Evan

Only let oneself become strong enough, good enough, can afford the life that you want to.

使用Jsoup 抓取页面的数据

标签： jsoup html android java

2015-12-22 09:46 299人阅读 评论(0) 收藏 举报

分类： 移动开发（38）

需要使用的是jsoup-1.7.3.jar包 如果需要看文档我下载请借一步到官网：http://jsoup.org/

这里贴一下我用到的 Java工程的测试代码

```java
<span style="font-size:14px;">package com.javen.Jsoup;

import java.io.IOException;

import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

public class JsoupTest {
    static String url="http://www.cnblogs.com/zyw-205520/archive/2012/12/20/2826402.html";
    /**
     * @param args
     * @throws Exception
     */
    public static void main(String[] args) throws Exception {

        // TODO Auto-generated method stub
        BolgBody();
        //test();
        //Blog();
        /*
         * Document doc = Jsoup.connect("http://www.oschina.net/")
         * .data("query", "Java") // 请求参数 .userAgent("I ’ m jsoup") // 设置
         * User-Agent .cookie("auth", "token") // 设置 cookie .timeout(3000) //
         * 设置连接超时时间 .post();
         */// 使用 POST 方法访问 URL

        /*
         * // 从文件中加载 HTML 文档 File input = new File("D:/test.html"); Document doc
         * = Jsoup.parse(input,"UTF-8","http://www.oschina.net/");
         */
    }

    /**
     * 获取指定HTML 文档指定的body
     * @throws IOException
     */
    private static void BolgBody() throws IOException {
        // 直接从字符串中输入 HTML 文档
        String html = "<html><head><title> 开源中国社区 </title></head>"
                + "<body><p> 这里是 jsoup 项目的相关文章 </p></body></html>";
        Document doc = Jsoup.parse(html);
        System.out.println(doc.body());


```

```
        // 从 URL 直接加载 HTML 文档
        Document doc2 = Jsoup.connect(url).get();
        String title = doc2.body().toString();
        System.out.println(title);
    }

    /**
     * 获取博客上的文章标题和链接
     */
    public static void article() {
        Document doc;
        try {
            doc = Jsoup.connect("http://www.cnblogs.com/zyw-205520/").get();
            Elements ListDiv = doc.getElementsByAttributeValue("class","postTitle");
            for (Element element :ListDiv) {
                Elements links = element.getElementsByTag("a");
                for (Element link : links) {
                    String linkHref = link.attr("href");
                    String linkText = link.text().trim();
                    System.out.println(linkHref);
                    System.out.println(linkText);
                }
            }
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }
    /**
     * 获取指定博客文章的内容
     */
    public static void Blog() {
        Document doc;
        try {
            doc = Jsoup.connect("http://www.cnblogs.com/zyw-205520/archive/2012/12/20/2826402.html").get();
            Elements ListDiv = doc.getElementsByAttributeValue("class","postBody");
            for (Element element :ListDiv) {
                System.out.println(element.html());
            }
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

}</span>
```

下面来介绍android中使用Jsoup异步解析网页的数据 **请注意： 这里很容易遇到一个乱码的问题**

1. 配置文件：AndroidManifest.xml中加 权限 <uses-permission android:name="android.permission.INTERNET"></uses-permission>
2. layout的布局文件

[html]

```
<span style="font-size:14px;"><LinearLayout 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"
    android:orientation="vertical" >

    <WebView
        android:id="@+id/webView"
        android:layout_width="fill_parent"
        android:layout_height="200dp" />

    <ScrollView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textView"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/hello_world" />
    </ScrollView>

</LinearLayout></span>
```

3.主要异步加载数据的代码

```
[java]
<span style="font-size:14px;">package com.javen.aaa;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.URL;

import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

import android.app.Activity;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.webkit.WebView;
import android.widget.TextView;

public class MainActivity extends Activity {
    private WebView webView;
    private TextView textView;
    private static final int DIALOG_KEY = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        webView = (WebView) findViewById(R.id.webView);
        textView=(TextView) findViewById(R.id.textView);
        try {
            ProgressAsyncTask asyncTask=new ProgressAsyncTask(webView,textView);
            asyncTask.execute(10000);
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public  String test() {
        StringBuffer buffer=new StringBuffer();
        Document doc;
        try {
            doc = Jsoup.connect("http://www.cnblogs.com/zyw-205520/").get();
            Elements ListDiv = doc.getElementsByAttributeValue("class","postTitle");
            for (Element element :ListDiv) {
                Elements links = element.getElementsByTag("a");
                for (Element link : links) {
                    String linkHref = link.attr("href");
                    String linkText = link.text().trim();
                    buffer.append("linkHref=="+linkHref);
                    buffer.append("linkText=="+linkText);

                    System.out.println(linkHref);
                    System.out.println(linkText);
                }
            }
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return buffer.toString();

    }
```

```
66.
67.         // 弹出"查看"对话框
68.         @Override
69.         protected Dialog onCreateDialog(int id) {
70.             switch (id) {
71.             case DIALOG_KEY: {
72.                 ProgressDialog dialog = new ProgressDialog(this);
73.                 dialog.setMessage("获取数据中  请稍候...");
74.                 dialog.setIndeterminate(true);
75.                 dialog.setCancelable(true);
76.                 return dialog;
77.             }
78.             }
79.             return null;
80.         }
81.
82.         public static String readHtml(String myurl) {
83.             StringBuffer sb = new StringBuffer("");
84.             URL url;
85.             try {
86.                 url = new URL(myurl);
87.                 BufferedReader br = new BufferedReader(new InputStreamReader(url.openStream(), "gbk"));
88.                 String s = "";
89.                 while ((s = br.readLine()) != null) {
90.                     sb.append(s + "\r\n");
91.                 }
92.             } catch (Exception e) {
93.                 e.printStackTrace();
94.             }
95.             return sb.toString();
96.         }
97.
98.     class ProgressAsyncTask extends AsyncTask<Integer, Integer, String> {
99.
100.         private WebView webView;
101.         private TextView textView;
102.         public ProgressAsyncTask(WebView webView,TextView textView) {
103.             super();
104.             this.webView=webView;
105.             this.textView=textView;
106.         }
107.
108.         /**
109.          * 这里的Integer参数对应AsyncTask中的第一个参数 这里的String返回值对应AsyncTask的第三个参数
110.          * 该方法并不运行在UI线程当中，主要用于异步操作，所有在该方法中不能对UI当中的空间进行设置和修改
111.          * 但是可以调用publish Progress方法触发onProgressUpdate对UI进行操作
112.          */
113.         @Override
114.         protected String doInBackground(Integer... params) {
115.             String str =null;
116.             Document doc = null;
117.             try {
118. //                String url ="http://www.cnblogs.com/zyw-205520/p/3355681.html";
119. //
120. //                doc= Jsoup.parse(new URL(url).openStream(),"utf-8", url);
121. //                //doc = Jsoup.parse(readHtml(url));
122. //                //doc=Jsoup.connect(url).get();
123. //                str=doc.body().toString();
124.                 doc = Jsoup.connect("http://www.cnblogs.com/zyw-205520/archive/2012/12/20/2826402.html").get();
125.                 Elements ListDiv = doc.getElementsByAttributeValue("class","postBody");
126.                 for (Element element :ListDiv) {
127.                     str=element.html();
128.                     System.out.println(element.html());
129.                 }
130.                 Log.d("doInBackground", str.toString());
131.                 System.out.println(str);
132.                 //你可以试试GBK或UTF-8
133.             } catch (Exception e) {
134.                 // TODO Auto-generated catch block
135.                 e.printStackTrace();
136.             }
137.             return str.toString() ;
138.             //return test();
139.         }
140.
141.         /**
142.          * 这里的String参数对应AsyncTask中的第三个参数（也就是接收doInBackground的返回值）
143.          * 在doInBackground方法执行结束之后在运行，并且运行在UI线程当中 可以对UI空间进行设置
144.          */
```

```
        @Override
        protected void onPostExecute(String result) {
            webView.loadData(result, "text/html;charset=utf-8", null);
            textView.setText(result);
            removeDialog(DIALOG_KEY);
        }

        // 该方法运行在UI线程当中,并且运行在UI线程当中 可以对UI空间进行设置
        @Override
        protected void onPreExecute() {
            showDialog(DIALOG_KEY);
        }

        /**
         * 这里的Intege参数对应AsyncTask中的第二个参数
         * 在doInBackground方法当中，，每次调用publishProgress方法都会触发onProgressUpdate执行
         * onProgressUpdate是在UI线程中执行，所有可以对UI空间进行操作
         */
        @Override
        protected void onProgressUpdate(Integer... values) {

        }
    }

}</span>
```

原文：http://www.cnblogs.com/zyw-205520/p/3421687.html

顶 1　　踩 0

- 上一篇 Fragment+ViewPager实现仿微信点击和滑动切换界面
- 下一篇 android AsyncTask介绍 AsyncTask和Handler对比

相关文章推荐

- 使用Jsoup 抓取页面的数据
- jsoup抓取其他网站的页面代码
- 使用java的html解析器jsoup和jQuery实现一个自...
- android 使用Jsoup 抓取页面的数据
- HttpURLConnection 和HttpClient+Jsoup处理...
- Android 利用jsoup 抓取腾讯应用市场的软件APP...
- jSoup Cookbook-提取数据 7 使用选择器语法查...
- 在android中使用jsoup解析页面链接
- jsoup 默认抓取页面大小为1M
- 使用Jsoup抓取页面的数据

猜你在找

机器学习之概率与统计推断
机器学习之凸优化
响应式布局全新探索
深度学习基础与TensorFlow实践
前端开发在线峰会
机器学习之数学基础
机器学习之矩阵
探究Linux的总线、设备、驱动模型
深度学习之神经网络原理与实战技巧
TensorFlow实战进阶：手把手教你做图像识别应用

查看评论