玉柴云移动端开发说明文档

撰写人：张林伟
邮箱：zhanglinwei@cloudist.cc
电话：18104081739

一. 开发准备

开发环境

- 系统要求：均可
- 开发工具：Android Studio 1.2+
- 构建工具：Gradle , 具体配置详见 `./build.gradle` 和 `./app/build.gradle` 文件

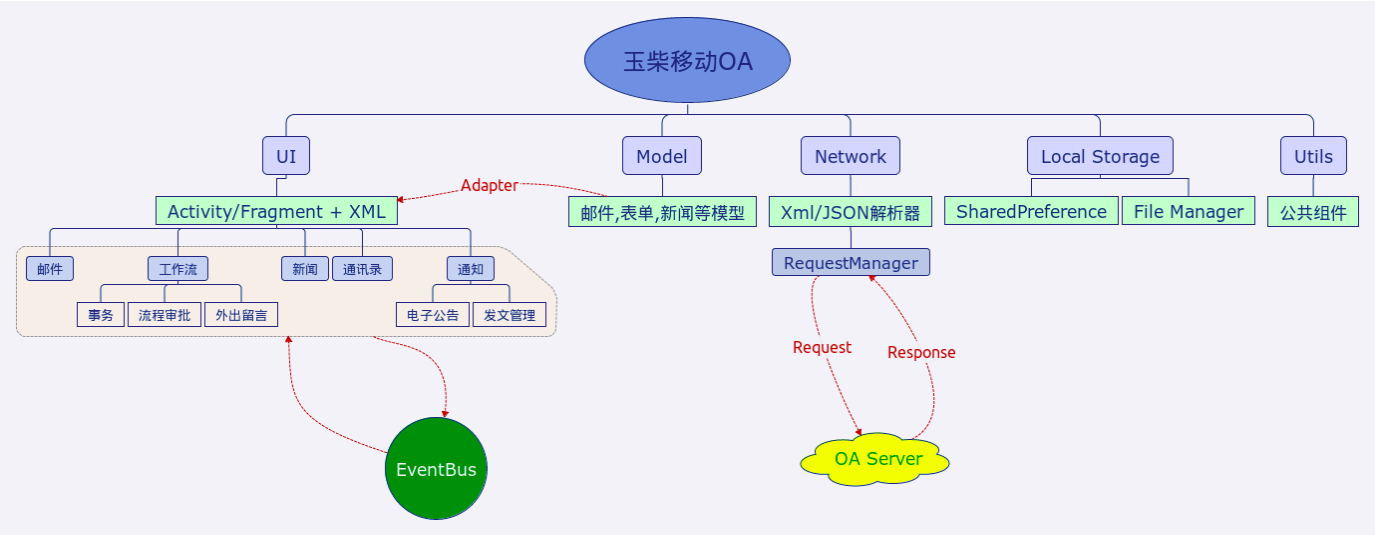
使用的第三方开源库

项目使用的所有第三方库详见 `./app/build.gradle` 文件下的 `dependencies` 字段。以下几个在开发中需要重点了解：

- Gson:：json自动解析工具 https://github.com/google/gson
- ButterKnife：视图绑定注入工具（注：这个库更新比较频繁，OA使用如下旧版本 http://www.cnblogs.com/mengdd/p/4595973.html ）
- Volley：异步网络请求框架 https://github.com/mcxiaoke/android-volley

- EventBus：全局消息总线 http://www.jianshu.com/p/2b2f2b26b810

二. 整体架构



系统的整体架构如图，UI部分由各种`Activity`和`Fragment`作为展示前端，同时还负责视图逻辑处理操作。在对功能模块进行修改的时候首先要找到他们对应的`Activity`和`Fragment`就行了，比如，事务相关的都是以`WorkFlow`开头，邮件相关以`Mail`开头，联系人相关以`Contact`开头。

Model是所有定义的抽象模型，他们之间以Adapter和UI层进行适配，大部分是列表方式展示。同时，在Model中还定义了模型的部分处理逻辑，比如，服务器返回的xml在解析的时候会找到他们对应的Model中的`parse`方法，这在后面的网络请求模块中会介绍到。

Network模块基于Volley框架，与玉柴OA网页版的Server进行通信，获取JSON或者xml数据来填充Model。

Local Storage是应用的本地存储，目前应用所使用的本地存储主要有`SharedPreferences`和文件存储，`SharedPreferences`用于保存登录用户的信息，节点信息等，文件存储主要是保存邮件的附件，电子公告和发文管理的附件，以及事务的附件。

Utils是一些公共组件，比如时间日期格式化，全局对话框等。

三. 模块开发

由于篇幅有限，以下只介绍几个较为重点的开发模块：

1. 网络请求

网络请求相关类请参阅`cc.cloudist.yuchaioa.network`包中的文件，例如：

- `LoginHandler` 登录操作
- `RequestManager` 负责所有网络请求的管理，定义了接口的传参以及调用
- `XRequest<T extends XParser>` 所有以xml为返回的请求，返回T类型，T类型必须实现xml的解析接口，具体实现实例有`MailList`，`ReceiverList`等
- `ApiRequest<T>` 所有以json为返回的请求，这里涉及到Gson的使用
- `StatManager` 用户行为统计请求的管理，目前只包含登录统计

2. 表单事务（OA待办）

表单格式化显示

由于表单类型多样，采用了WebView展示 + JavaScript格式化的方式进行。具体参考`WorkFlowDetailFragment`，其中，表单详情页格式化所使用的JS文件位于`./app/src/main/assets/workflow.js`，我们把表单分为了几种类型进行格式化，并且对一部分太过于复杂表单提示不支持。

类似的`WebView`格式化展示的方式还用在电子公告的显示上。

表单操作

表单流程操作（转发，审批提交，挂起，退回，直送）涉及到`WebView`中js和`WorkFlowDetailFragment`的交互，解决方案是采用了`JSBridge`技术。

1. Fragment 操作 WebView，使用`mWebView.loadJs(JS_CODE)`方法，使用场景如调用格式化的本地js文件，提交操作，退回挂起等操作。
2. WebView 回调 Fragment，在fragment中为WebView注册`BridgeInterface`,这样在JS中可以直接调用`BridgeInterface`中定义的回调方法，在JS中的使用方式如`bridge.showTipBox(op.text);`。使用场景如 ajax 请求传回，页面中的下载文件事件回调，web中显示提示对话框等。

3. 发文管理

`PostsDetailActivity`：由于网页版的发文管理是在网页上展示PDF文件，为了在移动端能够正常显示，我们将PDF文件先下载到本地，再使用`android-pdfview`(https://github.com/JoanZapata/android-pdfview)插件进行渲染。

4. 组织架构选择

`RecvSelectActivity`:组织架构的选择在邮件选择收件人以及表单选择提交人等时候都有用到，包含人员，部门，群组，节点四类。由于涉及到节点的展开，在UI的实现上使用了`RecyclerView`列表动态加载，每当点击展开的时候请求下一层的列表项，然后动态添加到列表中，但逻辑上`Receiver`之间是树状的数据结构。

四. 打包发布

首先确保根目录下的`./yuchaioa.jks`密钥文件存在，打包apk可使用以下两种方式:

命令行方式打包(推荐)

该打包方式构建需要`make`环境，建议在`OSX`或者`Linux`下执行。命令行切换到根目录下执行命令:

```
# beta包
make beta_build APK=yuchaioa.xxx.apk OUTPUT_DIR=~/

# release包
make release_build APK=yuchaioa.xxx.apk OUTPUT_DIR=~/
```

其中`APK`为包的名字，`OUTPUT_DIR`为你需要输出文件的目录，上面的例子即可在`~/`目录下生成名为`yuchaioa.xxx.apk`的apk包。

在Android Studio中打包

在 菜单->Build->Generate Signed APk->选择`./yuchaioa.jks`签名密钥文件，输入密码及alias（值均为：`yuchaioa`），开始打包。

推送更新

打包apk之后，在umeng的网页端上传apk发布更新日志，即可自动为旧版本用户推送更新。为了便于应用下载，应用还需要在蒲公英平台（https://www.pgyer.com/9i8l）进行上传发布。