

使用Android通过访问网络获取XML网页相应数据

1、在解析XML文件时，如果要解析出网页的源代码，直接解析出来用一个字符串保存就，将这个保存好的字符串给消息处理

```
@Override
public void run() {
    String path = etxt.getText().toString(); // 将文本框中的地址获取出来
    try {
        URL u = new URL(path); // 去访问拿到地址
        HttpURLConnection connection = (HttpURLConnection) u
            .openConnection();
        // 设置访问方式及响应等待时间          通过 HttpURLConnection
        connection.setRequestMethod("GET");
        connection.setReadTimeout(10000);
        int code = connection.getResponseCode(); // 获取访问地址的响应状态
        if (code == 200) {
            // 访问成功获取输入流，读取网页的内容获取源代码
            InputStream input = connection.getInputStream();
            String string = StrFac.getString(input);
            // txt.setText(string);

            mag.obj = string;
            her.sendMessage(mag);
            input.close();
        }
    }
}
```

如果要获取网页源码中的些数据可以在解析的时间将数据先保存到一个集合或中去；

```

public void run() {
    String path = "http://192.168.2.102:8080/MyServlets/news.xml";
    try {
        URL U = new URL(path);
        HttpURLConnection con = (HttpURLConnection) U
            .openConnection();
        con.setReadTimeout(5000);
        con.setRequestMethod("GET");
        int code = con.getResponseCode();
        if (code == 200) {
            InputStream in = con.getInputStream();
            XmlPullParser parser = Xml.newPullParser();
            parser.setInput(in, "utf-8");
            int eventType = parser.getEventType();
            while (eventType != parser.END_DOCUMENT) {
                if (eventType == parser.START_TAG) {
                    if ("news".equals(parser.getName())) {
                    }
                    if ("new".equals(parser.getName())) {
                        ns = new NewsID();
                        as.add(ns);
                    }
                    if ("biaoti".equals(parser.getName())) {
                        ns.setBiaoti(parser.nextText().toString()
                            .trim());
                    }
                }
                if ("biaoti".equals(parser.getName())) {
                    ns.setBiaoti(parser.nextText().toString()
                        .trim());
                }
                if ("other".equals(parser.getName())) {
                    ns.setOther(parser.nextText().toString()
                        .trim());
                }
                if ("times".equals(parser.getName())) {
                    ns.setTimes(parser.nextText().toString()
                        .trim());
                }
                if ("laiyuan".equals(parser.getName())) {
                    ns.setLaiyuan(parser.nextText().toString()
                        .trim());
                }
                if ("zaiyao".equals(parser.getName())) {
                    ns.setZaiyao(parser.nextText().toString()
                        .trim());
                }
            }
        }
    }
}

```

如果集合中存的是自定义对象，那么new对象要在XML文件的子节点下

向自定义对象添加解析到的数据

```

    }
    if ("zaiyao".equals(parser.getName())) {
        ns.setZaiyao(parser.nextText().toString()
            .trim());
    }
}

Message m = new Message();
m.what = 1;
han.sendMessage(m);

eventType = parser.next();
}
}

```

```

public class MyAdapter extends BaseAdapter {
    @Override
    public int getCount() {
        return as.size();
    }
    @Override
    public Object getItem(int position) {
        return null;
    }
    @Override
    public long getItemId(int position) {
        return 0;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        View v;
        if (convertView == null) {
            v = View.inflate(getApplicationContext(), R.layout.item, null);
        } else {
            v = convertView;
            item的条目 position
        }
        NewsID newsID = as.get(position);
        TextView biaoti = (TextView) v.findViewById(R.id.biaoti);
        biaoti.setText(newsID.getBiaoti());
        TextView other = (TextView) v.findViewById(R.id.other);
        other.setText(newsID.getOther());
    }
}

```

新闻客户端的开发流程：

- 1、在主布局中创建ListView组件
- 2、创建Item布局，添加相应的UI组件
- 3、获取ListView组件的id,连接网络解析XML文件，获取相应的数据存入集合中。
- 4、最后通过自定义对象获取相应的数据给相应的UI组件


```
<ListView
```

```
    android:id="@+id/lv" 主布局的ListView  
    android:layout_width=组件 "match_parent"  
    android:layout_height="match_parent" >
```

```
</ListView>
```

```
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical" >
```

```
    Item布局的UI组件  
    等同于 ListView  
    <TextView
```

```
        android:id="@+id/biaoti"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_marginBottom="3dp"  
        android:layout_marginLeft="5dp"  
        android:layout_marginRight="5dp"  
        android:text="111111"  
        android:textSize="30sp" />
```

```
    <LinearLayout
```

```
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"
```

```

public class NewsID {
    private String biaoti;
    private String other;
    private String times;
    private String laivuan;
    private String zaiyao;

    public String getBiaoti() {
        return biaoti;
    }
    public void setBiaoti(String biaoti) {
        this.biaoti = biaoti;
    }
    public String getOther() {
        return other;
    }
    public void setOther(String other) {
        this.other = other;
    }
    public String getTimes() {
        return times;
    }
}

```

自定义类的数据类型
与XML的要一值

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    readNews();
    wv = (ListView) findViewById(R.id.lv);
    as = new ArrayList<NewsID>();
    han = new Handler() {
        public void handleMessage(Message msg) {
            if (msg.what == 1) {
                // wv.setAdapter(new MyList());
                wv.setAdapter(new MyAdapter());
            }
        }
    };
}

```



```

public void readNews() {
    new Thread() {
        @Override
        public void run() {
            String path = "http://192.168.2.102:8080/MyServlets/news.xml";
            try {
                URL U = new URL(path);
                HttpURLConnection con = (HttpURLConnection) U
                    .openConnection();
                con.setReadTimeout(5000);
                con.setRequestMethod("GET");
                int code = con.getResponseCode();
                if (code == 200) {
                    InputStream in = con.getInputStream();
                    XmlPullParser parser = Xml.newPullParser();
                    parser.setInput(in, "utf-8");
                    int eventType = parser.getEventType();
                    while (eventType != parser.END_DOCUMENT) {
                        if (eventType == parser.START_TAG) {
                            if ("news".equals(parser.getName())) {
                                // ...
                            }
                        }
                    }
                }
            } catch (Exception e) {
                // ...
            }
        }
    }.start();
}

```

```

public long getItemId(int position) {
    return 0;
}
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View v;
    if (convertView == null) {
        v = View.inflate(getApplicationContext(), R.layout.item, null);
    } else {
        v = convertView;
    }
    // 通过自定义对象获取解析时添加的数据再给
    // Item指定的UI

```

```

NewsID newsID = as.get(position);
TextView biaoti = (TextView) v.findViewById(R.id.biaoti);
biaoti.setText(newsID.getBiaoti());
TextView other = (TextView) v.findViewById(R.id.other);
other.setText(newsID.getOther());
TextView times = (TextView) v.findViewById(R.id.times);
times.setText(newsID.getTimes());
TextView laiyan = (TextView) v.findViewById(R.id.laiyan);
laiyan.setText(newsID.getLaiyan());
TextView zaiyao = (TextView) v.findViewById(R.id.zaiyao);
zaiyao.setText(newsID.getZaiyao());
return v;

```