### javaweb

#### HTTP请求的GET与POST方式的区别？

答：理论上：get一般用于获取/查询资源信息，post一般用于更新数据。但是在实际开发中没有这么严格的要求。

区别：

（1）get请求的请求参数会附在URL后，以?分割；post请求参数包含在post的请求体中。

（2）get方式提交的数据是限制的（各大浏览器规定不一样），post没有限制。

（3）post提交方式比get方式提交安全。

#### 状态码

|--200：请求已成功，请求所希望的响应头或数据体将随此响应返回。

|--302：重定向

|--304：走缓存

|-- 404：请求的资源找不到

|--500：代码错误

#### 解释一下什么是servlet?

Servlet是javaweb的三大组件之一（servlet,filter,lisener）,属于动态资源。servlet的作用处理请求，服务器会把接收到的请求交给servlet来处理，servlet中通常需要：

|--接收请求数据；

|--处理请求；

|--完成响应。

#### 说一说Servlet的生命周期?

服务器会在servlet第一次被访问的时候创建servlet，一个servlet类，服务器只会创建一个实例对象。在servlet被创建后，服务器会调用init()方法，这个方法之后被调用一次。当服务器每次接收请求的时，都会调用service()方法。这个方法是多次被调用的。服务器关闭servlet才会离去。在服务器关闭前调用destory()方法。

注意：如果想在服务器启动的时候创建servlet，需要在web.xml中添加<load-on-startup>

#### 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 {

}

}

#### SERVLET API中forward() 与redirect()的区别？

（1）请求转发是一个请求，而重定向是两个请求；

 （2）请求转发后浏览器地址栏不会有变化，而重定向会有变化，因为重定向是两个请求；

 （3）请求转发的目标只能是本应用中的资源，重定向的目标可以是其他应用；

 （4）请求转发对AServlet和BServlet的请求方法是相同的，即要么都是GET，要么都是POST，因为请求转发是一个请求；

 （5）重定向的第二个请求一定是GET；

#### forward 和redirect的区别

forward是服务器请求资源，服务器直接访问目标地址的URL，把那个URL的响应内容读取过来，然后把这些内容再发给浏览器，浏览器根本不知道服务器发送的内容是从哪儿来的，所以它的地址栏中还是原来的地址。

redirect就是服务端根据逻辑,发送一个状态码,告诉浏览器重新去请求那个地址，一般来说浏览器会用刚才请求的所有参数重新请求，所以session,request参数都可以获取。

#### jsp有哪些内置对象?作用分别是什么? 分别有什么方法？

JSP共有以下9个内置的对象：

|--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 servlet正在执行的内容

|--out 用来传送回应的输出

|--config servlet的构架部件

|--page JSP网页本身

|--exception 针对错误网页，未捕捉的例外

request表示HttpServletRequest对象。它包含了有关浏览器请求的信息，并且提供了几个用于获取cookie, header, 和session数据的有用的方法。

response表示HttpServletResponse对象，并提供了几个用于设置送回 浏览器的响应的方法（如cookies,头信息等）

out对象是javax.jsp.JspWriter的一个实例，并提供了几个方法使你能用于向浏览器回送输出结果。

pageContext表示一个javax.servlet.jsp.PageContext对象。它是用于方便存取各种范围的名字空间、servlet相关的对象的API，并且包装了通用的servlet相关功能的方法。

session表示一个请求的javax.servlet.http.HttpSession对象。Session可以存贮用户的状态信息

applicaton 表示一个javax.servle.ServletContext对象。这有助于查找有关servlet引擎和servlet环境的信息

config表示一个javax.servlet.ServletConfig对象。该对象用于存取servlet实例的初始化参数。

page表示从该页面产生的一个servlet实例

#### JSP中动态INCLUDE与静态INCLUDE的区别？

动态INCLUDE用jsp:include动作实现

<jsp:include page=included.jspflush=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。运行时将页面包含。

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.htm %>。编译时将页面包含。

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

servlet和jsp都是服务器的组件。Servlet是一个接口，也是SUN公司提出的一种用户和WEB容器之间通信的标准。由用户实现其中的service()方法供WEB容器进行调用，从而实现servlet和WEB容器之间的交互。所以当用户要和WEB容器通信时必须实现这种标准。而JSP规范规定，由容器翻译好的JAVA类必须实现HttpJspPage接口，而这个接口是servlet的子接口，从这个意义上说，JSP的本质还是servlet。

JSP重在表示，解决了servlet页面输出困难的问题。而servlet重在业务处理，避免在页面出现过多的业务处理带来的阅读性和维护性的困难。它们可以很好的结合。 JSP侧重于视图，Servlet主要用于控制逻辑。在struts框架中,JSP位于MVC设计模式的视图层,而Servlet位于控制层。

Servlet编写后需要在WEB应用的web.xml进行注册，从而能让WEB容器识别用户编码的Servlet。但JSP由容器来管理，所以无需注册。jsp在编译后成一个servlet。

#### MVC的各个部分都有那些技术来实现?如何实现?

MVC是Model－View－Controller的简写。Model 代表的是应用的业务逻辑（通过JavaBean，EJB组件实现）， View 是应用的表示面（由JSP页面产生），Controller 是提供应用的处理过程控制（一般是一个Servlet），通过这种设计模型把应用逻辑，处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用。

#### session的工作原理？

当首次访问服务端时，服务器调用一个getSession()方法,创建一个session，session保存在服务端，session的id保存在客户端（cookie 中保存这个id）。

当第二次访问时，客户端会携带这个id，服务端通过这个id找到对应的session，如果服务端有这个session，就无需创建新的session。

#### 禁用Cookies后，Session还可以使用吗？

可以，使用URL重写。在每个页面中的每个链接和表单中都添加名为jSessionId的参数，值为当前sessionid。当用户点击链接或提交表单时也服务器可以通过获取jSessionId这个参数来得到客户端的sessionId，找到sessoin对象。

#### session与cookie的区别？

（1）cookie数据存放在客户的浏览器上，session数据放在服务器上。

（2）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。

（3）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。

（4）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。

（5）session保存的是对象，cookie中保存的是字符串。

#### 过滤器和拦截器的区别？

（1）拦截器是基于java的反射机制的，而过滤器是基于函数回调

（2）过滤器依赖于servlet容器，而拦截器不依赖于servlet容器

（3）拦截器只能对action请求起作用，而过滤器则可以对几乎所有的请求起作用

（4）拦截器可以访问action上下文、值栈里的对象，而过滤器不能

（5）在action的生命周期中，拦截器可以多次被调用，而过滤器只在容器初始化时调用一次

#### 监听器是什么?

Servlet监听器用于监听一些重要事件的发生，监听器对象可以在事情发生前、发生后可以做一些必要的处理。

1.Listener是Servlet的监听器

2.可以监听客户端的请求、服务端的操作等。

3.通过监听器，可以自动激发一些操作，如监听在线用户数量,当增加一个HttpSession时，给在线人数加1。

4.编写监听器需要实现相应的接口

5.编写完成后在web.xml文件中配置一下,就可以起作用了

6.可以在不修改现有系统基础上,增加web应用程序生命周期事件的跟踪

#### Filter生命周期？

服务器启动时会创建Filter实例，并且每个类型的Filter只创建一个实例，从此不再创建！在创建完Filter实例后，会马上调用init()方法完成初始化工作，这个方法只会被执行一次；每次用户访问目标资源的时候都会访问doFilter(),如果需要放行要调用FilteChain的doFilter()，如果不调用FilterChain的doFilter()方法，目标资源将无法被执行。服务器会在创建Filter对象之后，把Filter放到缓存中一直使用，通常不会销毁它。一般会在服务器关闭时销毁Filter对象，在销毁Filter对象之前，服务器会调用Filter对象的destory()方法。