Android使用PULL解析XML

Android项目中XML文件资源位于res/xml目录下（需要自己创建xml文件夹）

它们在部署时将被编译为二进制形式

XML解析一般分为两种：（1）DOM解析（2）SAX解析

PULL解析从本质上属于SAX解析（基于事件驱动，但比SAX解析更简单和形象）

words.xml

<**words**>  
 <**word value="huang"**/>  
 <**word value="ji"**/>  
 <**word value="sheng"**/>  
 <**word value="ai"**/>  
 <**word value="xu"**/>  
 <**word value="dan"**/>  
 <**word value="yun"**/>  
 <**word value="ke"**/>  
 <**word value="ta"**/>  
 <**word value="yong"**/>  
 <**word value="yuan"**/>  
 <**word value="ye"**/>  
 <**word value="bu"**/>  
 <**word value="hui"**/>  
 <**word value="zhi"**/>  
 <**word value="dao"**/>  
</**words**>

activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"**>  
  
 <**Button  
 android:id="@+id/BtnParser"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:text="解析XML"  
 android:onClick="parser"**/>  
  
 <**TextView  
 android:id="@+id/tvContent"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignLeft="@+id/BtnParser"  
 android:layout\_below="@+id/BtnParser"  
 android:text=""** />

</**RelativeLayout**>

MainActivity.java

**package** com.amaker.hjs.xmlparserdemo.xml;  
  
**import** android.content.res.Resources;  
**import** android.os.Handler;  
**import** android.os.Message;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.TextView;  
  
**import** org.xmlpull.v1.XmlPullParser;  
**import** org.xmlpull.v1.XmlPullParserException;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private static final int *MSG\_FINISH***=0x0001;  
 **private** TextView **tvContent**;  
 **private** Handler **handler** = **new** Handler(){  
 @Override  
 **public void** handleMessage(Message msg) {  
 **switch** (msg.**what**){  
 **case *MSG\_FINISH***:  
 List<String> contents = (List<String>) msg.**obj**;  
 *//在主线程中显示* **for** (String content: contents){  
 **tvContent**.append(content+**"\n"**);  
 }  
 **break**;  
 }  
 };  
 };  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **this**.**tvContent**=(TextView)findViewById(R.id.***tvContent***);  
  
 }  
  
 *//按钮点击事件处理* **public void** parser(View view) {  
 */\*List<String> contents = null;  
 try {  
 contents = this.getPullParserContent(getResources() , R.xml.words);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 for (String content : contents){  
 tvContent.append(content+"\n");  
 }\*/  
  
 /\*由于文件读取会非常耗时，所以我们应该使用线程\*/* **new** Thread(){  
 @Override  
 **public void** run() {  
 **try** {  
 List<String> contents = getPullParserContent(getResources(), R.xml.***words***);  
 *//完成工作，通知主线程  
 //Message msg = new Message();  
 //或用主线程的handler的方法来创建* Message msg = **handler**.obtainMessage();  
 msg.**what**=***MSG\_FINISH***;  
 msg.**obj**=contents;  
 **handler**.sendMessage(msg);*//放入主线程Looper的消息队列* } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 };  
 }.start();  
 }  
  
 **private** List<String> getPullParserContent(Resources res, **int** id) **throws** Exception{  
 List<String> contents=**null**;  
 String tagName;  
 *//XMLPullParser* XmlPullParser parser = res.getXml(id);*//取得资源，返回一个XmlPullParser解析器  
 //Pull解析本质是SAX解析* **int** eventType = parser.getEventType();*//首先要通过解析器知道当前事件的类型* **while** (eventType != XmlPullParser.***END\_DOCUMENT***) {  
 **switch** (eventType) {  
 **case** XmlPullParser.***START\_DOCUMENT***:*//文档开始  
 //Log.i("TEST","START\_DOCUMENT:");* contents=**new** ArrayList<String>();  
 **break**;  
 **case** XmlPullParser.***END\_DOCUMENT***:*//文档结束  
 //Log.i("TEST","END\_DOCUMENT:");* **break**;  
 **case** XmlPullParser.***START\_TAG***:*//标记（标签，元素，NODE，Element ，TAG）的开始* tagName=parser.getName();*//得到标记的名字* **if** (tagName.equals(**"word"**)){  
 String value=parser.getAttributeValue(0); *//得到标记的值* contents.add(value);  
 *//Log.i("TEST","START\_TAG:"+tagName+" "+value);* }  
 **break**;  
 **case** XmlPullParser.***END\_TAG***:  
 tagName=parser.getName();  
 Log.*i*(**"TEST"**,**"END\_TAG:"**+tagName);  
 **break**;  
 }  
 eventType=parser.next();  
 }  
 **return** contents;  
 }  
}