



《移动终端开发技术》 电子教案

第三单元 解析 XML 格式数据

所属专业（教研室）： 计算机软件技术

制定人： 陈媛媛

合作人：

制定时间： 2018年2月

日照职业技术学院



单元标题	解析 XML 格式数据	单元教学学时	4 课时
		在整体设计中的位置	第 28 次
授课班级		上课地点	一体化教室
上课时间	周 月 日第 节		
教学目标	能力目标	知识目标	素质目标
	能够熟练运用 Pull 解析方式及 SAX 解析方式解析 xml 格式数据。	1、掌握解析 xml 格式数据的方法； 2、掌握 Pull 解析方式； 3、掌握 SAX 解析方式。	1、养成积极主动学习意识； 2、养成勤于动手的习惯。
教学重点、难点	教学重点：Pull 解析方式 教学难点：SAX 解析方式		
教学方法	采用反转课堂教学模式，课前学生学习微课了解知识点，课上采用教师引导、演示，学生分组练习、讨论等教学方法。 运用多媒体、AndroidStudio 开发环境、实训助手、教学平台等辅助授课。		
课前需掌握的知识	配置 Apache 服务器 （1）将 Apache24 放到 d 盘下； （2）以管理员身份打开命令窗口：C:\Windows\System32，右键 cmd.exe 输入 d: cd Apache24 cd bin httpd -k install 把 apache 安装成 windows 后台服务。 启动服务，在地址栏输入 http://localhost:80/		
教学任务分解	任务一、Pull 解析方式 任务二、SAX 解析方式		
教学总结			



一、情景导入

通常情况下，每个需要访问网络的应用程序都会有一个自己的服务器，我们可以向服务器提交数据，也可以从服务器上获取数据。网上传输数据时最常用的格式有两种：XML 和 JSON。下面我们就一个一个地学习，首先学习如何解析 XML 格式的数据。

二、课前测试

如何使用 HttpURLConnection 访问网络？

三、知识讲解

任务一、Pull 解析方式

1) 配置 Apache 服务器

(1) 将 Apache24 放到 d 盘下；

(2) 以管理员身份打开命令窗口：C:\Windows\System32，右键 cmd.exe

输入 d: cd Apache24 cd bin httpd -k install

把 apache 安装成 windows 后台服务。

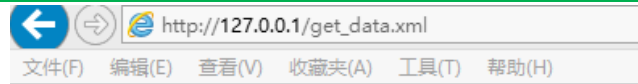
启动服务，在地址栏输入 http://localhost:80/

2) 新建 xml 文件

在 D:\Apache24\htdocs 下新建 get_data.xml

```
<apps>
  <app>
    <id>1</id>
    <name>Google Maps</name>
    <version>1.0</version>
  </app>
  <app>
    <id>2</id>
    <name>Chrome</name>
    <version>2.1</version>
  </app>
  <app>
    <id>3</id>
    <name>Google Play</name>
    <version>2.3</version>
  </app>
</apps>
```

在地址栏中输入 http://127.0.0.1/get_data.xml



```
<?xml version="1.0"?>
- <apps>
  - <app>
    <id>1</id>
    <name>Google Maps</name>
    <version>1.0</version>
  </app>
  - <app>
    <id>2</id>
    <name>Chrome</name>
    <version>2.1</version>
  </app>
  - <app>
    <id>3</id>
    <name>Google Play</name>
    <version>2.3</version>
  </app>
</apps>
```

3) 相关类及方法介绍

XmlPullParserFactory 为 XmlPullParserFactory 工厂类

XmlPullParserFactory factory = XmlPullParserFactory.newInstance();

newPullParser()方法来获取一个 XmlPullParser 类的实例

XmlPullParser xmlPullParser = factory.newPullParser();

setInput()方法将服务器返回的 xml 数据设置进去开始解析

xmlPullParser.setInput(new StringReader(xmlData));

getEventType()方法得到当前的解析事件，返回整数。

int eventType = xmlPullParser.getEventType();

getName()方法获取当前节点名。

nextText () 方法获取节点的具体内容

next()方法获取下一个解析事件

属性:

XmlPullParser.END_DOCUMENT 解析事件完成

XmlPullParser.START_TAG 开始解析节点

XmlPullParser.END_TAG 完成解析某个节点

4) 案例

在 HttpURLConnection 项目的基础上修改。

```
//创建 Request 对象, url ()方法来设置目标的网络地址
// Request request=new Request.Builder().url("http://www.baidu.com").build();
//指定访问的服务器地址是电脑本机, 对于模拟器来说, 电脑本机的 IP 地址为 10.0.2.2
Request request=new
Request.Builder().url("http://10.0.2.2/get_data.xml").build();
...
//得到返回的具体内容, response.body ()返回 responseBody 类可以方便的获取 string
String responseData=response.body().string();
//showResponse (responseData);
```



```
parseXMLWithPull(responseData);
...

//使用 Pull 解析 XML 格式数据
private void parseXMLWithPull(String xmlData) {
    try {
        //获取 XmlPullParserFactory 实例
        XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
        //获取 XmlPullParser 实例
        XmlPullParser xmlPullParser = factory.newPullParser();
        //将服务器返回的 xml 数据设置进去开始解析
        xmlPullParser.setInput(new StringReader(xmlData));
        //得到当前的解析事件
        int eventType = xmlPullParser.getEventType();
        String id = "";
        String name = "";
        String version = "";
        //解析事件没完成
        while (eventType != XmlPullParser.END_DOCUMENT) {
            //得到当前节点名
            String nodeName = xmlPullParser.getName();
            switch (eventType) {
                //开始解析节点
                //如果节点名等于 id、name 或 version，调用 nextText () 方法获取节点的具体内容
                case XmlPullParser.START_TAG: {
                    if ("id".equals(nodeName))
                        id = xmlPullParser.nextText();
                    else if ("name".equals(nodeName))
                        name = xmlPullParser.nextText();
                    else if ("version".equals(nodeName))
                        version = xmlPullParser.nextText();
                    break;
                }
                //完成解析某个节点
                case XmlPullParser.END_TAG: {
                    if ("app".equals(nodeName)) {
                        Log.d("MainActivity", "id is " + id);
                        Log.d("MainActivity", "name is " + name);
                        Log.d("MainActivity", "version is " + version);
                    }
                    break;
                }
                default:
```

```
        break;
    }
    eventType = xmlPullParser.next(); //获取下一个解析事件
}
} catch (Exception e) {
    e.printStackTrace();
}
}
```

运行结果:

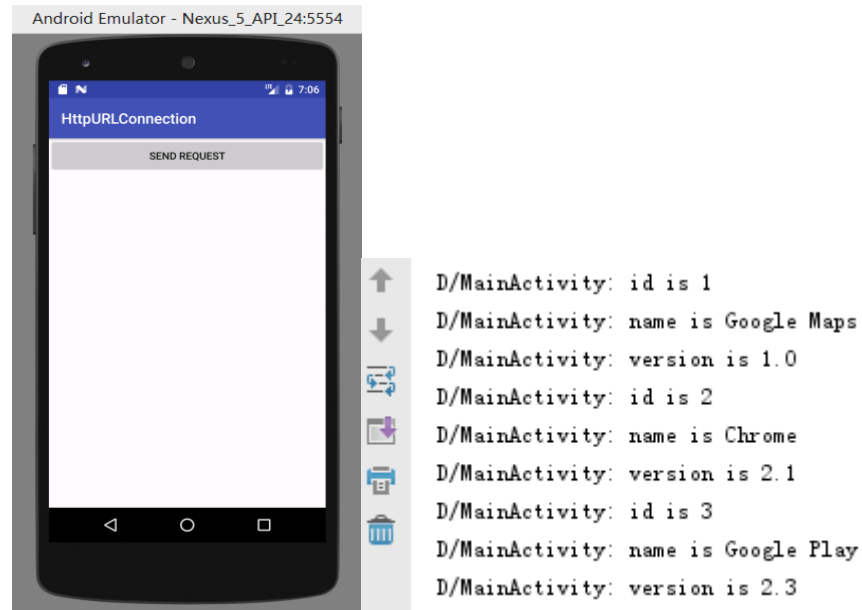


图1 使用 Pull 解析 xml 格式数据

任务二、SAX 解析方式

Simple API for XML，一种基于事件流驱动、通过接口方法解析的 XML 解析方法。在使用 SAX 解析 XML 文档时，关键在于自定义自己的 Handler 处理类 & 复写对应方法。

```
public class MyHandler extends DefaultHandler
{
    @Override
    public void startDocument() throws SAXException{
    }
    @Override public void startElement(String uri,String localName,String qName,
    Attributes attributes) throws SAXException{
    }
    @Override public void characters(char[] ch,int start,int length) throws
    SAXException{
    }
}
```

```
@Override public void endElement(String uri, String localName, String qName) throws SAXException{  
}  
  
@Override public void endDocument() throws SAXException{  
}  
}
```

运行结果:

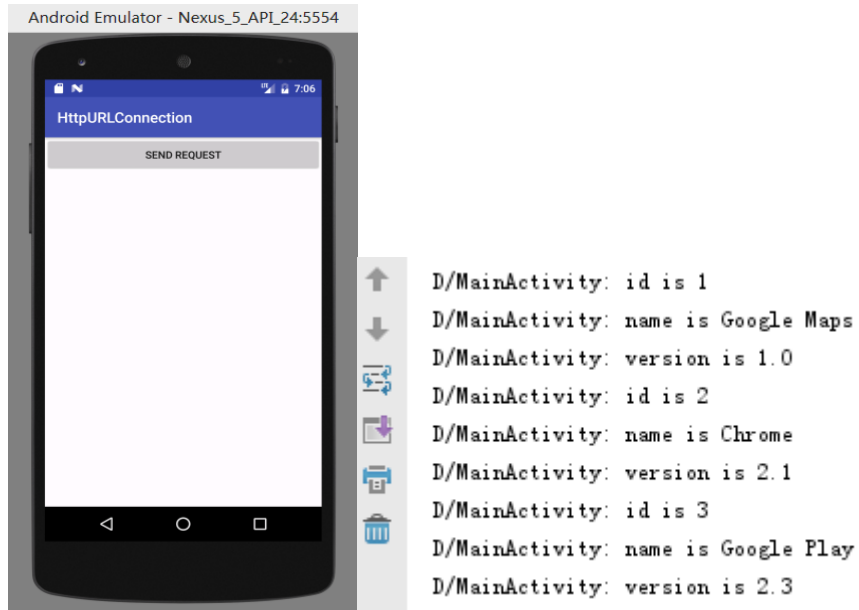


图2 使用 SAX 解析 xml 格式数据

四、演示作品

各小组演示汇报作品，教师与其他小组总结评价作品，各小组课后完善作品并提交到教学平台。

五、知识巩固

总结知识点，使用教学平台中的随堂练习题巩固本所学知识。

拓展
作业

- 1、配置 Apache 服务器
- 2、新建 xml 文件
- 3、运用 Pull 解析方式及 SAX 解析方式解析 xml 格式数据

教学
后记