

ALUMNET

# Fields Naming

```
public class MyClass {  
    public static final int SOME_CONSTANT = 42;  
    public int publicField;  
    private static MyClass sSingleton;  
    int mPackagePrivate;  
    private int mPrivate;  
    protected int mProtected;  
}
```

# Fields Naming (Previous)

```
private static Toolbar feedToolbar;  
private SlidingUpPanelLayout slidingUpPanelLayout;  
private SwipeRefreshLayout swipeRefreshLayout;  
private RecyclerView feedList;  
private LinearLayoutManager mLayoutManager;  
private TextView noFeed;  
private TextView selDesAllTags;
```

# Fields Naming (Current)

```
private static FeedFragment sFragmentInstance = null;  
private MainView.OnToolbarCallback mToolbarCallback;  
private FragmentFeedBinding mFeedBinding;  
private FeedPresenter mFeedPresenter;  
private Context mContext;  
private LayoutInflater mLayoutInflater;  
private FeedViewAdapter mFeedAdapter;  
private ArrayList<Feed> mBusinessFeedList;  
private ArrayList<Tag> mBusinessTagList;  
private boolean[] mSelectedTags;
```

# View Binding (Previous)

```
private static Toolbar feedToolbar;  
private SlidingUpPanelLayout slidingUpPanelLayout;  
private SwipeRefreshLayout swipeRefreshLayout;  
private RecyclerView feedList;  
private LinearLayoutManager mLayoutManager;  
private TextView noFeed;  
private TextView selDesAllTags;
```

# View Binding (Previous) Cont.

@Override

```
public View onCreateView(LayoutInflater inflater, final ViewGroup container,
                        Bundle savedInstanceState) {
    View root = inflater.inflate(R.layout.fragment_feed, container, false);
    filterSlidingLayout = (RelativeLayout) root.findViewById(R.id.filter_sliding_layout);
    feedToolBar = (ToolBar) root.findViewById(R.id.feed_toolbar);
    slidingUpPanelLayout = (SlidingUpPanelLayout) root.findViewById(R.id.sliding_layout);
    selDesAllTags = (TextView) root.findViewById(R.id.select_deselect_tags);
    final ChipCloud chipCloud = (ChipCloud) root.findViewById(R.id.chip_cloud);
    noFeed = (TextView) root.findViewById(R.id.feed_not_found);
    swipeRefreshLayout = (SwipeRefreshLayout) root.findViewById(R.id.swipe_refresh_layout);
    feedList = (RecyclerView) root.findViewById(R.id.feed_recycler_view);
    loadingBackground = (RelativeLayout) root.findViewById(R.id.loading_background);
}
```

# View Binding (Current)

```
private FragmentFeedBinding mFeedBinding;
```

```
@Override
```

```
public View onCreateView(LayoutInflater inflater, final ViewGroup container,  
                          Bundle savedInstanceState) {  
    mFeedBinding = DataBindingUtil  
        .inflate(mLayoutInflater, R.layout.fragment_feed, container, false);
```

# Widget IDs Naming Convention (Previous)

```
<android.support.v7.widget.RecyclerView  
    android:id="@+id/feed_recycler_view"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:overScrollMode="never"  
    android:visibility="gone" />
```



# Widget IDs Naming Convention (Current)

IDS

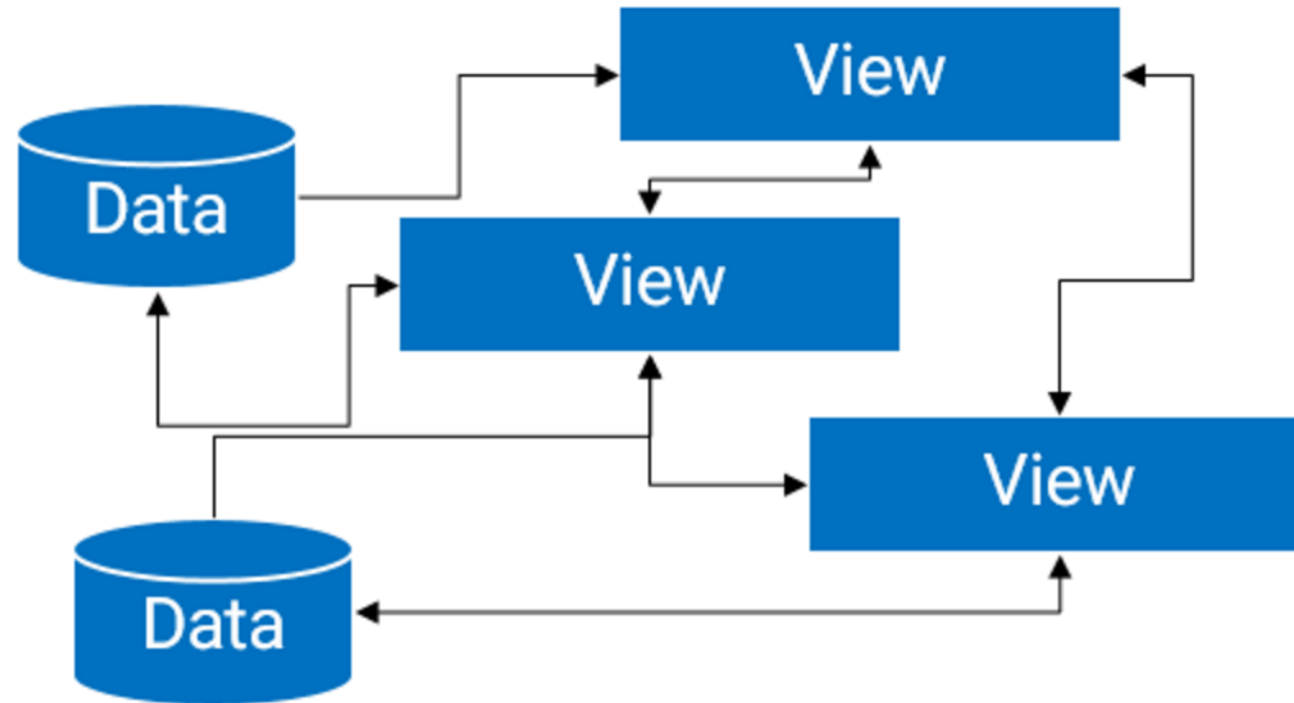
<WHAT>\_<WHERE>\_<DESCRIPTION>

<WHAT> is name of Android/Custom view class

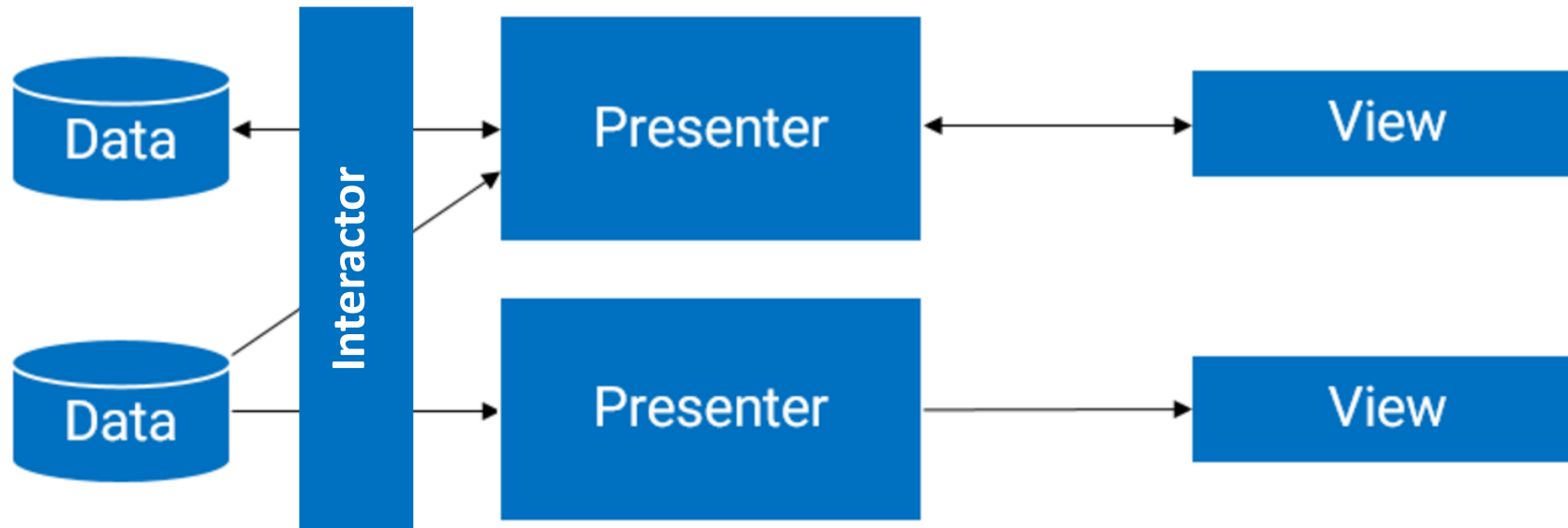
e.g. linearlayout\_main\_fragmentcontainer

```
<android.support.v7.widget.RecyclerView
    android:id="@+id/rv_feed_business_feed"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:overScrollMode="never"
    android:visibility="gone" />
```



# Previous Version
















# Current Version (MVP)



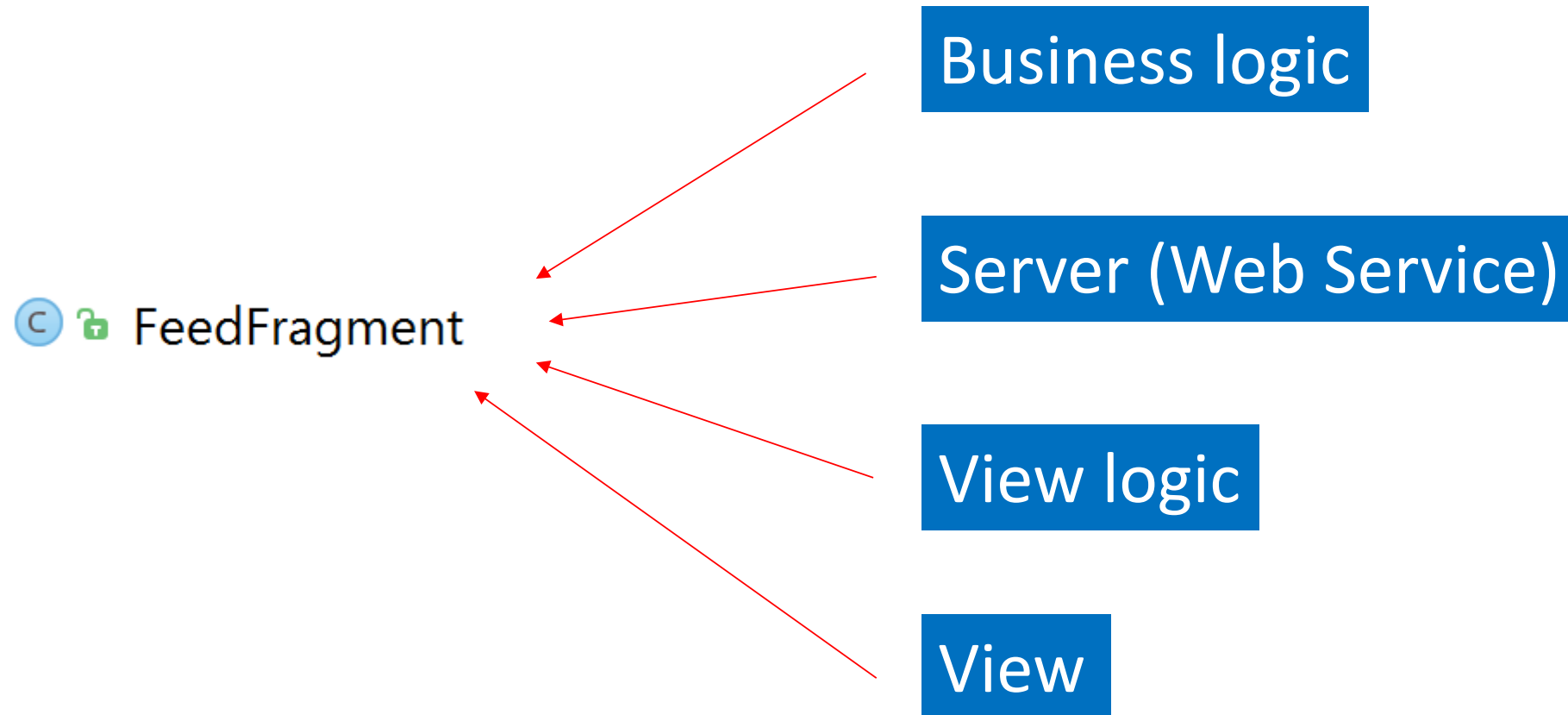
# Example (Business Feed)

  FeedFragment

▼  feed

-   FeedFragment
-   FeedInteractor
-   FeedInteractorImpl
-   FeedPresenter
-   FeedPresenterImpl
-   FeedView

# Example (Business Feed)



# Example (Business Feed)

▼ feed

Ⓒ 🔒 FeedFragment

Ⓘ 🔒 FeedInteractor

Ⓒ 🔒 FeedInteractorImpl

Ⓘ 🔒 FeedPresenter

Ⓒ 🔒 FeedPresenterImpl

Ⓘ 🔒 FeedView

View

Business logic

Server (Web Service)

View logic

