App-XLLib重构计划（Android）

一：搭建**Gradle**和**Sonatype Nexus** 搭建私有**maven**仓库

要求：

- 提供可以访问的**maven**地址
- 在**wiki**上提供搭建教程（输出地址：Android-知识体系）

二：**gitlab**上创建**XLLib**项目 命名规范：**XLLib**

要求：

- 支持脚本上传至**maven**仓库
- 在三个**app**中分别插拔可以运行

三：配置信息迁移（方便各个**app**添加）

由于**XLLib**为公共组件，尽量避免频繁发版本，为了方便对外添加或者修改配置信息，采用对外提供接口的方式进行添加或者修改。**XLLib**只维护一些基本的配置信息。

要求：

- **app**添加配置信息不需要每次上传**lib maven**版本。
- **app**各自管理各自独立的配置信息

方案1：

SettingUtil，对外公开一些特定业务组件的api函数，业务组件只需管理对应配置信息的key即可

方案2：

继承SettingUtil，新增一些属于特定业务组件的api函数。

四：文件处理整理

规范一下具体文件存放地址，区别开哪些是临时文件，哪些是永久保存的，方便做淘汰清理处理，目录层级要做到很容易查找。

目录大致分以下几个

- **voices** 音频文件目录
- **images** 图片文件目录
- **videos** 视频文件目录
- **uploads** 上传文件临时目录
- **downloads** 下载文件存放目录

要求实现以下几项：

- 目录层级管理
- 缓存有效期管理
- 淘汰清理机制

五：数据库方案调整

- **废除多种关系型数据库实现方式**
- 统一采用**GreenDao**关系型数据库
- 调整数据库的等级，非共享数据库放在具体业务组件中，通过**contentprovider**的方式对外提供共享数据接口

六：废除**Xutils**开源框架

项目未来采用解决方案遵循单一职责原则，项目中已经做了部分重构替代了**xutils**，本次查找哪些还在依赖**xutils**选择下面的替换方案。

要求：

- 图片替换成**glide**
- 网络处理替换成**okHttp**
- task任务替换成**XLExecutor**
- 数据库替换成 **GreenDao**

- Cache替换成**XLCache**

废除原因：

- **Xutils**职责太多，在项目太重
- **Xutils**已经基本上不再维护了
- 目前项目中逐步已经不再使用

七：事件总线方案替换

目前项目中采用**RxJava**实现的**RXBus**，针对宿主观察事件实现相对比较麻烦，而且实现多个事件观察更是导致代码不够简洁，由于采用**RxJava1** 对频繁发送事件支持并不理想。

要求：

- 废除**RxBus**替换成**EventBus**
- 同时**apt** 方案替换成**Android annotationProcessor**

八：路由方案替换

目前方案，通过单例设计模式，管理S**cheme Host**，然后拼接具体的业务path，参数采用**HashMap**管理相对来说比较隐式，路由接口地址如何方便管理，避免频繁修改路由导致的频繁上传 **maven**版本。

方案1：

通过对跳转目标进行一个**Scheme Url注解声明，然后通过编译生成代码的方式，管理起来URL与目标之间的关系，具体跳转调用编译生成的XLRouter进行相关业务跳转**

方案2：

类似目前网络解耦的方式，采用aop编程思想，注解反射机制，通过Java动态代理机制，动态实例化路由接口类。

九：模块之间数据通讯

业务模块之间传递参数，目前基于RxBus，会在XLLib中管理太多的Event类，导致频繁修改XLLib

- **模块之间通信不维护具体事件**

- **数据通知**

方案1：

通过广播的方式

方案2：

通过EventBus

十：**Tab**项动态加载

动态配置app壳工程的一级页面。目前采用硬编码的方式引入Fragment，不符合动态插拔逻辑。

- **通过配置文件加载**
- **不能硬编码**

方案：

基于运行时，然后通过功能关系映射表，动态实例化相应的目标页。