# JavaWeb

## Tomcat的优化经验

答：去掉对web.xml的监视，把JSP提前编辑成Servlet；有富余物理内存的情况下，加大Tomcat使用的JVM内存。

## 什么是Servlet？

答：可以从两个方面去看Servlet：

a、API：有一个接口servlet，它是servlet规范中定义的用来处理客户端请求的程序需要实现的顶级接口。

b、组件：服务器端用来处理客户端请求的组件，需要在web.xml文件中进行配置。

## Servlet API中forware()和redirect()的区别？

答：

a、浏览器显示

重定向会改变URL地址，请求转发不会改变URL地址。

b、资源共享

重定向不可以资源共享，请求转发可以。

c、功能

重定向可以用URL绝对路径访问其他Web服务器的资源，而请求转发只能在一个Web应用程序内进行资源转发，即服务器内部的一种操作。

d、效率

重定向效率低，相当于再一次请求；请求转发效率相对较高，跳转仅发生在服务器端。

## request.getParameter()和request.getAttribute()的区别？

答：

a、request.getParameter()获取的类型是String；

request.getAttribute()获取的类型是Object

b、request.getPrameter()获取的是POST/GET传递的参数值和URL中的参数；

request.getAttribute()获取的是对象容器中的数据值/对象

c、request.setAttribute()和request.getAttribute()可以发送、接收对象；

request.getParamter()只能接收字符串，官方不开放request.setParamter()（也就是没有这个方法）

setAttribute()和getAttribute()的传参原理：

setAttribute()是应用服务器把这个对象放在该页面所对应的一块内存中去，当你的页面服务器重定向到另外一个页面时，

应用服务器会把这块内存拷贝到另一个页面所对应的那块内存中。这个就可以通过getAttribute()获取到相应的参数值或者对象。

## JSP有哪些动作？作用分别是什么？

答：JSP一共有以下6中基本动作

JSP：include （当页面被请求的时候引入一个文件）

JSP：forward （将请求转到另一个页面）

JSP：useBean （获得JavaBean的一个实例）

JSP：setProperty （设置JavaBean的属性）

JSP：getProperty （获得JavaBean的属性）

JSP：plugin （根据浏览器类型为Java插件生成object或者embed两种标记）

## JSP常用指令有哪些？

答：page、include、taglib

a、page指令：定义页面的一些属性，常用属性：

contentType="text/html;charset=utf-8"; 向浏览器端输出数据的编码

pageEncoding="utf-8"; JSP编译成java文件时所用的编码

session="true" 是否自动创建session

b、include指令：引入一个静态的JSP页面

c、taglib指令：引入一个标签库

## 页面间对象传递的方法？

答：request、session、application、cookie等。

## XML语法

1 每个XML文档都由XML序言开始，在前面的代码中的第一行便是XML序言，<?xml version="1.0"?>

2 任何的起始标签都必须有一个结束标签。

3 标签必须按合适的顺序进行嵌套，所以结束标签必须按镜像顺序匹配起始标签。

4 标签如果有属性，属性值必须加双引号。

# XML解析四种方法

## DOM方法

Java自身原生的两种解析XML方式之一——DOM方法，原理是：首先在内存中创建一个Document对象，然后把XML文档读取进来赋值给这个dom对象。由于dom对象是基于树结构的，所以对dom对象进行遍历即可。对内存中的dom对象可以进行查询、修改、删除操作，还可以写回原XML文档保存修改。

优点：

a、由于整棵树在内存中，因此可以对xml文档随机访问

b、可以对xml文档进行修改操作

缺点：

a、整个文档必须一次性解析完

a、由于整个文档都需要载入内存，对于大文档成本高

总结：DOM方法解析XML文档，把元素、元素属性、元素值都看作Node类型，通过node.getNodeName()获取元素名、属性名，

通过getNodeValue()获取属性值、元素值，通过getChildNodes()获取子节点们，通过item(i)获取第i个属性或者第i个子节点。

## SAX方法

Java原生的XML解析方法之二——SAX方法，原理：通过parse(file,listener)函数用一个listener对xml文件进行查找，按顺序读取文档，遍历每个标签，当发现目标标签

时，读取标签的属性、结点值等信息并返回。

优点：

a、无需将整个xml文档载入内存，因此消耗内存少

b、可以继承ContentHandler创建多个执行不同查询的listener进行解析操作

缺点：

a、不能随机的访问xml中的节点

b、不能修改文档

c、查询依次就要对XML文档从头到尾遍历一次

总结：SAX解析XML文档的结点名是通过事件函数的参数qName获取的，属性是通过参数attributes的getValue("属性名")获取的，

结点值是通过当前事件函数发生时，characters(char[] ch, int start, int length)方法中的内容获取的。

## JDOM方法

JDOM方法是根据DOM方法的众多繁琐操作进行包装得到的，上面我们看到，DOM方法解析XML文档其实是很繁琐的，而且很混乱，标签、属性、换行空格都当作结点类型来处理。JDOM方法定义了一系列通俗、好记的方法来解析XML，方法的底层封装了一系列DOM操作，但是我们不必亲自去进行这些繁琐的工作了。

优点：

a、DOM方式的优点:查找方便，可以修改

缺点

a、DOM方式的缺点:装载整个文档,对内存容量要求高

在JDOM中，同一了根节点、普通结点、属性等全为Element类型。

## DOM4J方法

Dom4j是目前最流行、最好用的XML解析工具，解析XML的速度最快。

操作步骤：

1：创建SAXReader：SAXReader reader = new SAXReader();

2：创建文件输入流打开xml文件：InputStream in = new FileInputStream("XXX.xml");

3：通过reader和输入流读取xml文件到内存创建Document对象：Document dom = reader.read(in);

4：获取根节点：Element root=dom.getRootElement();

5：获取子节点列表：List<Element> childNodes = root.elements();

6：遍历子节点：Element node = childNodes.get(i);

7：读取结点信息：

1）结点属性值：node.attributeValue("属性名")；

2）结点名：node.getName()；

3）结点值：node.getValue();

4）子结点文本值：node.elementText("子结点名")

总结：

1）DOM、JDOM、DOM4j都是把xml文档读取到内存中，生成dom对象进行遍历的；

DOM是Java原生的，所以比较繁琐；

JDOM是对DOM操作的封装，更加通俗、易记，操作也快了一点；

DOM4j解析xml的函数上与JDOM差不多，只不过有几个相同功能的函数名字不同而已，过程都是一样的；但由于底层使用了Xpath等方法加快了索引，所以检索性能更快。

2）SAX是基于事件驱动的，查询事件监听器继承自DefaultHandler，定义了检索xml过程中遇到开始标签、结束标签时执行的事件函数，从而查找需要的信息并返回而不是把整个文档都加载进来。