
网络数据抓取

陈 浩

目录

| | |
|---------------------------------------|----|
| Android 中的 HTTP 编程..... | 2 |
| 1 HttpClient | 2 |
| 2 Post 和 Get 在 HttpClient 中的使用..... | 2 |
| 2.1 GET 方法..... | 2 |
| 2.2 POST 方法..... | 3 |
| 2.3 注意 | 5 |
| HTML 网页数据抓取 | 5 |
| 1 正则表达式 （简单、规律性极强） | 5 |
| 1.1 实战案例..... | 6 |
| 1.2 学习途径..... | 7 |
| 2 JSOUP （复杂多变） | 7 |
| 2.1 Jsoup 是数据来源： | 7 |
| 2.2 Jsoup 查询..... | 7 |
| 2.2.1 DOM 方法查找元素 | 7 |
| 2.2.2 选择器的方法查找元素..... | 8 |
| 2.4 实战案例 | 9 |
| 2.5 学习途径..... | 10 |
| SQLite Databases 数据存储..... | 10 |
| 1 SQLite Databases 介绍 | 10 |
| 2 SQLite 数据类型..... | 11 |
| 3 使用 SQLiteOpenHelper 对数据库进行版本管理..... | 12 |
| 4 SQLite 基本操作..... | 13 |
| 5 综合图 | 13 |

Android 中的 HTTP 编程

1 HttpClient

HTTP 协议是现在 Internet 上使用得最多，也是最重要的协议之一，越来越多的 Android 应用程序需要直接通过 HTTP 协议来访问网络资源。虽然在 Android 的 `java.net` 包中已经提供了访问 HTTP 协议的基本功能，但对于 Android 的应用程序来说还是不够丰富和灵活。HttpClient 是 Apache Jakarta Common 下的子项目，用来提供高效的、最新的、功能丰富的支持 HTTP 协议的客户端编程工具包。

一般情况下我们使用浏览器来访问一个 Web 服务器，用来浏览页面查看信息或者提交一些数据等。所访问的这些页面有的仅仅是一些普通页面，有点需要用户登录后方可使用，有的需要认证以及通过加密方式传输。目前的浏览器（谷歌、火狐、百度，等等）处理这些情况都不会构成问题。但对于 Android 应用程序访问普通网页还好，但访问其他的一些网页就比较困难了。HttpClient 就是专门设计用来简化 HTTP 客户端与服务端间各种通信编程的。通过它可以让原来复杂的事情轻松解决。

在 java 中也是可以使用，但至少导入 `httpclient-4.4.1.jar`、`httpcore-4.4.1.jar`、`commons-logging-1.2.jar` 这三个 jar 包。

2 Post 和 Get 在 HttpClient 中的使用

HttpClient 提供的主要的功能如下：

- 实现了所有的 HTTP 的方法
- 支持自动转向
- 支持 HTTPS 协议
- 支持代理服务器

HTTP 请求方法中最常用的是 GET 方法和 POST 方法。

2.1 GET 方法

GET 方法要求服务器将 URL 定位的资源放在响应报文的数据部分，回送给客户端。使用 GET 方法时，请求参数和对应的值附加在 URL 后面，利用一个问号（“？”）代表 URL 的结尾和请求参数的开始。

```
public String doGet(String url) {
```

```

String pageInfo = "";
// 取得默认的 HttpClient 实例
HttpClient httpClient = new DefaultHttpClient();
// 取得默认的 HttpGet 实例
HttpGet httpGet = new HttpGet(url);
try {
    //连接到服务器
    HttpResponse response = httpClient.execute(httpGet);
    // 验证 Http 状态码 200 OK
    if (response.getStatusLine().getStatusCode() == 200) {
        // 获取数据内容
        InputStream inputStream = response.getEntity().getContent();
        // 数据内容InputStream 格式 转换为 字符串格式
        pageInfo = inputStreamToString(inputStream);
    }
} catch (ClientProtocolException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

return pageInfo;
}

```

具体代码 见项目 YangtzeHttpSamples

2.2 POST 方法

POST 方法要求被请求的服务器接收附在请求后面的数据，常用于提交表单。POST 方法将请求参数封装在 HTTP 请求数据中，以键值对的形式出现，可以传输大量的数据。

```

public String doPost() {
    String pageInfo = "";
    // 取得默认的 HttpClient 实例
    HttpClient httpClient = new DefaultHttpClient();
    String url =
"http://i.meishi.cc/login.php?redirect=http%3A%2F%2Fwww.meishij.net%2F";
    HttpPost httpPost = new HttpPost(url);
    try {
        List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();
        nameValuePairs.add(new BasicNameValuePair("redirect",

```

```

"http://www.meishij.net/"));
    nameValuePairs.add(new BasicNameValuePair("username", "ch93211"));
    nameValuePairs.add(new BasicNameValuePair("password", "201101201"));
    //使用utf-8 格式对数据进行编码
    httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs, "utf-8"));
    //连接到服务器
    HttpResponse response = httpClient.execute(httpPost);
    // 验证 Http 状态码 200 OK
    if (response.getStatusLine().getStatusCode() == 200) {
        // 获取数据内容
        InputStream inputStream = response.getEntity().getContent();
        // 数据内容InputStream 格式 转换为 字符串格式
        pageInfo = inputStreamToString(inputStream);
    }
} catch (ClientProtocolException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
//返回网页数据
return pageInfo;
}

```

具体代码 见项目 YangtzeHttpSamples

上面的 GET 和 POST 方法是以 InputStream 的形式返回页面的信息，很多情况下需要以 String 字符串的格式。

```

private String inputStreamToString(InputStream inputStream) {
    String s = "";
    String line = "";
    // 定义 BufferedReader , 载入 InputStreamReader
    BufferedReader rd = new BufferedReader(new InputStreamReader(inputStream));
    try {
        while ((line = rd.readLine()) != null) {
            s += line;
        }
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return s;
}

```

2.3 注意

1. Android 开发应用程序时，在 android 4.0 以上版本不能在主线程（UI 线程）中访问网络，否则运行时报 `android.os.NetworkOnMainThreadException` 异常。
2. 网页转码即 web 网页转换为 wap 网页，可以通过百度转码实现，提供的接口有 <http://gate.baidu.com/tc?from=opentc&src=>，只需在 "src=" 后面加入 web 网址就可以了，例如：
<http://gate.baidu.com/tc?from=opentc&src=http://news.yangtzeu.edu.cn/index.html>
3. Android 开发应用程序时，如果应用程序需要访问网络权限，需要在 `AndroidManifest.xml` 中加入以下代码：
`<uses-permission android:name="android.permission.INTERNET"></uses-permission>`
4. HTTP 状态码

| 状态代码 | 状态信息 | 含义 |
|------|-----------------------|-------------------------------|
| 200 | OK | 客户端请求成功 |
| 400 | Bad Request | 请求出现语法错误。 |
| 403 | Forbidden | 服务器收到请求，但拒绝提供服务 |
| 404 | Not Found | 无法找到指定位置的资源。这也是一个常用的应答。 |
| 500 | Internal Server Error | 服务器遇到了不可预期的错误 |
| 503 | Service Unavailable | 服务器由于维护或者负载过重未能应答。一段时间后可能恢复正常 |

HTML 网页数据抓取

1 正则表达式（简单、规律性极强）

正则表达式使用单个字符串来描述、匹配一系列符合某个句法规则的字符串。在很多文本编辑器里，正则表达式通常被用来检索、替换那些符合某个模式的文本。

正则表达式在网页数据抓取中主要有以下两方面的作用：

- 对 URL 连接进行过滤，只提取特定格式的链接
- 提取网页内容

```
<a href="http://www.baidu.com">Baidu</a>
```

```

<a href='http://www.baidu.com'>Baidu</a>
<a href=http://www.baidu.com>Baidu</ a>
<a class="l23" title="abc" href="http://www.baidu.com">Baidu</a>
<a href="http://www.baidu.com">Baidu</a>
<a href=http://www.baidu.com >Baidu</a>
<a href=http://www.360.com >360</a>

```

通过下面的正则表达式：

`<[aA]\s+.*?[hH][rR][eE][fF]=\s*("[\']*|)(.*?)\1(\s[>]*)*?>(.*?)<[/aA]>` 找出网页中的链接。

优点：强大的符号系统，编码简单，不需引入第三方 Jar 包

缺点：它最大的优点同时也是它最大的缺点，表达式难以书写、可读性差

1.1 实战案例

```

/**
 * 获取符合规则的字符串
 * @param strHtml 源字符串
 */
private static void getRegText(String strHtml) {
    //正则表达式
    String reg =
"<[aA]\\s+.*?[hH][rR][eE][fF]=\\s*(\"|\\'|)?(.*?)(\\1)(\\s[>]*)*?>(.*?)"
">\\[/aA]>";

    Pattern pattern = Pattern.compile(reg);
    Matcher matcher = pattern.matcher(strHtml);

    while (matcher.find()) {
        //找到匹配的字符串
        String findText = matcher.group();
        System.out.println(findText);
    }
}

```

具体代码 见项目 YangtzehtmlGetSamples

执行结果：

```
<a href="http://www.baidu.com">Baidu</a>
<a href='http://www.baidu.com'>Baidu</a>
<a href=http://www.baidu.com>Baidu</ a>
    <a class="l23" title="abc" href="http://www.baidu.com">Baidu</a>
<a href="http://www.baidu.com">Baidu</a>
<a href=http://www.baidu.com >Baidu</a>
<a href=http://www.360.com >360</a>
```

1.2 学习途径

简单学习: <http://deerchao.net/tutorials/regex/regex.htm>

深入学习: 马士兵正则表达式

测试工具:

网页 (<http://tool.oschina.net/regex>)

RegexTest (http://www.jb51.net/tools/regex_tester/index.htm)

2 JSOUP (复杂多变)

jsoup 是一款 Java 的 HTML 解析器,可直接解析某个 URL 地址、HTML 文本内容。它提供了一套非常省力的 API,可通过 DOM, CSS 以及类似于 jQuery 的操作方法来取出和操作数据。

[解析 HTML 文档的项目](#)很多,如 htmlparser、NekoHTML、JTidy、HtmlCleaner 等其中比较出名的有 htmlparser,但现在用的很少,因为 htmlparser 很少更新。

2.1 Jsoup 是数据来源:

1. 一个文件 File
2. 一个网站的 HTML 字符串 String
3. 一个网址 URL

2.2 Jsoup 查询

2.2.1 DOM 方法查找元素

Elements 这个对象提供了一系列类似于 DOM 的方法来查找元素,抽取并处理其中的数据。

查找元素

- [getElementById\(String id\)](#)
- [getElementsByTag\(String tag\)](#)
- [getElementsByClass\(String className\)](#)
- [getElementsByAttribute\(String key\)](#) (and related methods)

元素数据

- [attr\(String key\)](#) 获取属性 [attr\(String key, String value\)](#) 设置属性
- [text\(\)](#) 获取文本内容 [text\(String value\)](#) 设置文本内容
- [attributes\(\)](#) 获取所有属性
- [id\(\)](#), [className\(\)](#) and [classNames\(\)](#)
- [html\(\)](#) 获取元素内 HTML [html\(String value\)](#) 设置元素内的 HTML 内容
- [data\(\)](#) 获取数据内容（例如：script 和 style 标签）
- [tag\(\)](#) and [tagName\(\)](#)

2.2.2 选择器的方法查找元素

jsoup elements 对象支持类似于 CSS (或 jquery) 的选择器语法，来实现非常强大和灵活的查找功能。这个 `select` 方法在 Document, Element, 或 Elements 对象中都可以使用。且是上下文相关的，因此可实现指定元素的过滤，或者链式选择访问。

Selector 选择器

- [tagname](#): 通过标签查找元素，比如：a
- [#id](#): 通过 ID 查找元素，比如：#logo
- [.class](#): 通过 class 名称查找元素，比如：.masthead
- [\[attribute\]](#): 利用属性查找元素，比如：[href]
- [\[attr=value\]](#): 利用属性值来查找元素，比如：[width=500]
- [ns|tag](#): 通过标签在命名空间查找元素，比如：可以用 fb|name 语法来查找 <fb:name> 元素
- [\[attr^=value\]](#), [\[attr\\$=value\]](#), [\[attr*=value\]](#): 利用匹配属性值开头、结尾或包含属性值来查找元素，比如：[href*=path/]
- [\[attr~regex\]](#): 利用属性值匹配正则表达式来查找元素，比如：
img[src~=(?i)\.(png|jpe?g)]

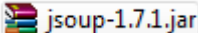
Selector 选择器组合使用

- [el#id](#): 元素+ID, 比如: `div#logo`
- [el.class](#): 元素+class, 比如: `div.masthead`
- [el\[attr\]](#): 元素+class, 比如: `a[href]`
- 任意组合, 比如: `a[href].highlight`
- [ancestor child](#): 查找某个元素下子元素, 比如: 可以用`.body p` 查找在 "body"元素下的所有 `p` 元素

伪选择器 selectors

- [:lt\(n\)](#): 查找哪些元素的同级索引值 (它的位置在 DOM 树中是相对于它的父节点) 小于 `n`, 比如: `td:lt(3)` 表示小于三列的元素
- [:gt\(n\)](#): 查找哪些元素的同级索引值大于 `n`, 比如: `div p:gt(2)`表示哪些 `div` 中有包含 2 个以上的 `p` 元素
- [:eq\(n\)](#): 查找哪些元素的同级索引值与 `n` 相等, 比如: `form input:eq(1)`表示包含一个 `input` 标签的 `Form` 元素

优点: 强大的查询功能 `selector` , API 更新快、可读性强

缺点: 需使用第三方 Jar 包 

2.3 实战案例

```
/**
 * 获取符合规则的字符串
 * @param strHtml 源字符串
 */
private static void getHtmlText(String strHtml) {
    //获得一个Document实例对象
    Document doc = Jsoup.parse(strHtml);
    // Selector选择器
    Elements content = doc.select(".test2");
    //查找元素
    Elements links = content.select("a");
    for (Element link : links) {
        //元素数据
        String linkHref = link.attr("href");
        String linkText = link.text();
        System.out.println(linkHref);
        System.out.println(linkText);
    }
}
```

```
}
```

具体代码 见项目 YangtzehtmlGetSamples

执行结果：

```
http://www.baidu.com Baidu
http://www.baidu.com Baidu
http://www.baidu.com Baidu
http://www.baidu.com Baidu
http://www.baidu.com Baidu
http://www.baidu.com Baidu
http://www.360.com 360|
```

2.4 学习途径

<http://www.open-open.com/jsoup/>

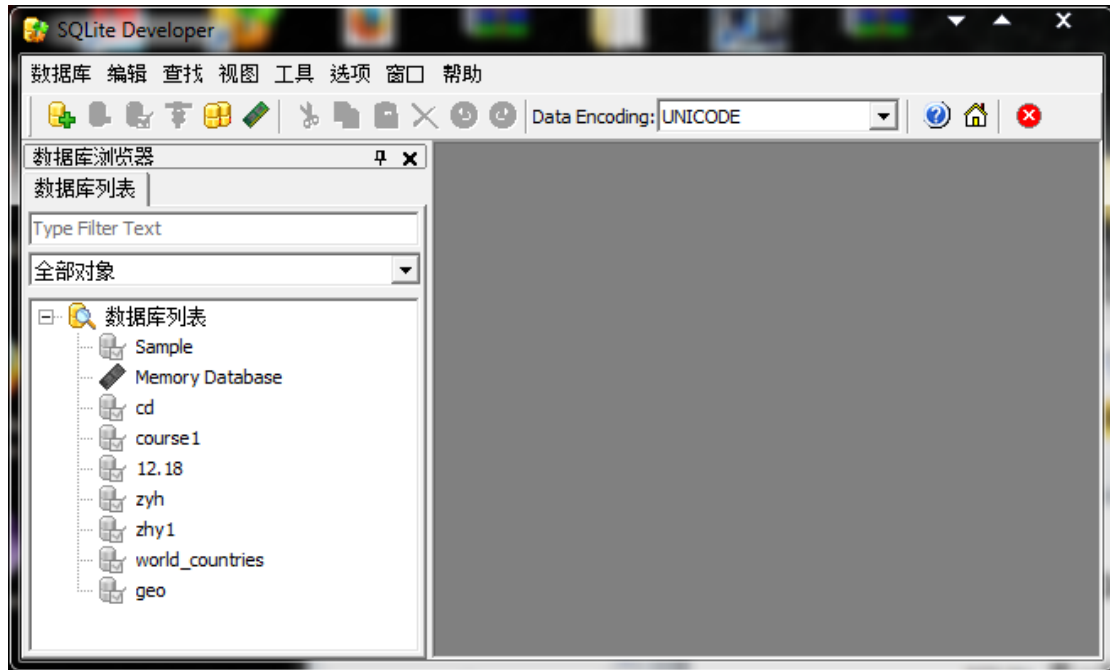
SQLite Databases 数据存储

1 SQLite Databases 介绍

SQLite 是一款轻型的关系数据库。它设计的目标是以嵌入式的方式应用于各种相关产品中。它占用资源非常低，在嵌入式设备中，可能只需要几百 KB 的内存就够了。

Android 在运行时集成了 SQLite，所以每个 Android 应用程序都可以使用 SQLite 数据库。它存储在手机的“data/data/<项目文件夹>/databases/”下。

SQLite 数据库一般在 DOS 窗口下操作，但也有可视化的工具（SQLite Developer），运行界面如下图：



2 SQLite 数据类型

| 数据类型 | 解释 |
|---------|--------|
| NULL | 空类型 |
| INTEGER | 带符号的整型 |
| REAL | 浮点型 |
| TEXT | 字符串文本 |
| BLOB | 二进制对象 |

SQLite3 支持 NULL、INTEGER、REAL (浮点数字)、TEXT(字符串文本)和 BLOB(二进制对象)数据类型，虽然它支持的类型只有五种，但实际上 sqlite3 也接受 varchar(n)、char(n)、decimal(p,s) 等数据类型，只不过在运算或保存时会转成对应的五种数据类型。

SQLite 最大的特点是你可以把各种类型的数据保存到任何字段中，而不用担心字段声明的数据类型是什么。例如：可以在 Integer 类型的字段中存放字符串，或者在布尔型字段中存放浮点数，或者在字符型字段中存放日期型值。但有一种情况例外：定义为 INTEGER PRIMARY KEY 的字段只能存储 64 位整数，当向这种字段保存除整数以外的数据时，将会产生错误。另外，SQLite 在解析 CREATE TABLE 语句时，会忽略 CREATE TABLE 语句中跟在字段名后面的数据类型信息，如下面语句会忽略 name 字段的类型信息：

```
CREATE TABLE person (personid integer primary key autoincrement, name varchar(20))
```

3 使用 SQLiteOpenHelper 对数据库进行版本管理

如何才能实现在用户初次使用或升级软件时自动在用户的手机上创建出应用需要的数据库表呢？如何才能实现在软件升级的时候，对数据表结构进行更新？在 Android 系统，为我们提供了一个名为 SQLiteOpenHelper 的抽象类，必须继承它才能使用，它是通过对数据库版本进行管理来实现前面提出的需求。

为了实现对数据库版本进行管理，SQLiteOpenHelper 类提供了两个重要的方法：

onCreate(SQLiteDatabase db) 前者用于初次使用软件时生成数据库表

onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)，后者用于升级软件时更新数据库表结构。

当调用 SQLiteOpenHelper 的 getWritableDatabase()或者 getReadableDatabase()方法获取用于操作数据库的 SQLiteDatabase 实例的时候，如果数据库不存在，Android 系统会自动生成一个数据库，接着调用 onCreate()方法，onCreate()方法在初次生成数据库时才会被调用，在 onCreate()方法里可以生成数据库表结构及添加一些应用使用到的初始化数据。

onUpgrade()方法在数据库的版本发生变化时会被调用，一般在软件升级时才需改变版本号，而数据库的版本是由程序员控制的，假设数据库现在的版本是 1，由于业务的变更，修改了数据库表结构，这时候就需要升级软件，升级软件时希望更新用户手机里的数据库表结构，为了实现这一目的，可以把原来的数据库版本设置为 2，并且在 onUpgrade()方法里面实现表结构的更新。

```
public void onCreate(SQLiteDatabase db) {
    //执行有更改的sql语句
    db.execSQL("CREATE TABLE person " +
               "(personid integer primary key autoincrement, " +
               "name varchar(20), " +
               "amount integer)");
}

//在数据库版本每次发生变化时都会把用户手机上的数据库表删除，然后再重新创建
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
    db.execSQL("DROP TABLE IF EXISTS person");
    onCreate(db);
}
```

具体代码 见项目 YangtzeDQLiteSamples

4 SQLite 基本操作

SQLite 可以解析大部分标准 SQL 语句，如：

查询语句：select * from 表名 where 条件子句 group by 分组字句 having ...
order by 排序子句

如：select * from person

select * from person order by id desc

select name from person group by name having count(*)>1

插入语句：insert into 表名(字段列表) values(值列表)。如： insert into person(name, age) values('传智' ,3)

更新语句：update 表名 set 字段名=值 where 条件子句。如： update person set name= '传智 ' where id=10

删除语句：delete from 表名 where 条件子句。如： delete from person where id=10

分页 SQL 与 mysql 类似，下面 SQL 语句获取 5 条记录，跳过前面 3 条记录

select * from Account limit 5 offset 3 或者 select * from Account limit 3,5

5 综合图

