# Room + LiveData 提交数据案例

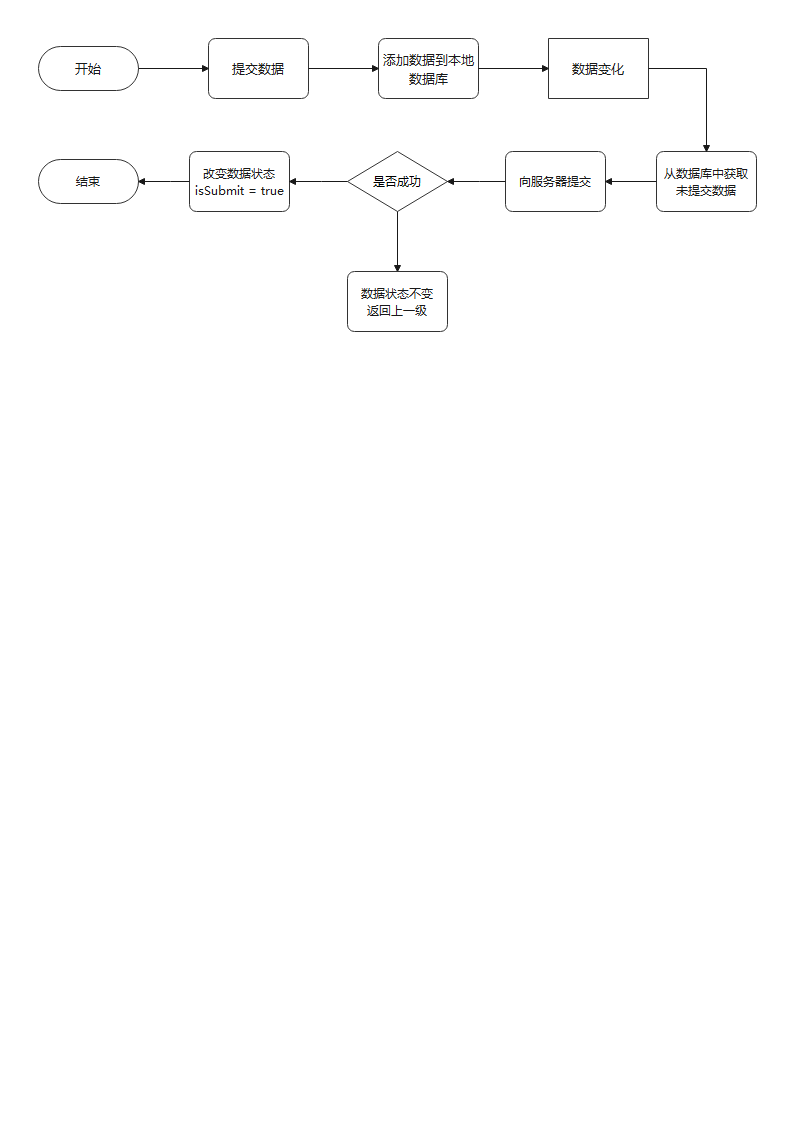
### 技术栈

AS：Android Studio Chipmunk | 2021.2.1 Patch 1；Gradle：7.2.1；

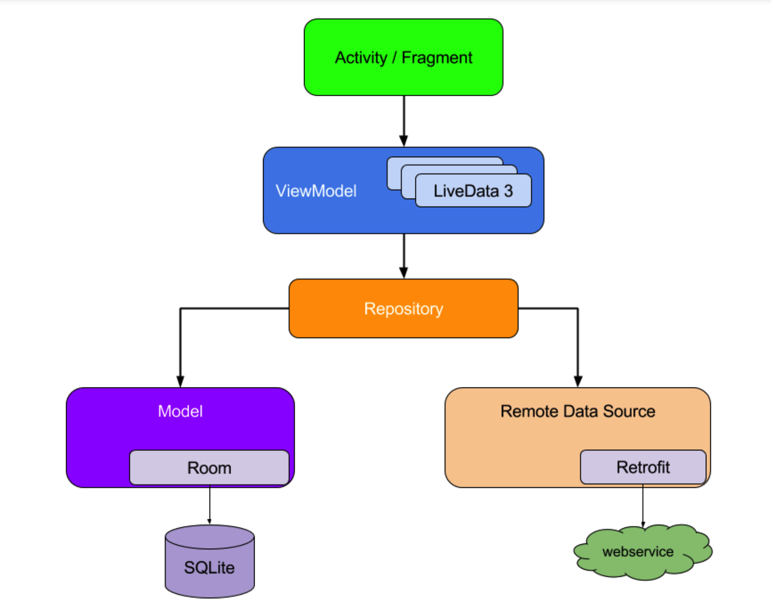
编程语言：Kotlin 1.6.10；

项目架构：MVVM；

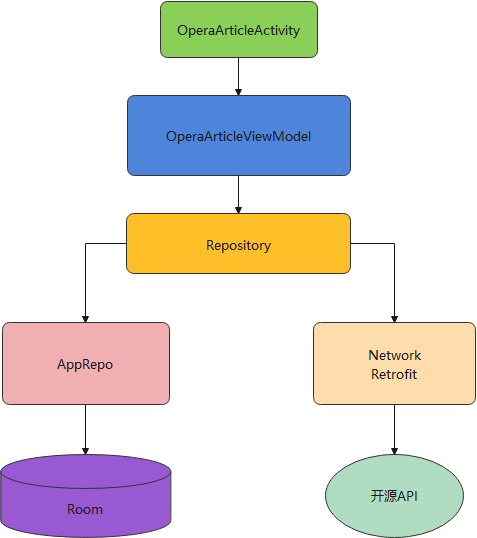
### 流程图



### MVVM架构图



### 案例架构图



### 相关框架

**Lifecycle**

包括viewmodel、Livedata、runtime组件。

**Coroutines**

包括coroutines-android、coroutines-core组件。

**Room**

数据库框架。

**Retrofit**

网络请求框架。

### 数据表说明

案例中设计了一张文章表 tb\_article，字段包括 articleId [Long]（文章ID）、title [String]（文章标题）、content [String]（文章内容）、isSubmit [Boolean]（是否提交），其中ID字段是主键并且自增。

### 关键类

OperaArticleActivity：添加文章数据页面，主要用于提交文章数据；

OperaArticleViewModel：文章操作页面，*数据与UI* 交互层，主要用于保存 *添加*或*修改* 文章数据变量；

Repository：仓库层，主要用于处理网络请求数据，对请求到的数据进行处理；

Network：网络请求层，主要用于构建请求接口动态对象，封装网络请求结果；

Article：文章数据实体类，同时也是 Room 的 Entity实体；

ArticleDao：文章数据库Dao层，Room 的 Dao 接口；

ArticleDatabase：文章数据库管理类，单例模式访问，Room的 Database类；

AppRepo：数据库Dao层工具类，便捷处理对本地数据的操作。

### 层级说明

**网络请求层**

创建ServiceCreator（Retrofit构建器）、创建DefaultServices（api接口）；编写Network（网络请求工具），此处主要是统一网络请求入口，对Retrofit请求结果进行统一封装。

**仓库层**

创建Repository类，定义添加文章方法，方法中通过liveData函数，将结果通知到观察者。

**数据与UI交互层**

创建OperaArticleViewModel类，定义相关liveData变量与操作方法。

**视图UI层**

仅持有VieModel层，将相关操作通过ViewModel去传达；自身观察需监听的变量即可。