**Android项目框架简介**

**Tips：本简介中尚未对android studio的配置进行介绍**

**一.DbUtils模块**

**1.android中的orm框架，一行代码就可以进行增删改查；**

**2.支持事务，默认关闭；**

**3.可通过注解自定义表名，列名，外键，唯一性约束，NOT NULL约束，CHECK约束等（ 需要混淆的时候请注解表名和列名）；**

**4.支持绑定外键，保存实体时外键关联实体自动保存或更新；**

**自动加载外键关联实体，支持延时加载；**

**5.支持链式表达查询，更直观的查询语义。**

**二.Http模块**

**1.android中的orm框架，一行代码就可以进行增删改查；**

**2.支持同步，异步方式的请求；**

**3.支持大文件上传，上传大文件不会oom；**

**4.支持GET，POST，PUT，MOVE，COPY，DELETE，HEAD，OPTIONS，TRACE，CONNECT 请求**

**5.下载支持301/302重定向，支持设置是否根据Content-Disposition重命名下载的文件；**

**返回文本内容的请求(默认只启用了GET请求)支持缓存，可设置默认过期时间和针对 当前请求的过期时间。**

**三.Bitmap图片模块**

**1. 加载bitmap的时候无需考虑bitmap加载过程中出现的oom和android容器快速滑 动时候出现的图片错位等现象；**

**2.支持加载网络图片和本地图片；**

**3.内存管理使用lru算法，更好的管理bitmap内存；**

**4.可配置线程加载线程数量，缓存大小，缓存路径，加载显示动画等...(具体的设置在后 续使用方法中)**

**Tips:框架所需权限**

**<!-- 访问账户Gmail列表权限 -->**

**<uses-permission android:name="android.permission.GET\_ACCOUNTS" />**

**<!-- 使用证书权限 -->**

**<uses-permission android:name="android.permission.USE\_CREDENTIALS" />**

**<!-- 管理账户权限 -->**

**<uses-permission android:name="android.permission.MANAGE\_ACCOUNTS" />**

**<!-- 验证账户权限 -->**

**<uses-permission android:name="android.permission.AUTHENTICATE\_ACCOUNTS" />**

**<uses-permission android:name="com.android.launcher.permission.READ\_SETTINGS" />**

**<!-- 连续广播权限 -->**

**<uses-permission android:name="android.permission.BROADCAST\_STICKY" />**

**<!-- 读写系统设置权限 -->**

**<uses-permission android:name="android.permission.WRITE\_SETTINGS" />**

**<!-- 这个权限用于进行网络定位 -->**

**<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />**

**<!-- 这个权限用于访问GPS定位 -->**

**<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />**

**<!-- 用于访问wifi网络信息，wifi信息会用于进行网络定位 -->**

**<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />**

**<!-- 获取运营商信息，用于支持提供运营商信息相关的接口 -->**

**<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />**

**<!-- 这个权限用于获取wifi的获取权限，wifi信息会用来进行网络定位 -->**

**<uses-permission android:name="android.permission.CHANGE\_WIFI\_STATE" />**

**<!-- 写入扩展存储，向扩展卡写入数据，用于写入离线定位数据 -->**

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

**<!-- 访问网络，网络定位需要上网 -->**

**<uses-permission android:name="android.permission.INTERNET" />**

**<!-- 允许应用读取低级别的系统日志文件 -->**

**<uses-permission android:name="android.permission.READ\_LOGS" />**

**<!-- 在SDCard中创建与删除文件权限 -->**

**<uses-permission android:name="android.permission.MOUNT\_UNMOUNT\_FILESYSTEMS" />**

**<!-- 读写手机状态和身份 -->**

**<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />**

**<!-- 向外拨打电话的权限 -->**

**<uses-permission android:name="android.permission.CALL\_PHONE" />**

**<!-- 调用照相机相关权限 -->**

**<uses-permission android:name="android.permission.CAMERA" />**

**<!-- 发送短信权限 -->**

**<uses-permission android:name="android.permission.SEND\_SMS" />**

**<!-- 读送短信权限 -->**

**<uses-permission android:name="android.permission.READ\_SMS" />**

**<!-- 写短信权限 -->**

**<uses-permission android:name="android.permission.WRITE\_SMS" />**

**<!-- 接收短信权限 -->**

**<uses-permission android:name="android.permission.RECEIVE\_SMS" />**

**<permission**

**android:name="oem.permission.SENDMAIL"**

**android:protectionLevel="normal" />**

**<uses-feature android:name="android.hardware.camera" />**

**<uses-feature**

**android:name="android.hardware.camera.autofocus"**

**android:required="false" />**

**<!-- Push service 运行需要的权限 -->**

**<!-- 开机启动权限 -->**

**<uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED" />**

**<!-- 振动权限 -->**

**<uses-permission android:name="android.permission.VIBRATE" />**

**<uses-permission android:name="android.permission.ACCESS\_DOWNLOAD\_MANAGER" />**

**<uses-permission android:name="android.permission.DOWNLOAD\_WITHOUT\_NOTIFICATION" />**

**<!-- 禁用键盘锁权限 -->**

**<uses-permission android:name="android.permission.DISABLE\_KEYGUARD" />**

**<!-- 更改系统设置的权限 -->**

**<uses-permission android:name="android.permission.CHANGE\_CONFIGURATION" />**

**<!-- 访问闪光灯的权限 -->**

**<uses-permission android:name="android.permission.FLASHLIGHT" />**

**<!-- 发送短信验证码的权限 -->**

**<uses-permission android:name="android.permission.READ\_CONTACTS" />**

**<uses-permission android:name="android.permission.RECEIVE\_SMS" />**

**<!-- 统计SDK用的 -->**

**<uses-permission android:name="android.permission.GET\_TASKS" />**

**DbUtils使用方法：**

**DbUtils使用方法：**

|  |  |
| --- | --- |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12**  **13**  **14**  **15**  **16**  **17**  **18**  **19**  **20**  **21**  **22**  **23**  **24**  **25**  **26**  **27**  **28**  **29**  **30**  **31** | **DbUtils db = DbUtils.create(this);**  **User user = new User(); //这里需要注意的是User对象必须有id属性，或者有通过@ID注解的属性**  **user.setEmail("wyouflf@qq.com");**  **user.setName("wyouflf");**  **db.save(user); // 使用saveBindingId保存实体时会为实体的id赋值**  **...**  **// 查找**  **Parent entity = db.findById(Parent.class, parent.getId());**  **List<Parent> list = db.findAll(Parent.class);//通过类型查找**  **Parent Parent = db.findFirst(Selector.from(Parent.class).where("name","=","test"));**  **// IS NULL**  **Parent Parent = db.findFirst(Selector.from(Parent.class).where("name","=", null));**  **// IS NOT NULL**  **Parent Parent = db.findFirst(Selector.from(Parent.class).where("name","!=", null));**  **// WHERE id<54 AND (age>20 OR age<30) ORDER BY id LIMIT pageSize OFFSET pageOffset**  **List<Parent> list = db.findAll(Selector.from(Parent.class)**  **.where("id" ,"<", 54)**  **.and(WhereBuilder.b("age", ">", 20).or("age", " < ", 30))**  **.orderBy("id")**  **.limit(pageSize)**  **.offset(pageSize \* pageIndex));**  **// op为"in"时，最后一个参数必须是数组或Iterable的实现类(例如List等)**  **Parent test = db.findFirst(Selector.from(Parent.class).where("id", "in", new int[]{1, 2, 3}));**  **// op为"between"时，最后一个参数必须是数组或Iterable的实现类(例如List等)**  **Parent test = db.findFirst(Selector.from(Parent.class).where("id", "between", new String[]{"1", "5"}));**  **DbModel dbModel = db.findDbModelAll(Selector.from(Parent.class).select("name"));//select("name")只取出name列**  **List<DbModel> dbModels = db.findDbModelAll(Selector.from(Parent.class).groupBy("name").select("name", "count(name)"));**  **...**  **List<DbModel> dbModels = db.findDbModelAll(sql); // 自定义sql查询**  **db.execNonQuery(sql) // 执行自定义sql**  **...** |

**http模块使用方法：**

**使用开源开源项目：详见tools文件夹中toolhttp.java文件里面具体使用方法。**

**1.下载文件**

ToolHTTP.*get*(**this**, "http://www.baidu.com", **null**,**new** BinaryHttpResponseHandler() {

@Override

**public** **void** onSuccess(**int** arg0, ~~Header~~[] arg1, **byte**[] arg2) {

}

@Override

**public** **void** onFailure(**int** arg0, ~~Header~~[] arg1, **byte**[] arg2, Throwable arg3) {

}

@Override

**public** **void** onProgress(**int** bytesWritten, **int** totalSize) {

**super**.onProgress(bytesWritten, totalSize);

}

});

### **2.get方法**

ToolHTTP.get(apiAdress, photoAndHashparams2,

**new** JsonHttpResponseHandler() {

@SuppressWarnings("unchecked")

@Override

**public** **void** onSuccess(**int** statusCode, Header[] headers,

JSONObject response) {

}

@Override

**public** **void** onFailure(**int** statusCode, Header[] headers,

String responseString, Throwable throwable) {

}

@Override

**public** **void** onSuccess(**int** statusCode, Header[] headers,

JSONArray timeline) {

}

});

3.post方法

ToolHTTP.post(apiAdress, photoAndHashparams2,

**new** JsonHttpResponseHandler() {

@SuppressWarnings("unchecked")

@Override

**public** **void** onSuccess(**int** statusCode, Header[] headers,

JSONObject response) {

}

@Override

**public** **void** onFailure(**int** statusCode, Header[] headers,

String responseString, Throwable throwable) {

}

@Override

**public** **void** onSuccess(**int** statusCode, Header[] headers,

JSONArray timeline) {

}

});

Tips:(具体的使用方法参见toolhttp.java文件的工具类)

**Bitmap图片模块使用方法：**

**首先，初始化bitmap模块:**

ToolImage.initImageLoader(getApplicationContext());

**具体使用方法：**

**（1）.直接加载**

1. **public void displayImage(String uri, ImageAware imageAware) {**
2. **displayImage(uri, imageAware, null, null, null);**
3. **}**

**（2）.** **选项加载**

1. **public void displayImage(String uri, ImageAware imageAware, DisplayImageOptions options) {**
2. **displayImage(uri, imageAware, options, null, null);**
3. **}**

**（3）.选项+监听加载：**

1. **public void displayImage(String uri, ImageAware imageAware, DisplayImageOptions options,**
2. **ImageLoadingListener listener) {**
3. **displayImage(uri, imageAware, options, listener, null);**
4. **}**

**获取其他路径的方法，将路径更改：**

"http://site.com/image.png" // from Web

"file:///mnt/sdcard/image.png" // from SD card

"file:///mnt/sdcard/video.mp4" // from SD card (video thumbnail)

"content://media/external/images/media/13" // from content provider

"content://media/external/video/media/13" // from content provider (video thumbnail)

"assets://image.png" // from assets

"drawable://" + R.drawable.img // from drawables (non-9patch images)

**详细使用方法https://github.com/nostra13/Android-Universal-Image-Loader**

**项目具体的命名方法：**

**1.com+公司名+工程名称+activity**

**示例:com.itteam.roadlamp.activity**

**作用:主要存放项目activity和fragment界面**

**2.com+公司名+工程名称+ service**

**示例:** **com.itteam.roadlamp.service**

**作用:** **activity相关的类如监听器，线程，适配器等非常多的类**

**3.com+公司名+工程名称+ manager**

**示例: com.itteam.roadlamp.manager**

**作用:主要操作用户的管理model类。**

**4.com+公司名+工程名称+ base**

**示例:**