedirect() 是重新定向，前后页面不是一个 request。所以请求转发可以通过 request.setAttribute() 来传递参数，sendRedirect 只能通过 URL 重写传递数据。

### 5、什么情况下调用 doGet() 和 doPost()?

Jsp 页面中的 FORM 标签里的 method 属性为 get 时调用 doGet()，为 post 时调用 doPost()。

### 6、get 和 post 的区别

1.get 是从服务器上获取数据，post 是向服务器传送数据。

2.get 会把参数数据队列加到请求的 URL 后面，在 URL 中可以看到。

3.post 方法通过把请求参数值放在请求体中，一起传送到 ACTION 属性所指的 URL 地址。

4.get 传送的数据量较小，post 传送的数据量较大

5.get 安全性非常低，post 安全性较高。但是执行效率却比 Post 方法好。

建议:

- 1、get 方式的安全性较 Post 方式要差些, 包含机密信息的话, 建议用 Post 数据提交方式;
- 2、在做数据查询时, 建议用 Get 方式; 而在做数据添加、修改或删除时, 建议用 Post 方式;

## 7、Request 对象的主要方法

setAttribute(String name,Object): 设置名字为 name 的 request 的参数值

getAttribute(String name): 返回由 name 指定的属性值

getAttributeNames(): 返回 request 对象所有属性的名字集合, 结果是一个枚举的实例

getCookies(): 返回客户端的所有 Cookie 对象, 结果是一个 Cookie 数组

getParameter(String name): 获得客户端传送给服务器端的有 name 指定的参数值

getParameterNames(): 获得客户端传送给服务器端的所有参数的名字, 结果是一个枚举的实例

getParameterValues(String name): 获得有 name 指定的参数的所有值

getSession([Boolean create]): 返回和请求相关 Session

## 8、request.getAttribute()和 request.getParameter()有何区别?

getParameter 得到的都是 String 类型的。

getAttribute 则可以是对象。

getParameter()是获取 POST/GET 方式传递的参数值;

getAttribute()是获取对象容器中的数据值;

getParameter: 用于在用表单或 url 重写传值时接收数据用。

getAttribute: 用于服务器端重定向时, 即在 sevellet 中使用了 forward 函数,或 struts 中使用了 mapping.findForward。getAttribute 只能收到程序用 setAttribute 传过来的值。

可以用 setAttribute,getAttribute 发送接收对象.而 getParameter 显然只能传字符串。

## 9、jsp 有哪些内置对象?作用分别是什么?

request 用户端请求, 此请求会包含来自 GET/POST 请求的参数

response 网页传回用户端的回应

pageContext 网页的属性是在这里管理

session 与请求有关的会话期

application Web 应用对象

out 对应当前响应对象的输出流

config 与当前 JSP 页面的 Servlet 对应的 ServletConfig 对象

page JSP 网页本身

exception 针对错误网页, 仅当<%@pageisErrorPage=" True" %>时有效

## 10、jsp 有哪些动作?作用分别是什么?

jsp:include: 在页面被请求的时候引入一个文件。

jsp:useBean: 寻找或者实例化一个 JavaBean。

jsp:setProperty: 设置 JavaBean 的属性。

jsp:getProperty: 输出某个 JavaBean 的属性。

jsp:forward: 把请求转到一个新的页面。

jsp:plugin: 根据浏览器类型为 Java 插件生成 OBJECT 或 EMBED 标记

### 11、jsp 指令有哪些?

三种 JSP 指令: page, include, taglib

包含指令(Include): 用来包含文件和合并文件内容到当前的页面。

页面指令(Page): 用来定义 JSP 页面中特定的属性, 比如错误页面和字符编码。

Taglib 指令: 用来声明页面中使用的自定义的标签库。

### 12、JSP 中动态 include 与静态 include 的区别?

答: 动态 include 用 `jsp:include` 动作实现, `<jsp:include page=included.jsp flush=true />` 它总是会检查所含文件中的变化, 适合用于包含动态页面, 并且可以带参数。

静态 include 用 `include` 指令实现, 不会检查所含文件的变化, 适用于包含静态页面  
`<%@include file=included.htm %>`

### 13、页面间对象传递的方法

用 request, session, application, cookie 等。

### 14、什么是JSP 页面?

JSP 页面是一种包含了静态数据 HTML 和 java 元素两种类型的文本的文本文档。

### 15、JSP 和 Servlet 有哪些相同点和不同点, 他们之间的联系是什么?

JSP 是 Servlet 技术的扩展。JSP 编译后是 "Servlet"。

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 是 Java 和 HTML 组合成的一个扩展名为 .jsp 的文件。

JSP 侧重于视图, Servlet 主要用于控制逻辑。

### 16、JSP 请求是如何被处理的?

浏览器首先要请求一个以 .jsp 扩展名结尾的页面, 发起 JSP 请求, 然后, Web 服务器读取这个请求, 使用 JSP 编译器把 JSP 页面转化成一个 Servlet 类。需要注意的是, 只有当第一次请求页面或者是 JSP 文件发生改变的时候 JSP 文件才会被编译, 然后服务器调用 servlet 类, 处理浏览器的请求。一旦请求执行结束, servlet 会把响应发送给客户端。

### 17、JSP 有什么优点?

JSP 页面是被动态编译成 Servlet 的, 因此, 开发者可以很容易的更新展现代码。

JSP 页面可以被预编译。

JSP 页面可以很容易的和静态模板结合, 包括: HTML 或者 XML, 也可以很容易的和产生动态内容的代码结合起来。

开发者可以提供让页面设计者以类 XML 格式来访问的自定义的 JSP 标签库。

开发者可以在组件层做逻辑上的改变, 而不需要编辑单独使用了应用层逻辑的页面。

### 18、MVC 的各个部分都有那些技术来实现?如何实现?

Model 模型层, 主要负责出来业务逻辑以及数据库的交互 (通过 JavaBean, EJB 组件, hibernate, spring 等实现)

View 视图层, 主要用于显示数据和提交数据 (由 JSP+JSTL 或 Struts 标签+HTML 等实现)

Controller 控制器, 主要是用作捕获请求并控制请求转发 (由 Servlet 或 Struts2 的 action 实现)

### 19、什么是 cookie? session 和 cookie 有什么区别?

cookie 是 Web 服务器发送给浏览器的一块信息。浏览器会在本地文件中给每一个 Web 服务器存储 cookie。以后浏览器在给特定的 Web 服务器发请求的时候, 同时会发送所有为该服务器存储的 cookie。

session 和 cookie 的区别:

1. session 存在于服务器端。Cookie 存在于客户端。
2. session 能够存储任意的 Java 对象, cookie 只能存储 String 类型数据。
3. cookie 不是很安全, 别人可以分析存放在本地的 cookie 并进行 cookie 欺骗, 考虑到安全应当使用 session。
4. session 会在一定时间内保存在服务器上。当访问增多, 会比较占用你服务器的性能, 考虑到减轻服务器性能方面, 应当使用 cookie。
5. 单个 cookie 保存的数据不能超过 4K, 很多浏览器都限制一个站点最多保存 20 个 cookie。
6. cookie 的存在时间一般比较长, session 一般较短, 浏览器窗口关闭时 session 失效