# **[Xutils3的使用](http://www.cnblogs.com/wangying222/p/5590437.html)**

Xutils是前两年很火的一个三方库(githup地址),是一个工具类，分为4个模块：DbUtils、HttpUtils、ViewUtils、 BitmapUtils，还有一个非常使用功能就是LogUtils，可以用来输出Log日志。这个第三方库已经很久没有更新了，对最新的6.0系统支持 也不是很好，所以现在出了一个新的Xutils3(githup地址)，今天就来看看新的Xutils3有什么改变。

我们先来看看**Xutils**的4个模块的使用：

使用记得加上权限：

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.INTERNET" />

dBUtils:

用来操作数据库

//创建数据库

DbUtils dUtils = DbUtils.create(MainActivity.this, "Student");

try {//创建表，表名为包名\_StudentInfo

dUtils.createTableIfNotExist(StudentInfo.class);

} catch (DbException e) {

e.printStackTrace();

}

BitmapUtils：

用来加载ImageView的图片资源，可以加载本地、网络

BitmapUtils bUtils = new BitmapUtils(this); bUtils.display(imgv,"http://image.tianjimedia.com/uploadImages/2013/221/JGR58NXB5V62.jpg");

viewUtils:

可以用来代替findViewbyId，也可以用来给控件添加点击事件

// 使用ViewUtils绑定控件

@ViewInject(R.id.radioGroup)

private RadioGroup rg;

@ViewInject(R.id.imgv)

private ImageView imgv;

@ViewInject(R.id.txtv1)

private TextView txtv;

// ViewUtils支持绑定点击事件

@OnClick(R.id.btn)

public void btnShow(View view) {}

记得在Activity的oncreate方法里面写上下面的代码：

// 注入绑定事件

ViewUtils.inject(this);

HttpUtils：

用来上传文件，下载文件

HttpUtils hUtils = new HttpUtils(5 \* 1000);

hUtils.send(HttpMethod.GET, "http://blog.csdn.net/qq379454816", new RequestCallBack<String>() {

@Override

public void onSuccess(ResponseInfo<String> responseInfo) {

txtv.setText(responseInfo.result);

}

@Override

public void onFailure(HttpException error, String msg) {

txtv.setText(error.getMessage() + "," + msg);

}

});

最后来演示一个LogUtils的使用：

//设置是否允许输出对应级别的，false为不输出

LogUtils.allowE=false;

//以error的形式输出日志

LogUtils.e("ViewUtils调用成功！");

ok，就简单地介绍一下其使用，4个主要模块使用还是有一定规律的，除了DbUtils使用create创建对象外其余的3个模块都是使用new创建对象。

下面来看下**Xutils3**：

* xUtils3变化较多所以建立了新的项目不在旧版(github.com/wyouflf/xUtils)上继续维护, 相对于旧版本:
  1. HTTP实现替换HttpClient为UrlConnection, 自动解析回调泛型, 更安全的断点续传策略.
  2. 支持标准的Cookie策略, 区分domain, path...
  3. 事件注解去除不常用的功能, 提高性能.
  4. 数据库api简化提高性能, 达到和greenDao一致的性能.
  5. 图片绑定支持gif, webp; 支持圆角, 圆形, 方形等裁剪, 支持自动旋转...

以上是官方的说明，下面来看具体使用。

Xutils3的使用方式和Xutils有很大的改变，所以请新建一个工程，按照下面的步骤操作:

1、下载类库(类库下载)，将下载文件解压，里面有一个jar包和一个文件夹，都复制到lib目录下。

2、添加权限：网络访问权限和SD卡写入权限，这个和Xutils是一样的

3、然后新建一个Application类，在里面对Xutils3做初始化工作，代码如下：

@Override

public void onCreate() {

super.onCreate();

// 初始化

x.Ext.init(this);

// 设置是否输出debug

x.Ext.setDebug(true);

}

以上弄完就可以进入使用阶段了。

xutils3和Xutils使用方式有一些区别，以前是使用new得到各模块的对象，现在使用：x.模块名()得到对象，回调接口也使用统一的一个，回调接口支持泛型。

**BitMapUtils的使用：**

// 设置加载图片的参数

ImageOptions options = new ImageOptions.Builder()

// 是否忽略GIF格式的图片

.setIgnoreGif(false)

// 图片缩放模式

.setImageScaleType(ScaleType.CENTER\_CROP)

// 下载中显示的图片

.setLoadingDrawableId(R.drawable.ic\_launcher)

// 下载失败显示的图片

.setFailureDrawableId(R.drawable.ic\_launcher)

// 得到ImageOptions对象

.build();

// 加载图片

x.image().bind(imgv, imagUrl, options, new CommonCallback<Drawable>() {

@Override

public void onSuccess(Drawable arg0) {

LogUtil.e("下载成功");

}

@Override

public void onFinished() {

LogUtil.e("下载完成");

}

@Override

public void onError(Throwable arg0, boolean arg1) {

LogUtil.e("下载出错，" + arg0.getMessage());

}

@Override

public void onCancelled(CancelledException arg0) {

LogUtil.e("下载取消");

}

});

// 加载本地图片

// x.image().bind(imgv, "assets://test.gif", options);

// x.image().bind(iv\_big\_img, new File("/sdcard/test.gif").toURI().toString(), imageOptions);

// x.image().bind(iv\_big\_img, "/sdcard/test.gif", imageOptions);

// x.image().bind(iv\_big\_img, "file:///sdcard/test.gif", imageOptions);

// x.image().bind(iv\_big\_img, "file:/sdcard/test.gif", imageOptions);

**HttpUtlis的使用：**

// 请求参数

RequestParams params = new RequestParams("http://www.baidu.com");

x.http().get(params, new CommonCallback<String>() {

@Override

public void onCancelled(CancelledException arg0) {}

@Override

public void onError(Throwable arg0, boolean arg1) {}

@Override

public void onFinished() {}

@Override

public void onSuccess(String arg0) {

// 成功下载，显示到txtv上面

txtv.setText(arg0);

}

});

**viewUtils的使用基本没变，只是绑定点击事件的时候有些出入：**

// 使用ViewUtils绑定控件

@ViewInject(R.id.txtv)

private TextView txtv;

@ViewInject(R.id.imgv)

private ImageView imgv;

**绑定事件：**

/\*\*

\* 使用BitmapUtils显示图片

\* 使用ViewUtils设置按钮的点击事件,方法必须要私有化， 参数格式和type的参数一致,为了混淆方便，方法名要以Event或者Click结尾

\* type可以不写，默认是点击事件类型

\*/

@Event(value = R.id.btn, type = View.OnClickListener.class)

private void btnClick(View view) {

**最后来演示DbUtils：**

先建一个类，用来生成数据库中的表,这里新建一个StudentInfo类(注意里面的@标签的使用,使用这些标签生成表的结构信息)内容如下：

@Table(name = "info")

public class StudentInfo {

@Column(name = "id", isId = true)

private int id;

@Column(name = "name")

private String name;

@Column(name = "age")

private int age;

public int getId() {return id;}

public void setId(int id) {this.id = id;}

public String getName() {return name;}

public void setName(String name) {this.name = name;}

public int getAge() {return age;}

public void setAge(int age) {this.age = age;}}

然后看如何使用：

DbManager.DaoConfig daoConfig = new DaoConfig()

// 数据库的名字

.setDbName("SudentInfo")

// 保存到指定路径

// .setDbDir(new File(Environment.getExternalStorageDirectory().getAbsolutePath()))

// 数据库的版本号

.setDbVersion(1)

// 数据库版本更新监听

.setDbUpgradeListener(new DbUpgradeListener() {

@Override

public void onUpgrade(DbManager arg0, int arg1, int arg2) {

LogUtil.e("数据库版本更新了！");

}

});

DbManager manager = x.getDb(daoConfig);

try {

//创建对象，用来生成表

StudentInfo info = new StudentInfo();

info.setAge(16);

info.setName("小花");

//创建表

manager.saveOrUpdate(info);

} catch (DbException e) {e.printStackTrace();}