# Android基本功一XML和JSON的应用

**关键字：**

XML DTD

XML Schema

JSON

**本章难点：**

XML Schema的语法使用

## 项目简介

### 关于XML

XML**,** 即可扩展标记语言**(**eXtensible Markup Language**),**是一种用于标记电子文件使其具有结构性的标记语言。

一说起标记语言，人们常常会想起HTML。没错，作为标记语言，二者都没有任何“动作行为”，都只是通过在文本文件中添加一些标记，来传递更多的信息。

但不同于用于展示数据的HTML，XML侧重于数据如何进行储存传输。因此，它逐渐演变成了一种跨平台的数据交换格式，一种轻量级的持久化方案。通过XML，人们可以在不同平台，不同系统之间进行数据交换。同时还可以将程序状态保存到XML文件中，而无须使用关系数据库。

XML广泛应用于Java开发以及Web开发。因为Java作为跨平台编程语言，需要频繁地在不同平台和系统之间数据交换，而XML正好能给予这种支持。

同时XML也是Web技术重要基础，常用于简化数据的存储和共享被大量应用与Web开发中。

xml文档具有严格的定义格式，如图 1-1所示，该xml文档记录的是每一种花的信息。分别是从属于哪一间花店，花的名字是什么，价格是多少等信息。值得一提的是，xml没有预定义的标签（例如本例子中的<Vendor>和<Name>）,所有的标签都是由文档的创建者发明的。

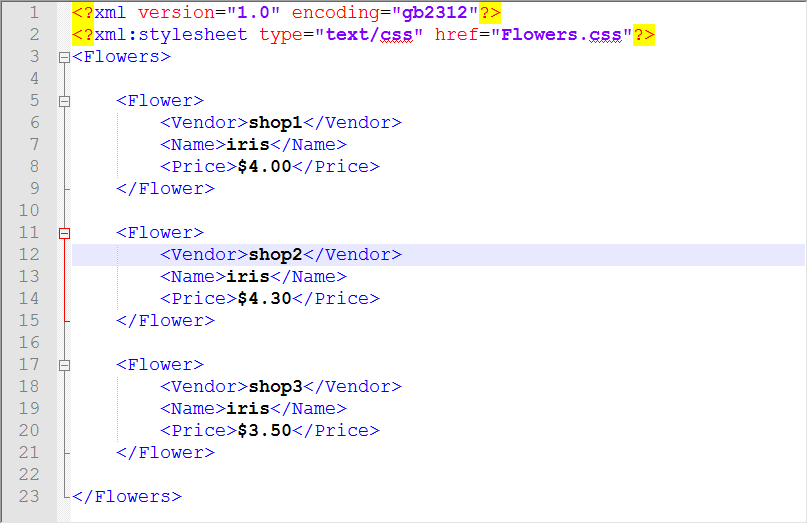


图 1-1 一个xml文档

### 关于JSON

尽管XML如此优秀，具有如此广泛的应用前景。但它也经常被诟病有标记格式繁杂、带宽占用多、需要大量代码用于解析等问题，如果有其他更加简洁好用的数据交换格式，那么XML将面临巨大挑战。

JSON全称为JavaScript Object Notation，即基于JavaScript的对象符号，也是一种轻量级数据交换格式。目前，JSON在主流编程语言中已经被广泛使用。

与XML相比，JSON除了有更简单的语法结构，还提供了在多种编程语言之间完成数据交换的能力。

JSON本质是一组字符串，相关元素会使用特定的符号标注，一般以键值对(key:value)为书写格式。

1.{}双括号表示对象

2.[]中括号表示数组

3.""*双*引号内是属性或值

4.: 冒号表示后者是前者的值(这个值可以是字符串、数字、也可以是另一个数组或对象)

例如 {"name": "Jack"} 可以理解为是一个name为Jack的对象，而[{"name": "Jack"},{"name": "Mike"}]就表示包含两个name分别为Jack和Mike的对象数组。

下面我们来看看同一组数据在xml格式和Json格式之间到底有什么不同。如图1-2所示是一个xml格式的文档。



图 1-2 一个简单的xml文档

而上述文档用JSON格式表示则如图 1-3所示。

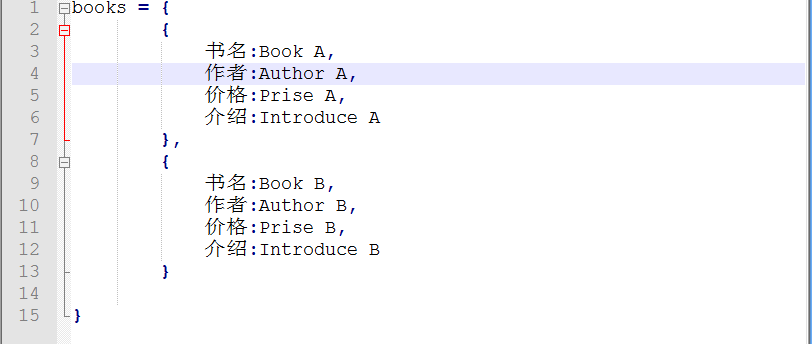


图 1-3 用JSON表示的文档

同时,JSON也可以创建Javascript对象和使用Javascript语法,实现灵活多用。如图 1-4所示。图1-4的结果显示为：姓名：Name A性别：male。

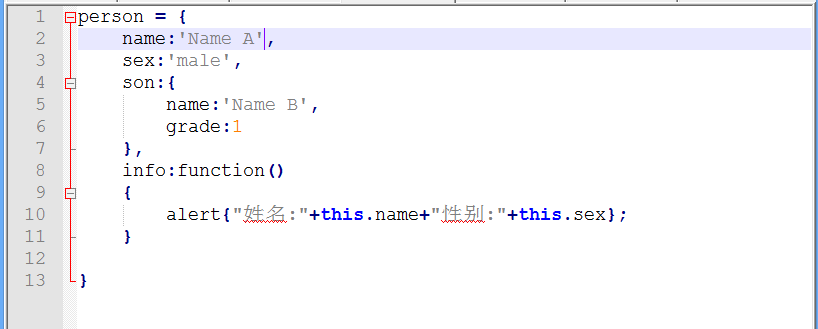


图 1-4 JSON也可以使用Javascript语法

## 案例设计与实现

### 2.1XML的两种定义格式：DTD与Schema

虽然XML允许开发者自由拓展元素，但在实际使用中，进行信息交换的双方需要一种更具体的语义约束才能保证数据交换。

DTD和XML Schema正是两种为XML定义语句约束的工具：前者简单易用，但功能相对较弱。后者比前者要复杂一些，但功能强大得多，支持丰富的数据类型,而且允许开发者自定义数据类型。图2-1找的很好，我作为用过xml的人，看的很清楚，但是作为初学者，你可能不知道2者之间的区别，需要借助搜索引擎查看它们的相关描述和区别。

图 2-1展示的是它们之间的区别。



图 2-1 区别

#### XML DTD

XML DTD(XML的文档类型定义)，具有简单的语法格式和功能。它能够定义XML文档的元素结构，元素（类型，取值，属性）限制，元素内属性（类型，数值等）限制等。图2-2为DTD的一个示例。

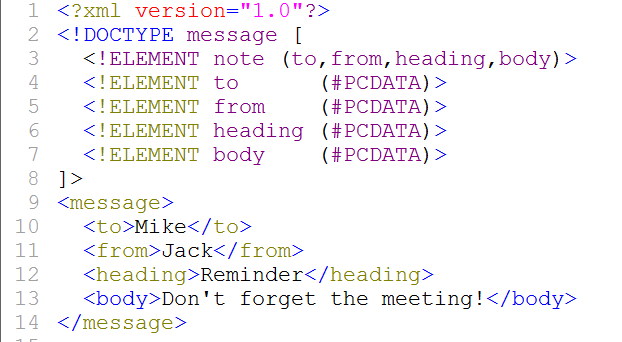


图2-2 DTD

XML DTD既可以作为一个单独文件编写，也可以编写在XML文件中，当DTD编写在XML文件中时，可参考如图2-2所示。

而当DTD文件作为一个单独的文件编写的时候有两种情况。

1.当作为单独编写的DTD文件保存在本地时，可以在需要引用的XML文件中这样写：

**<!DOCTYPE文档根结点 SYSTEM "DTD文件的URL">**

如图2-2的例子即为：

**<!DOCTYPE message SYSTEM "file///d:/ massage.dtd">**

【注意】DTD文件的URL必须是完整路径，不然无法读取，这里给出一个保险的方法：直接把DTD文件拖进浏览器打开，地址栏里面的就是DTD文件在本地的完整路径。

1. 当作为单独编写的DTD文件保存在公共文件夹里面时，可以在需要引用的XML文件中这样写：

**<!DOCTYPE文档根结点 PUBLIC "DTD名称" "DTD文件的URL">**

**例我们需要引用一个在WEB上的DTD文件：**

**<!DOCTYPE web-app PUBLIC "-//SunMicrosystems, Inc.//DTD Web Application 2.3//EN"**  **"http://java.sun.com/dtd/web-app\_2\_3.dtd">**

#### 2.XML Schema

随着XML技术领域的发展，XML DTD不能完全满足XML自动化处理的要求，例如不能很好地实现应用程序不同模块的互相协调，缺乏对文档结构、属性、数据类型等约束的足够描述等等，所以W3C于2001年5月正式推荐XML Schema为XML的标准模式。

相比XML DTD内置的10种数据类型，XML Schema支持高达37种数据类型定义，并可以很好地支持用户自定义类型数据、丰富的命名空间。但是这也令XML Schema变得比XML DTD更加复杂了。图 2-3是XML Schema的一个例子。



图 2-3 XML Schema的一个例子

下面，我们要实现三个案例。

### 2.2功能实现

#### XML DTD的应用

本次演示是用XML DTD中内、外部普通实体的引用，来约束一个XML文档

**首先，要注意普通实体的声明和引用的方法：**

**内部普通实体**就是实体内容在DTD文件本身中。内容普通实体一般包含常用文本或较难输入的文本内容。在定义实体引用时，以 & 开头，以 ; 结束。

内部普通实体语法格式：

<!ENTITY name“text”>

ENTITY:关键字 说明是个实体。

name：实体引用的名称，可以在XML文档中使用实体引用。

text：实体内容。如图2-4所示。

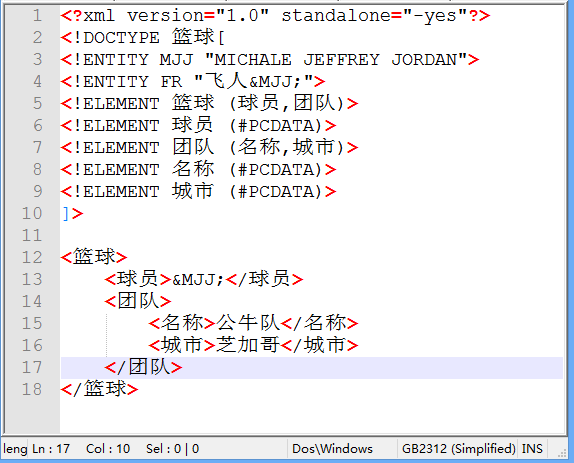


图 2-4

**外部普通实体**就是在XML文档意外包含的数据，这些数据通过URI定位的资源引入到文档中。两种类型：

1.SYSTEM

<!ENTITY name SYSTEM "URI">

2.PUBLIC

<!ENTITY name PUBLIC FPI "URI">

FPI：Formal Public Identifier，即正式公用标志符：如果外部普通实体的内容有许多< > & 就可能会使XML文档出现语法错误。

**如:**

外部DTD文件：test1.dtd。

如图 2-5所示。

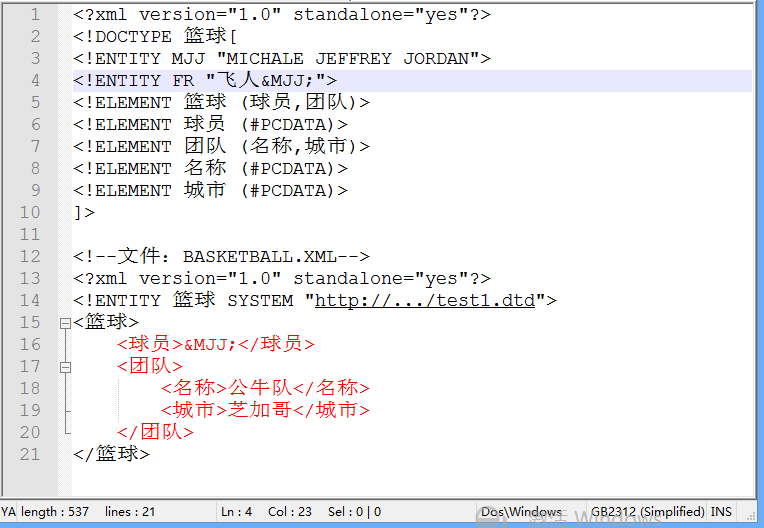


图 2-5

#### XML Schema的应用

本次演示是用XML Schema在内、外部实现对XML文档的约束

XML Schema支持丰富的数据类型、支持丰富命名空间机制、对整个xml文档或者文档局部进行效验，完全遵循xml规范，比起XML DTD 更加容易学习。

实现方式如下，如图2-6到图2-9所示。

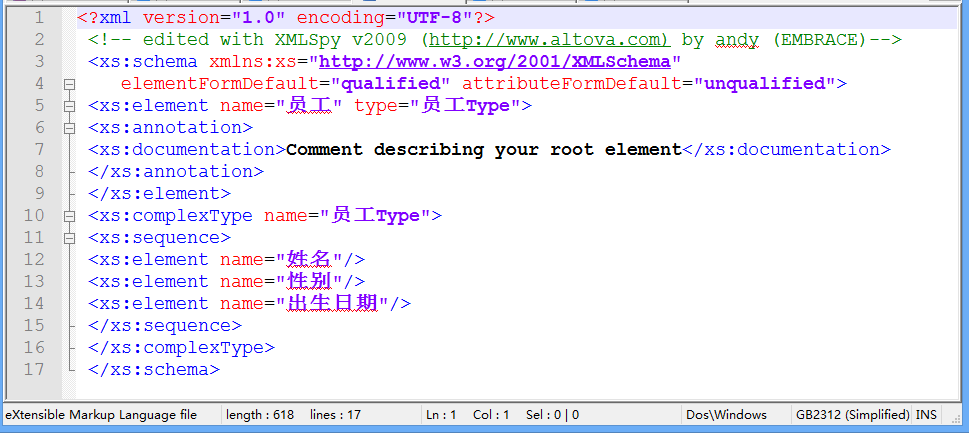


图 2-6

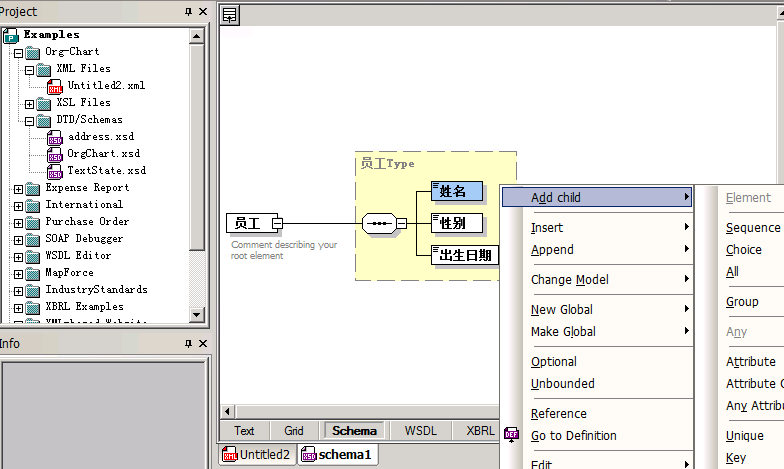


图 2-7

本次案例使用了XMLspy编辑器用于生成XML Schema，按照逻辑步骤很容易就可以添加。

然后新建一个XML文件便可以调用刚刚生成的Schema。

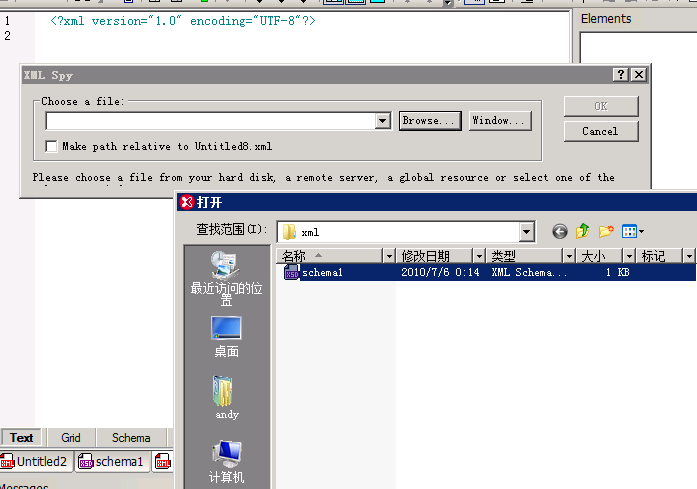


图 2-8

引用为外部约束，成功。

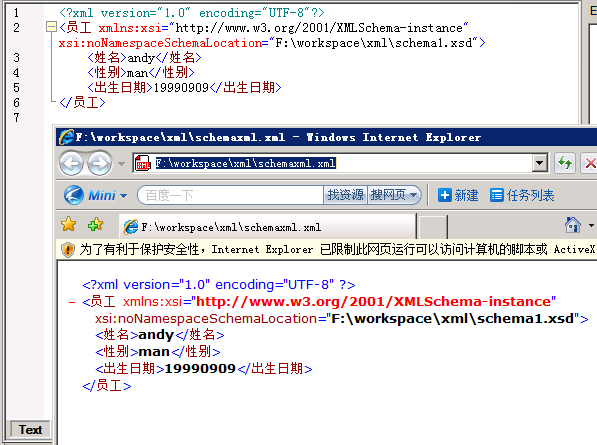


图 2-9

#### 如何利用DOM加载XML

DOM是XML文档的编程接口，它定义了访问和处理 XML 文档的标准方法，即 如何在程序中访问和操作XML文档，是与平台好语言无关的结构，DOM将XML文档作为一个树状结构进行操作，如图2-10，其步骤为：首先DOM解析器把该XML文档加载到内容中去，在内存中，XML文档的逻辑形式以树的结构存在，**而树叶被定义为节点** ，利用程序对XML文档的操作都是建立在树的结构形式上的。

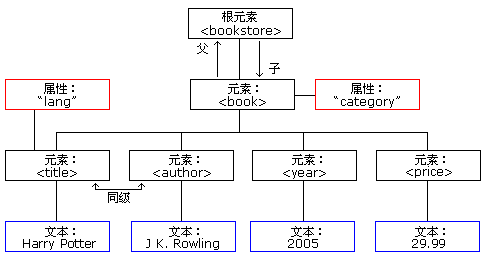
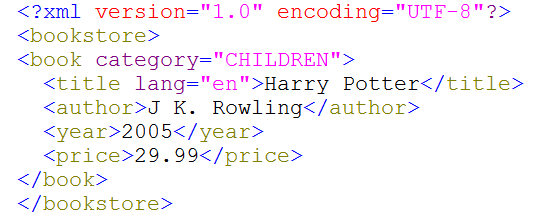


图2-10 DOM将XML文档视为一个树状结构

在上面的例子中，根节点是 <bookstore>。文档中的所有其他节点都被包含在 <bookstore> 中。根节点 <bookstore> 有一个 <book> 节点。<book> 节点有四个节点：<title>, <author>, <year> 以及 <price>，其中每个节点都包含一个文本节点，"Harry Potter", "J K. Rowling", "2005" 以及 "29.99"。

下面我们利用javascript片段尝试加载一个xml文件。

【注意】不同浏览器中内置的XML解析器是不一样的，下面我们是以微软（IE）的XML解析器。

例：

xmlDoc=new ActiveXObject("Microsoft.XMLDOM");

//创建空的微软 XML 文档对象

xmlDoc.async="false";

//关闭异步加载，这样可确保在文档完整加载之前，解析器不会继续执行脚本

xmlDoc.load("books.xml");

//告知解析器加载名为 "books.xml" 的文档

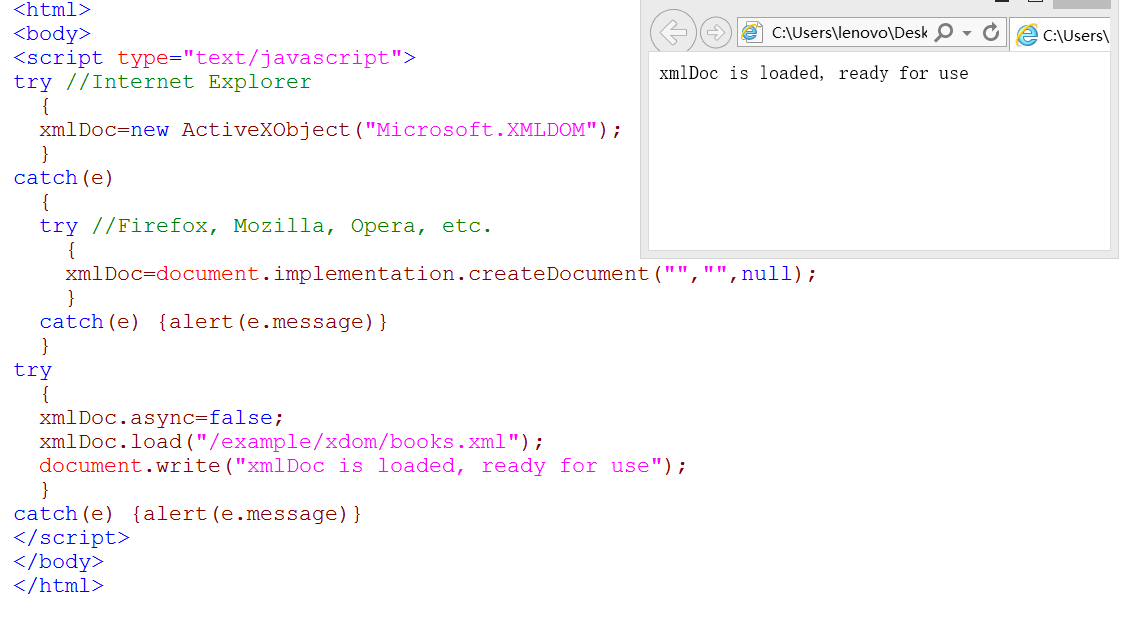


图2-11 XML已经通过DOM加载进去了

下面的示例是通过JAVA的DOM加载XML

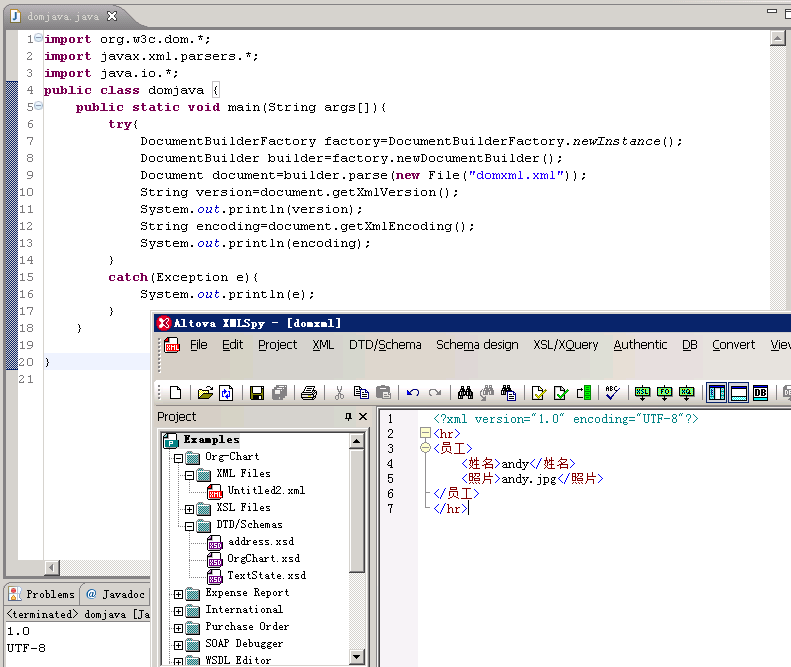


图 2-10

代码分析：

**java：**

DocumentBuilderFactory factory=DocumentBuilderFactory.newInstance ();

DocumentBuilderFactory是一个抽象类，定义工厂API，使应用程序能够从XML文档获取生成DOM对象树的解析器。这句话表示：创建一个API工厂factory，在这里可以获得API.

DocumentBuilder builder=factory.newDocumentBuilder();

DocumentBuilder定义API，使其从XML文档中获取DOM文档实例。使用此类，程序员可以从XML中获取一个Document对象。用newDocumentBuilder()方法获取。获取此类实例之后，将解析XML。这句话的意思是：

标记表示创建一个该类的对象作为实现Document对象API。

Document document=builder.parse( new  File( "domxml.xml" ));

在内存中加载xml文件，并创建对象，在加载XML文档的时候，DocumentBuilder的实例化对象会在内存中建立一个节点树，并且形成一个可以操作改节点树的对象Document。

String version=document.getXmlVersion();

当document对象创建好以后，就可以对XML文档的节点树进行相关操作。getXmlVersion()方法表示获得该XML文档使用的版本号

System. out .println(version);

String encoding=document.getXmlEncoding();

getXmlEncoding()方法表示获得该XML文档使用的编码形式。

System. out .println(encoding);

## 项目心得

至此，我们掌握了XML的最基本语法，理解了它可以用来标记数据、定义数据类型，是一种允许用户对自己的标记语言进行定义的源语言。 非常适合[万维网](http://baike.baidu.com/view/7833.htm" \t "http://baike.baidu.com/_blank)传输，提供统一的方法来描述和交换独立于[应用程序](http://baike.baidu.com/view/330120.htm)或供应商的结构化数据。

## 参考资料

1. 《疯狂XML讲义（第二版）》 李刚
2. W3C SCHOOL: <http://www.w3school.com.cn/json/json_intro.asp>
3. 百度文档 Json结构、实例：

<http://wenku.baidu.com/view/b4c6a00e581b6bd97f19ea31.html>