面试题——Java Web

1. jsp 和 servlet 有什么区别？

答：jsp经编译后就成了Servlet（Jsp的本质就是Servlet，JVM只能识别Java类，不能识别Jsp代码，Web容器将Jsp的代码编译成JVM能够识别的java类）

jsp更擅长表现于页面显示，servlet更擅长于逻辑控制

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

2. jsp 有哪些内置对象？作用分别是什么？

答：Page，pageContext，request，response，session，application，out，config，exception

Page指的是JSP被翻译成Servlet的对象的引用.

pageContext对象可以用来获得其他8个内置对象,还可以作为JSP的域范围对象使用.pageContext中存的值是当前的页面的作用范围

request代表的是请求对象,可以用于获得客户机的信息,也可以作为域对象来使用，使用request保存的数据在一次请求范围内有效。

Session代表的是一次会话，可以用于保存用户的私有的信息,也可以作为域对象使用，使用session保存的数据在一次会话范围有效

Application：代表整个应用范围,使用这个对象保存的数据在整个web应用中都有效。

Response是响应对象,代表的是从服务器向浏览器响应数据.

Out:JSPWriter是用于向页面输出内容的对象

Config：指的是ServletConfig用于JSP翻译成Servlet后 获得Servlet的配置的对象.

Exception:在页面中设置isErrorPage=”true”，即可使用，是Throwable的引用.用来获得页面的错误信息。

3. 说一下 jsp 的 4 种作用域？

答：

|  |  |
| --- | --- |
| application | 在所有应用程序中有效 |
| session | 在当前会话中有效 |
| request | 在当前请求中有效 |
| page | 在当前页面有效 |

4. session 和 cookie 有什么区别？

答：存储位置不同：session 存储在服务器端；cookie 存储在浏览器端。

安全性不同：cookie 安全性一般，在浏览器存储，可以被伪造和修改。

容量和个数限制：cookie 有容量限制，每个站点下的 cookie 也有个数限制。

存储的多样性：session 可以存储在 Redis 中、数据库中、应用程序中；而 cookie 只能存储在浏览器中

5. 说一下 session 的工作原理？

答：session 的工作原理是客户端登录完成之后，服务器会创建对应的 session，session 创建完之后，会把 session 的 id 发送给客户端，客户端再存储到浏览器中。这样客户端每次访问服务器时，都会带着 sessionid，服务器拿到 sessionid 之后，在内存找到与之对应的 session 这样就可以正常工作了

6. 如果客户端禁止 cookie 能实现 session 还能用吗？

答：可以用，session 只是依赖 cookie 存储 sessionid，如果 cookie 被禁用了，可以使用 url 中添加 sessionid 的方式保证 session 能正常使用

7. spring mvc 和 struts 的区别是什么？

答：

拦截级别：struts2 是类级别的拦截；spring mvc 是方法级别的拦截。

数据独立性：spring mvc 的方法之间基本上独立的，独享 request 和 response 数据，请求数据通过参数获取，处理结果通过 ModelMap 交回给框架，方法之间不共享变量；而 struts2 虽然方法之间也是独立的，但其所有 action 变量是共享的，这不会影响程序运行，却给我们编码和读程序时带来了一定的麻烦。

拦截机制：struts2 有以自己的 interceptor 机制，spring mvc 用的是独立的 aop 方式，这样导致struts2 的配置文件量比 spring mvc 大。

对 ajax 的支持：spring mvc 集成了ajax，所有 ajax 使用很方便，只需要一个注解 @ResponseBody 就可以实现了；而 struts2 一般需要安装插件或者自己写代码才行。

8. 如何避免 sql 注入？

答：

使用预处理 PreparedStatement。

使用正则表达式过滤掉字符中的特殊字符。

9. 什么是 XSS 攻击，如何避免？

答：

XSS 攻击：即跨站脚本攻击，它是 Web 程序中常见的漏洞。原理是攻击者往 Web 页面里插入恶意的脚本代码（css 代码、Javascript 代码等），当用户浏览该页面时，嵌入其中的脚本代码会被执行，从而达到恶意攻击用户的目的，如盗取用户 cookie、破坏页面结构、重定向到其他网站等。

预防 XSS 的核心是必须对输入的数据做过滤处理。

10. 什么是 CSRF 攻击，如何避免？

答：

CSRF：Cross-Site Request Forgery（中文：跨站请求伪造），可以理解为攻击者盗用了你的身份，以你的名义发送恶意请求，比如：以你名义发送邮件、发消息、购买商品，虚拟货币转账等。

防御手段：

验证请求来源地址；

关键操作添加验证码；

在请求地址添加 token 并验证；