# [xUtils更新到3.0后的基本使用规则](http://www.androidchina.net/4177.html)

## 1.xUtils中的IOC框架

使用xUtils的第一步就是必须创建自己的Application类，代码如下：

**public** **class** LYJApplication **extends** Application {

    @Override

**public** **void** onCreate() {

**super**.onCreate();

        x.Ext.init(**this**);//Xutils初始化

    }

}

在AndroidManifest.xml的application标签中添加如下代码：

android:name=”.LYJApplication”

这样初始化就算完成了。

使用IOC框架的代码如下：

**import** org.xutils.view.annotation.ContentView;

**import** org.xutils.view.annotation.Event;

**import** org.xutils.view.annotation.ViewInject;

**import** org.xutils.x;

@ContentView(value = R.layout.activity\_main)

**public** **class** MainActivity **extends** AppCompatActivity {

    @ViewInject(value = R.id.mybut)

**private** Button mybut;

    @Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

        x.view().inject(**this**);

    }

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

**private** **void** onButtonClick(View v){

**switch** (v.getId()){

**case** R.id.mybut:

                Toast.makeText(**this**,"你好我是Xutils的IOC功能",Toast.LENGTH\_SHORT).show();

**break**;

        }

    }

}

需要解释的以下几点：

其一：使用IOC必须全部为私有，不然无效，这里就做演示了，不信你可以把用到IOC框架的注解的成员变量及方法全部换成public ,那么全部会无效，当然除了ContentView例外。

其二，所有用到IOC成员变量，使用的时候，必须在x.view().inject(this)后，如果写在前面，那么程序会崩溃。

## 2.xUtils加载图片功能

现在我们需要设置两个权限，如下：

|  |  |
| --- | --- |
| 1  2 | <**uses-permission** android:name="android.permission.INTERNET" />  <**uses-permission** android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" /> |

接下来就是加载网络图片到imageView中：

x.image().bind(image,”[http://pic.baike.soso.com/p/20090711/20090711101754-314944703.jpg](http://pic.baike.soso.com/p/20090711/20090711101754-314944703.jpg" \o "" \t "http://www.androidchina.net/_blank)“);

也可以设置参数：

ImageOptions imageOptions = **new** ImageOptions.Builder()

        .setSize(DensityUtil.dip2px(120), DensityUtil.dip2px(120))//图片大小

        .setRadius(DensityUtil.dip2px(5))//ImageView圆角半径

        .setCrop(**true**)// 如果ImageView的大小不是定义为wrap\_content, 不要crop.

        .setImageScaleType(ImageView.ScaleType.CENTER\_CROP)

        .setLoadingDrawableId(R.mipmap.ic\_launcher)//加载中默认显示图片

        .setFailureDrawableId(R.mipmap.ic\_launcher)//加载失败后默认显示图片

        .build();

x.image().bind(image, "<http://pic.baike.soso.com/p/20090711/20090711101754-314944703.jpg>",imageOptions);

你也可以将第2个参数设置为图片文件路径，那么将从SD卡中加载图片。

## 3.xUtils操作数据库

我们都知道，一个App中操作数据库的地方有很多，就像是否登录一样，有些地方必须登录后才能操作，那么肯定是全局变量，所以，必须将数据库的初始化放在Application，且必须提供获取数据库的方法，使得在应用程序的任何地方都可以直接获取数据库，并操作数据库，不然重复的获取与释放只能增加内存无谓的消耗。

初始化数据库：

**public** **class** LYJApplication **extends** Application {

**private** DbManager.DaoConfig daoConfig;

**public** DbManager.DaoConfig getDaoConfig() {

**return** daoConfig;

    }

    @Override

**public** **void** onCreate() {

**super**.onCreate();

        x.Ext.init(**this**);//Xutils初始化

        daoConfig = **new** DbManager.DaoConfig()

                .setDbName("lyj\_db")//创建数据库的名称

                .setDbVersion(1)//数据库版本号

                .setDbUpgradeListener(**new** DbManager.DbUpgradeListener() {

                    @Override

**public** **void** onUpgrade(DbManager db, **int** oldVersion, **int** newVersion) {

                        // TODO: ...

                        // db.addColumn(...);

                        // db.dropTable(...);

                        // ...

                    }

                });//数据库更新操作

    }

}

上面的注释明了，有必要说明的一点是setDbDir(new File(“/sdcard”))，可以将数据库存储在你想存储的地方，如果不设置，那么数据库默认存储在/data/data/你的应用程序/database/xxx.db下。这里我们就默认放在应用程序下。

我们首先创建一个实体类，如下：

@Table(name="lyj\_person")

**public** **class** LYJPerson {

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

**private** **int** id;

    @Column(name = "name")

**private** String name;

    @Column(name = "age")

**private** String age;

**public** String getAge() {

**return** age;

    }

**public** **void** setAge(String age) {

**this**.age = 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;

    }

}

通过实体类可以直接操作数据库。

我们在Application中加入如下代码，向数据库添加数据：

DbManager db = x.getDb(daoConfig);

LYJPerson person1=**new** LYJPerson();

person1.setName("liyuanjinglyj");

person1.setAge("23");

LYJPerson person2=**new** LYJPerson();

person2.setName("xutilsdemo");

person2.setAge("56");

**try** {

    db.save(person1);

    db.save(person2);

} **catch** (DbException e) {

    e.printStackTrace();

}

在Activity中操作获取数据库数据的代码如下：

DbManager db = x.getDb(((LYJApplication)getApplicationContext()).getDaoConfig());

**try** {

    List<LYJPerson> lyjPersons=db.selector(LYJPerson.**class**).findAll();

**for** (**int** i=0;i<lyjPersons.size();i++){

        Log.i("liyuanjinglyj","LYJPerson"+i+".name="+lyjPersons.get(i).getName());

        Log.i("liyuanjinglyj","LYJPerson"+i+".name="+lyjPersons.get(i).getAge());

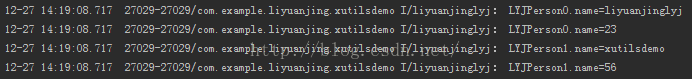
    }

} **catch** (DbException e) {

    e.printStackTrace();

}

那么肯定会得到如下结果：



## 4.xUtils的网络请求

Android规定UI线程是不能涉及网络任务的，所以，这里主要简单介绍Xutils的异步网络请求，同步的自行探究。

使用格式如下：

RequestParams params = **new** RequestParams("<http://blog.csdn.net/mobile/experts.html>");

x.http().get(params, **new** Callback.CommonCallback&lt;String&gt;() {

    @Override

**public** **void** onSuccess(String result) {

        Document doc = Jsoup.parse(result);

        Element div = doc.select("div.list\_3").get(0);

        Elements imgs = div.getElementsByTag("img");

**for** (**int** i = 0; i &lt; imgs.size(); i++) {

            Element img = imgs.get(i);

            Log.i("liyuanjinglyj",img.attr("alt"));

        }

    }

    @Override

**public** **void** onError(Throwable ex, **boolean** isOnCallback) {

    }

    @Override

**public** **void** onCancelled(Callback.CancelledException cex) {

    }

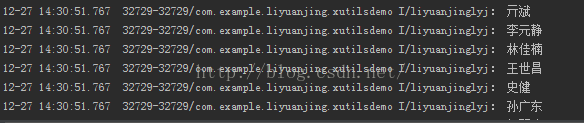
    @Override

**public** **void** onFinished() {

    }

});

这里获取的是CSDN移动博客专家的HTML页面信息，看看下面的日志，就知道Xutils网络功能还是很强大的。



本文最后附带了一下粗略模仿CSDN APP的源码，有意者可以下载看看，里面用到另一个开发框架，我用来专门处理图片的（afinal）。都说xUtils是afinal的进化版，不过在图片方面，我们觉得xUtils还有点不足。

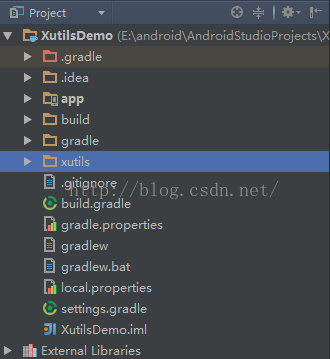
[http://download.csdn.net/detail/liyuanjinglyj/9379103](http://download.csdn.net/detail/liyuanjinglyj/9379103" \o "" \t "http://www.androidchina.net/_blank)

## 5.导入xUtils工程到Android Studio

下载地址如下：

[https://github.com/wyouflf/xUtils3/tree/master](https://github.com/wyouflf/xUtils3/tree/master" \o "" \t "http://www.androidchina.net/_blank)

㈠将下载的工程复制到Project目录下：



㈡添加到settings.gradle文件：

include ‘:app’,':xutils’

㈢编译到工程中

dependencies {

compile fileTree(dir: 'libs', include: ['\*.jar'])

compile 'com.android.support:appcompat-v7:23.0.1'

compile project(':xutils')

}

㈣将xutils文件夹下的build.gradle中的版本与最低版本调整到与创建工程一致

compileSdkVersion 23

buildToolsVersion "23.0.1"

defaultConfig {

minSdkVersion 15

targetSdkVersion 23

versionCode 20151224

versionName version

}

㈤添加如下代码到build.gradle（Project:XutilsDemo）中

dependencies {

classpath 'com.android.tools.build:gradle:1.3.0'

classpath 'com.jfrog.bintray.gradle:gradle-bintray-plugin:1.2'

classpath 'com.github.dcendents:android-maven-gradle-plugin:1.3'

// NOTE: Do not place your application dependencies here; they belong

// in the individual module build.gradle files

}

其中红色标记为添加的代码。

点击Sync now就可以使用xUtils了。