

New UI

RxJava vs. Request

- Request方案的不足
 - 线程调度不如Rx方便
 - 链式调用不如Rx方便
 - 需要RequestManager以及重复定义逻辑Runnable
 - Context的处理存在可能的泄露

RxRequest

```
public void execute(final RxCallback<T> callback) {
    createObservable()
        .observeOn(observeOnScheduler())
        .subscribeOn(subscribeScheduler())
        .doFinally(new Action() {
            @Override
            public void run() throws Exception {
                doFinally();
            }
        })
        .subscribe(new Consumer<T>() {
            @Override
            public void accept(T t) throws Exception {
            }
        }, new Consumer<Throwable>() {
            @Override
            public void accept(Throwable throwable) throws Exception {
            }
        }, new Action() {
            @Override
            public void run() throws Exception {
            }
        })
    );
}
```


RxRequest的使用

```
QueryArgs args = QueryBuilder.libraryAllBookQuery(null, ...);  
RxMetadataRequest request = new RxMetadataRequest(dataManager, args, false);  
request.execute(new RxCallback<RxMetadataRequest>() {  
    @Override  
    public void onNext(RxMetadataRequest rxMetadataRequest) {  
    }  
});
```


MVC

- ✦ 数据与UI一定要彻底分离
- ✦ Model (不要依赖view和controller)
- ✦ View (不要依赖controller, 对于model的依赖, 降低到基本数据结构的依赖)
- ✦ Controller (使用Model 和 View)
- ✦ 对于复杂应用使用action

Action & RxRequest

- ✦ 数据与UI要彻底分离
- ✦ RxRequest主要处理数据，不涉及UI，便于开发测试与重用
- ✦ Action与具体的应用逻辑绑定，涉及界面，分解activity的功能，将activity中的函数，分散到各个Action类，一个类完成一个功能（函数->类）

Data Binding

- 分层，逻辑清晰，关注数据
- 减少界面代码
- 提高资源可重用性

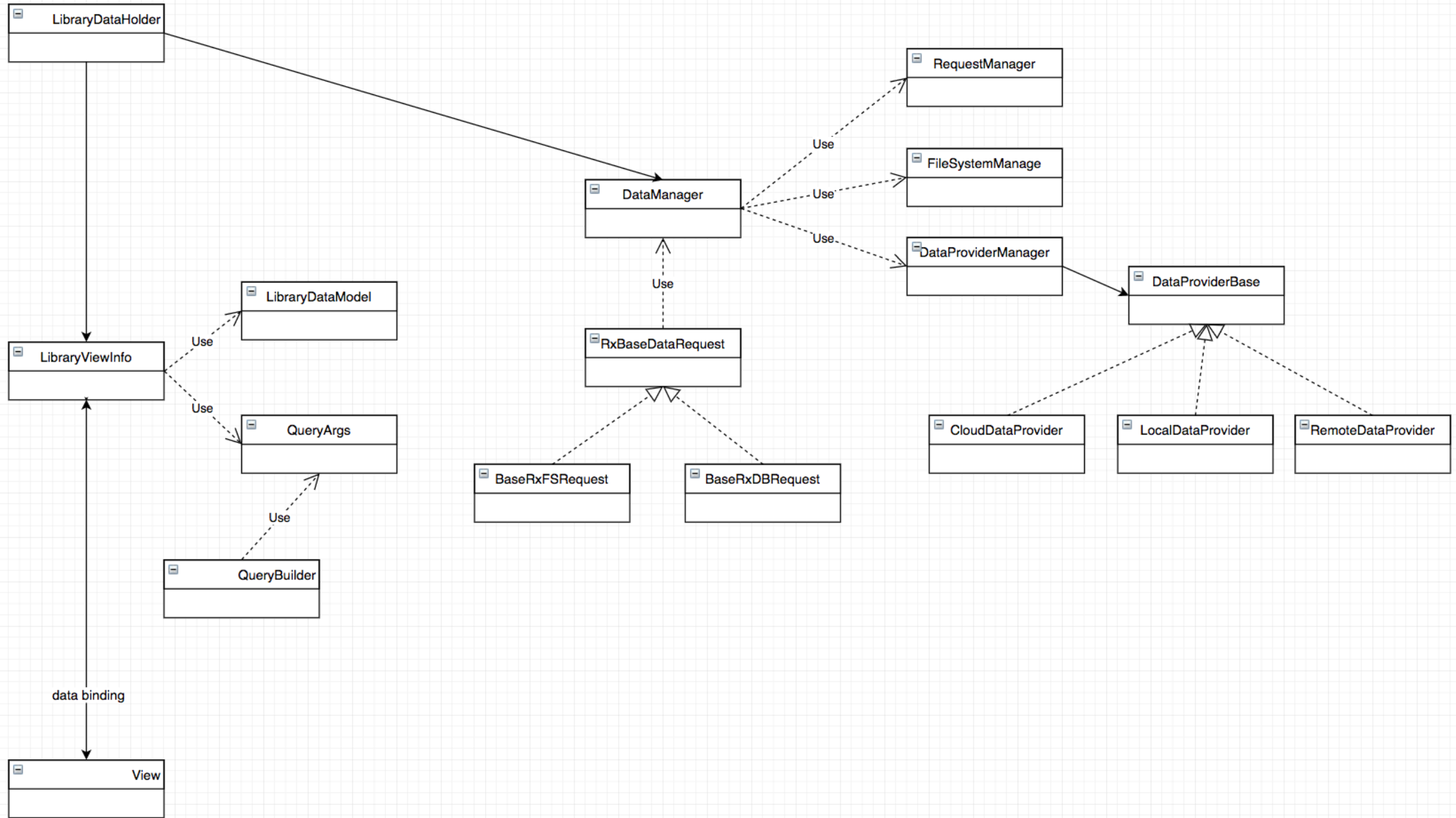
Menu

✦ Menu -> MenuItem -> menu.xml

```
<data>
    <import type="com.onyx.android.sdk.ui.data.MenuId"/>
    <import type="com.onyx.android.sdk.ui.data.MenuItem"/>
    <import type="android.util.SparseArray"/>
    <import type="com.onyx.edu.note.R"/>

    <variable
        name="item"
        type="SparseArray<MenuItem>"/>
</data>

<include
    bind:item="@{item[MenuId.NORMAL_PEN_STYLE].setMenuIcon(R.drawable.ic_shape_pencil)}"
    layout="@layout/sub_menu_item"/>
```

Library

- ✦ LibraryItem —> UI (data binding)
- ✦ LibraryViewInfo -> LibraryModel