

Giới thiệu về cấu trúc code và thư mục trong dự án

1. Using **BaseActivity.java** vs **BaseActivityDefaultCallBack.java** :
- Load Framgnet Ta sẽ sử dụng các phương thức trong **ISupport.java** để xử lý các nghiệp vụ với Framgnet có thể vào trong **SupportActivity.java** để tìm hiểu cách sử dụng

○ Activity khi extends **BaseActivity** or **BaseActivityDefaultCallBack** Thì phương thức **findViewById** sẽ được thay thế bằng **bindView**

○ Sự khác biệt **BaseActivity** vs **BaseActivityDefaultCallBack**

BaseActivity	BaseActivityDefaultCallBack
implements BaseView.java	implements RetrofitCallback.DefaultViewCallBack.java
Ở đây dùng cho trường hợp muốn cusotom lại sử kiện callBack từ Presenter quay lại View	Ở đây dùng cho giá trị callBack default . khi 1 Activity extends sẽ Override Phương thức showContent Trả về giá trị E extends ServerResponse
@Deprecated onBackPressed	@Deprecated onBackPressed
new function onBackPressedSupport	new function onBackPressedSupport

1. Using **BaseFragment.java** vs **BaseFragmentDefaultCallBack** :
- Load Child Framgnet Ta sẽ sử dụng các phương thức trong **ISupportFragment.java** để xử lý các nghiệp vụ với Child Framgnet có thể vào trong **SupportFragment.java** để tìm hiểu cách sử dụng

○ Fragment khi extends **BaseFragment** or **BaseFragmentDefaultCallBack** Thì phương thức **findViewById** sẽ được thay thế bằng **bindView**

○ **BaseFragment<T extends BasePresenter>** vs **BaseFragmentDefaultCallBack<T extends BasePresenter>** ở đây giá trị **T** Là 1 **Presenter extends BasePresenter** được truyền vào

○ Khi Fragment khi extends Sẽ tự động **@Override** Các phương thức:

Funtion	Document
getLayoutId()	return 1 id layout resources (Đây sẽ là phương thức set layout cho Framgnet)
initInject()	các Framgnet sẽ phải khai báo getFragmentComponent().inject(this) Để đăng ký Fragment hiện tại với Dagger và dùng để khởi tạo các view
onViewReady()	Sau khi khởi tạo xong view sẽ gọi tới hàm này

- Sự khác biệt **BaseActivity** vs **BaseActivityDefaultCallBack** :

BaseFragment	BaseFragmentDefaultCallBack
implements BaseView.java	implements RetrofitCallback.DefaultViewCallBack.java
Ở đây dùng cho trường hợp muốn cusotom lại sử kiện callBack từ Presenter quay lại View	Ở đây dùng cho giá trị callBack default . khi 1 Activity extends sẽ Override Phương thức showContent Trả về giá trị E extends ServerResponse
@Deprecated onBackPressed	@Deprecated onBackPressed
new function onBackPressedSupport	new function onBackPressedSupport

1. Sử dụng RetrofitHelper.java :
 - Sử dụng RetrofitHelper.java Trong Presenter
 - Ta Sử dụng @Inject Để khởi tạo
 - Vidu: java
 - private RetrofitHelper mRetrofitHelper;
 - @Inject
 - public TestPresenter(RetrofitHelper mRetrofitHelper) {
 - this.mRetrofitHelper = mRetrofitHelper;
 - }
 -
1. Sử dụng RxPresenter.java :
 - RxPresenter được sử dụng với mô hình MVP Nó sẽ làm nhiệm vụ xử lý logic và giao tiếp giữa view và model
 - Giới thiệu về các hàm vs cách sử dụng:

Funtion	Document
Presenter	nó là 1 interface Dùng để giao tiếp với model khi được view gọi
View	nó là 1 interface Dùng để giao tiếp với View khi Model trả lại kết quả theo phương thức CallBack
OnRetrofitCallback	Được sử dụng khi Framgnet or Activity extends BaseActivityDefaultCallBack or BaseFragmentDefaultCallBack vs View extends từ RetrofitCallback.DefaultViewCallBack Cách sử dụngbodyCall.enqueue(new OnRetrofitCallback<TestBean>().onRetrofitCallback(view))

1. Sử dụng RequestService.java :
 - Ở đây sẽ khai báo các request sử dụng retrofit2.http

1. Sử dụng **BaseAdapter.java** :
- Khi extends **BaseAdapter** Sẽ có 2 giá trị truyền vào: **VH extends BaseAdapterViewHolder**, **E extends BaseBean**

○ Các component trong Adapter

Funtion	Document
datas	Đây là List<E> E được khai báo khi extends BaseAdapter datas Được sử dụng xuyên suốt adapter như 1 data gốc.
setData(List<E> datas)	Phương thức này dùng để chuyển vào 1 list<E> E phải giống với giá trị được chuyển vào khi extends BaseAdapter Khi chuyển vào đây giá trị datas thì toàn bộ data cũ trong datas sẽ bị clear
appendData(List<E> datas)	Phương thức này dùng để chuyển vào 1 list<E> E phải giống với giá trị được chuyển vào khi extends BaseAdapter Khi chuyển vào đây giá trị datas sẽ được add thêm
appendData(E data)	Phương thức này dùng để chuyển vào 1 E E phải giống với giá trị được chuyển vào khi extends BaseAdapter
notifyData()	Sẽ không dùng cách notifile default của apdater khi muốn notifyData sẽ sử dụng phương thức này
onViewEmpty()	return 1 ids_layout khi không có data
onViewError()	return 1 ids_layout khi gặp 1 Exception
onViewReady(ViewHolder holder, int position)	Dùng khi view adapter được khởi tạo thành công
class OnItemClickListener	Đây sẽ thay cho ItemClick thông thường khi muốn tạo 1 sự kiện click có callBack ra ngoài view ta sẽ sử dụng nó Giá trị cần chuyển vào OnItemClickListener(E item, int position) E phải giống với giá trị được chuyển vào khi extends BaseAdapter , position vị trí item click hiện tại

1. Sử dụng **BaseAdapterCallback.java** :
- Trong Adapter khi muốn sử dụng Callback đều phải extend lại **BaseAdapterCallback**

○ Các component trong Callback

Funtion	Document
onItemClick(E item, int position, View view)	Trả về 2 giá trị item Object của item được click, position vị trí item click

<code>onEmptyData(boolean isEmpty)</code>	Trả về giá trị isEmpty Check nếu isEmpty == true thì data rỗng khi này ngoài view sẽ xử lý , thường data rỗng khi mới bắt đầu load data
<code>onAdapterError(String error)</code>	Trả về giá trị error Trả về các Exception trong adapter