



【济南中心】JavaEE就业班同步笔记第一阶段：JavaWeb之XML&Tomcat

小鲁哥哥 • 2016-12-29 21:37:03

[点击查看济南黑马校区最新开班计划](#)

【济南中心】JavaEE就业班同步笔记第一阶段： JavaWeb之核心技术--XML&Tomcat

1 案例一：使用XML作为配置文件的方式完成模拟Tomcat.

1.1 需求：

现在有一套页面,这套页面想被其他人访问到.端口号可以进行修改的-(不修改源码的基础上).

1.2 分析：

1.2.1 技术分析：

【XML的概述】

什么是XML：

什么是 XML?

- XML 指可扩展标记语言 (EXtensible Markup Language)
- XML 是一种标记语言，很类似 HTML
- XML 的设计宗旨是传输数据，而非显示数据
- XML 标签没有被预定义。您需要自行定义标签。
- XML 被设计为具有自我描述性。
- XML 是 W3C 的推荐标准

[回帖](#)

XML的作用：

- * 传输 和 存取数据.
- * 软件的配置文件.

XML的使用:

* XML的语法.

【XML的语法】

XML的基本语法:

* XML必须有关闭标签

* 区分大小写

* 属性需要有引号

* 标签必须正确嵌套.

XML的文档声明

* 文档声明: 通常出现在XML的第一行第一列的位置!!!

* 写法:

* `<?xml 属性名="属性值" 属性名="属性值" ...?>`

* `version` :必须的. 使用"1.0"

* `encoding` :字符集. 是使用浏览器打开的时候采用的默认的字符集的编码.

* `standalone` :描述XML文档是否需要依赖其他的文件.

XML的注释

* `<!-- XML的注释 -->`

XML的元素

* 元素 (标签) 的命名规范:

XML 命名规则

XML 元素必须遵循以下命名规则:

- 名称可以含字母、数字以及其他的字符
- 名称不能以数字或者标点符号开始
- 名称不能以字符 "xml" (或者 XML、Xml) 开始
- 名称不能包含空格

***** 名称中不能包含冒号(:)

XML的属性

* 属性的名称规范与元素一致.

***** 属性需要使用引号!

XML的特殊字符和CDATA区

* XML的特殊字符:

<	<	小于
>	>	大于
&	&	和号
'	'	单引号
"	"	引号

* XML的CDATA区:(CDATA:Character Data)

* <![CDATA[内容]]>

【XML的解析】

XML的解析:从XML文档中获得想要的数据(通过代码完成的.)

XML的解析的方式(****):

* DOM解析:DOM:Document Object Model.

* SAX解析:Simple Api for XML.

* DOM和SAX的区别:

dom:一次性将文档加载到内存,形成树形结构进行解析

缺点:文档特别大易造成内存溢出

优点:对xml进行增删改操作比较有利

sax:事件驱动方式,一行一样进行解析

缺点:不能对文档进行增删改

优点:如果文档特别大不会造成内存溢出

针对这两种解析的方式,不同的公司提供了不同的API的实现.

* JAXP :SUN公司提供的一套XML的解析的API.

* JDOM :开源组织提供了一套XML的解析的API-jdom.

* DOM4J :开源组织提供了一套XML的解析的API-dom4j.

* pull :主要应用在Android手机端解析XML.

□DOM4J的入门案例步骤:

* 【步骤一】导入jar包.dom4j-1.6.1.jar

* 【步骤二】创建解析器

* 【步骤三】解析文档获得代表文档的Document对象.

* 【步骤四】获得跟节点.

* 【步骤五】从跟节点下查找其他的节点.

* 代码实现:

```
[mw_shl_code=java,true]@Test
```

```

/**
 * 获得元素的内容:查询的操作.
 */
public void demo1() throws Exception{
    // 创建解析器
    SAXReader reader = new SAXReader();
    // 解析XML的文档
    Document document = reader.read("xml/demo1.xml");
    // 获得跟节点
    Element root = document.getRootElement();
    System.out.println(root.getName());
    // 查找跟节点下的子节点. element() elements();
    Element pElement = root.element("person"); // 查找的是第一个person元素
    // root.elements("person").get(1); // 查找的是第二个person元素
    Element nElement = pElement.element("name");
    Element aElement = pElement.element("age");
    Element sElement = pElement.element("sex");
    System.out.println(nElement.getText());
    System.out.println(aElement.getText());
    System.out.println(sElement.getText());
}[/mw_shl_code]

```

XPath:

- * dom4j支持XPath的jar包.
 - * jaxen-1.1-beta-6.jar
- * dom4j的XPath支持的API:
 - * List document.selectNodes(String xPath);
 - * Node document.selectSingleNode(String xPath);
- * 代码:

```

[/mw_shl_code=java,true]@Test
/**
 * DOM4J的XPath的写法:
 */
public void demo2() throws Exception{
    // 创建解析器:
    SAXReader reader = new SAXReader();
    // 解析XML返回Document对象.

```

```

Document document = reader.read("xml/demo1.xml");
/*List<Node> list = document.selectNodes("//name");
for (Node node : list) {
    Element element = (Element) node;
    System.out.println(element.getText());
}*/

List<Node> list = document.selectNodes("//person['@id']");
for (Node node : list) {
    Element element = (Element) node;
    System.out.println(element.attributeValue("id"));
}

```

}[mw_shl_code]1.3 代码实现:

【步骤一】：将基础班的Socket编写的代码复制到工程中.

【步骤二】：新建xml.在conf/server.xml

【步骤三】：使用DOM4J解析XML

```

[mw_shl_code=java, true]public class MyServer {
    public static void main(String[] args) throws Exception {
        // 解析XML:
        // 创建解析器
        SAXReader reader = new SAXReader();
        // 解析XML文档:
        Document document = reader.read("conf/server.xml");
        Element element = (Element)document.selectSingleNode("//Connector['@port']");
        String port = element.attributeValue("port");
        // ServerSocket 对象可以监听端口
        ServerSocket serversocket = new ServerSocket(Integer.parseInt(port));
        while(true) {
            Socket socket = serversocket.accept();    // 等待客户端的连接请求，一旦有请求过来，
就结束阻塞，返回客户端对象
            //System.out.println(socket.getInetAddress());
            // 一旦有客户来访问，就另开一个新线程去提供服务，main线程继续等待下一个客户的连
接
            new Thread(new MyService(socket)).start();
        }
    }
}

```

}[mw_shl_code]

1.4 总结:

1.4.1 XML的约束:

XML的约束的概述:

* 什么是XML的约束:就是用来约束XML的文档中可以出现哪些标签, 不能出现哪些标签, 标签中是否有顺序, 出现的次数.

* XML的约束:用来规范XML的写法.

***** XML的约束的种类及区别?

* DTD 和 Schema :

* 区别:

1.DTD语法是自成一体的.Schema语法就是XML的语法.

2.Schema的语法就是XML的语法所以更容易被解析器所解析.

3.Schema支持名称空间.

4.Schema有比DTD更加强大的语义和语法的约束.

DTD的语法:(了解)

* DTD的引入方式:

* 内部的DTD:

[mw_shl_code=xml,true]<!DOCTYPE persons [

]>[/mw_shl_code] * 外部的DTD:

* 一种本地DTD:

[mw_shl_code=xml,true]<!DOCTYPE persons SYSTEM "unknown.dtd">[/mw_shl_code]

* 一种网络DTD:

[mw_shl_code=xml,true]<!DOCTYPE persons PUBLIC "-//UNKNOWN/" "unknown.dtd">[/mw_shl_c

ode]

* DTD的语法:

* 元素:

* <!ELEMENT 元素名 元素类型>

* 元素类型:

* EMPTY

* ANY

* 子元素

* 是否有序: 使用 逗号(,)或者竖线(|)表示.

* 出现的次数: ?:零次或一次 +: 一次或多次 *: 零次或多次

* PCDATA

* 属性:

* <!ATTLIST 元素名称 属性名称 属性的类型 属性的约束>

* 属性的类型:

* ID类型:表示属性值需要是唯一的.

* CDATA类型:普通的字符串.

* 枚举:

* 属性的约束:

值	解释
值	属性的默认值
#REQUIRED	属性值是必需的
#IMPLIED	属性不是必需的
#FIXED value	属性值是固定的

Schema的语法:(了解)

```
[mw_shl_code=xml,true]<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

名称空间:一个XML只能引入一个DTD约束文档.使用了Schema约束XML文档,一个XML可以引入多个Schema的约束!!!

如果再多个Schema文档中定义了相同的属性名称 该怎么办?

* 名称空间类似于java中的package.通过名称空间区分 标签或属性来自于哪个文档的!!! 通常名称空间唯一的不重复的即可.一般情况下使用一个URL地址表示一个名称空间.

xmlns :xml name sapace .代表当前的文档应用的名称空间.

targetNamespace :目标名称空间.

elementFormDefault :

```
-->
```

```
<schema
```

```
xmlns="http://www.w3.org/2001/XMLSchema"
```

```
targetNamespace="http://www.itheima.com/ee25"
```

```
elementFormDefault="qualified">
```

```
<!-- 复杂标签 -->
```

```
<element name="persons">
```

```

<!-- 复杂类型 -->
<complexType>
  <sequence maxOccurs="unbounded" minOccurs="1">
    <element name="person">
      <complexType>
        <sequence>
          <!-- 简单标签 -->
          <element name="name" type="string"/>
          <element name="age" type="int"/>
          <element name="sex" type="string"/>
        </sequence>
        <attribute name="id" type="string" use="required"/>
      </complexType>
    </element>
  </sequence>
</complexType>
</element>
</schema>[/mw_shl_code]

```

2 案例二：使用Tomcat发布一个WEB项目.

2.1 需求:

有一个web项目,需要通过tomcat服务器被其他人访问到.

2.2 分析:

2.2.1 技术分析:

【软件的架构】

C/S架构的软件:

* Client / Server 客户端和服务端端的软件. 都是需要在PC端安装的软件. 比如 QQ,迅雷,暴风..

.

* 优点:效果炫,一部分代码写到客户端(速度快).

* 缺点:服务器端更新,客户端都需要更新.

B/S架构的软件:

* Browser / Server 浏览器端和服务端端的软件. 不需要安装到PC端,只需要有一个浏览器即可. 比如 京东,网银,WEBQQ,WEB迅雷...

* 优点:服务器端更新,客户端浏览器不需要进行更新.

* 缺点:效果不炫,所有的代码运行都在服务器端,导致服务器压力过大.

- * 效果: 使用HTML5,CSS3可以做出很炫效果.
- * 服务器端压力:搭建服务器的集群.而且还有AJAX技术.

【WEB的资源】

静态资源:

- * HTML,CSS,JS,图片...

动态资源:

- * PHP:
- * ASP:
- * Servlet/JSP:

【服务器】

硬件:其实就是一台配置很高的电脑.

软件:必须在一台电脑上安装服务器软件.这台电脑称为是WEB服务器.

【常见WEB服务器】

Apache :发布PHP的.LAMP:Linux Apache MySQL PHP.

IIS :发布ASP的.

WebSphere :IBM公司研发,收费的大型服务器软件,支持EE的所有的开发规范.

WebLogic :BEA公司研发,收发的大型服务器软件,支持EE的所有的开发规范.

Tomcat :Apache组织研发,免费的小型服务器软件,支持Servlet/JSP的开发规范.

JBoss :

【WEB动态资源目录结构】

website

|-----静态资源

|-----WEB-INF

|-----web.xml :必须的

|-----classes :可选的

|-----lib :可选的

2.3 案例实现:

2.3.1 Tomcat的安装:












【下载Tomcat】

<http://tomcat.apache.org/download-70.cgi>

【安装Tomcat】

解压即可.

【了解Tomcat的目录结构】

 bin	2016/4/26 16:06	文件夹	
 conf	2016/4/26 16:07	文件夹	
 lib	2016/4/26 16:06	文件夹	
 logs	2016/4/26 16:07	文件夹	
 temp	2016/4/26 16:06	文件夹	
 webapps	2016/4/26 16:06	文件夹	
 work	2016/4/26 16:07	文件夹	
 LICENSE	2016/4/11 10:58	文件	57 KB
 NOTICE	2016/4/11 10:58	文件	2 KB
 RELEASE-NOTES	2016/4/11 10:58	文件	9 KB
 RUNNING.txt	2016/4/11 10:58	文本文档	17 KB

- * bin :tomcat的执行的文件.
- * conf :tomcat的配置文件.
- * lib :tomcat运行的需要的jar包.
- * logs :tomcat的运行的日志文件.
- * temp :tomcat产生临时文件存放的路径.
- * webapps :tomcat发布的web项目的路径.
- * work :tomcat运行JSP的时候,JSP翻译成Servlet的代码存放的路径.

2.3.2 发布一个WEB项目到Tomcat中.

发布一个WEB工程到Tomcat中共有三种方式:

- * 1.直接将工程copy到tomcat/webapps下.
- * 2.配置tomcat虚拟路径:
- * 3.配置tomcat虚拟路径:

【方式一：直接将项目copy到webapps下】

- * 将工程复制到webapps即可.

【方式二：配置tomcat的虚拟路径-不推荐】

- * 在tomcat/conf/server.xml文件中进行配置:
- * 在<Host>标签下配置:

* <Context path="/itheima" docBase="C:/website"/>

【方式三：配置tomcat的虚拟路径-推荐】

- * 在tomcat/conf/Catalina/localhost/xxx.xml
- * 配置<Context docBase="C:/website"/>

* xxx作为虚拟路径的名称.

2.4 总结:

2.4.1 Tomcat常见问题:

【没有配置JAVA_HOME引发tomcat一闪消失】

* 一定要配置JAVA_HOME环境变量.

【端口号冲突的问题】

* 一种:将占用端口的应用结束.

* netstat -ano

* 打开任务管理器结束:

* 二种:修改自身应用的端口号.

* tomcat/conf/server.xml



回帖

— 36条回帖 —



Mr_Maty

不错，顶起

沙发 • 2016-12-29 21:41:53

回帖

来自宇宙超级黑马专属苹果客户端



@211

不错，很好的！

藤椅 • 2017-1-1 16:29:19

回帖



小陵不懂

这样的笔记真不错哦

板凳 • 2017-1-2 09:52:48

回帖



小虎同学

不错不错，顶起来

回帖

来自宇宙超级黑马专属苹果客户端



事难懂

好东西，果断收藏了

地板 • 2017-1-3 13:29:05

回帖



hittor

感谢分享!

7# • 2017-1-3 19:20:15

回帖



Yin,,,Yan

好帖子 收藏下

8# • 2017-1-10 00:11:23

回帖



185447328

asdkgjkadfhgadjhgdajhgljdkfahkjdaflhgkjdfahglida

9# • 2017-1-10 14:16:20

回帖



zmanx

我们还没开始毕业班呢,基础知识还有好多不懂的

10# • 2017-2-12 21:52:42

回帖



爱吃橘子的小泽

定一下,我们现在还在努力

11# • 2017-2-12 22:02:37

回帖



zmanx

一起加油吧,我还是小白,加油吧

12# • 2017-2-12 22:05:34

回帖



a690223483

谢谢分享

13# • 2017-2-26 11:28:14

回帖



xulinml

谢谢老师~! ~感谢感谢

14# • 2017-2-28 08:58:17

回帖



为过去的我买单

谢谢分享!

15# • 2017-3-2 10:05:18

回帖



zhouxiaoyang



16# • 2017-3-26 23:26:34

回帖



ys5214

666666666666666666

17# • 2017-3-31 11:16:51

回帖



Cerol

力挺楼主, 支持你 666

18# • 2017-4-12 17:19:41

回帖



hrywzn

学习一下,真不错

19# • 2017-5-22 22:23:54

回帖



3371906349

非常感谢楼主大大的无私奉献

20# • 2017-5-29 23:17:56

回帖

上一页

1/2

下一页