**LIBRARIES**

<https://github.com/amitshekhariitbhu/awesome-android-complete-reference#performance-and-optimization>

<https://github.com/futurice/android-best-practices>

<https://github.com/github/gitignore/blob/master/Android.gitignore>

Bug fixes

**11/30/2015**

1. Lower case string search for nric does not work

Resolution:

**UsersORM.java**

**public static** ArrayList<UsersORM> searchAllByName(String name, String location\_uuid) {  
 Log.*d*(**"debug\_tag"**, **"name"** + Encryption.*Encrypt*(name));  
 **return new** Select().from(UsersORM.**class**)  
 .where(**"(name = ? or name LIKE ? or "** +  
 **"name LIKE ? or "** +  
 **"name LIKE ? or ic = ? or ic = ? ) and location\_uuid = ? and "** +  
 **"deleted\_at IS NULL and is\_terminated = 0"**,  
 **new** String(name),  
 **new** String(**"%"** + name),  
 **new** String(**"%"** + name + **"%"**),  
 **new** String(name + **"%"**),  
 Encryption.*Encrypt*(name.toUpperCase()),  
 Encryption.*Encrypt*(name.toLowerCase()),  
 location\_uuid).execute();  
}

**Future fixes:**

1. Kiosk mode

<http://stackoverflow.com/questions/19920052/disable-the-notification-panel-from-being-pulled-down>

<http://stackoverflow.com/questions/25284233/prevent-status-bar-for-appearing-android-modified?answertab=active#tab-top>

<http://stackoverflow.com/questions/17549478/how-to-disable-home-and-other-system-buttons-in-android>

<http://stackoverflow.com/questions/4548947/how-to-prevent-custom-home-launcher-app-restart-activity#comment4987787_4549337>

Disable notification pull down

- you can call the method on onCreate()

*/\*\*  
 \* Created by herdmacbook1 on 30/11/15.  
 \*  
 \** ***@see*** *for Singleton pattern https://gist.github.com/Akayh/5566992  
 \*/***public class** NoNotificationBarPullDownLayout **extends** ViewGroup {  
 **public static final** String ***TAG*** = **"NoNotificationBarPullDownLayout"**;  
 **public static** NoNotificationBarPullDownLayout *mInstance*;  
 **public** NoNotificationBarPullDownLayout(Context context) {  
 **super**(context);  
 }  
  
 **public static** NoNotificationBarPullDownLayout getInstance(Context applicationContext){  
 **if**(*mInstance* == **null**) {  
 *mInstance* = **new** NoNotificationBarPullDownLayout(applicationContext);  
 }  
 **return** *mInstance*;  
 }  
  
 @Override  
 **protected void** onLayout(**boolean** changed, **int** l, **int** t, **int** r, **int** b) {  
 }  
  
 @Override  
 **public boolean** onInterceptTouchEvent(MotionEvent ev) {  
 Log.*d*(***TAG***, **"onInterceptTouchEvent() notification pull down \*\*\*\*\*\*\*\*\*\*Intercepted"**);  
 **return true**;  
 }  
}

To disable notification pull down on onResume() and enable again on onPause():

@Override  
**protected void** onResume() {  
 *disableNotificationBarPullDown* = NoNotificationBarPullDownLayout.*getInstance*(getApplicationContext());  
 disableNotificationBarPullDown(*disableNotificationBarPullDown*);  
 **super**.onResume();  
}

@Override  
**protected void** onPause() {  
 enableNotificationBarPullDown(getDisableNotificationBarPullDown());  
 *//disableNotificationBarPullDown(getDisableNotificationBarPullDown());* **super**.onPause();  
}

**public** NoNotificationBarPullDownLayout disableNotificationBarPullDown(NoNotificationBarPullDownLayout disableNotificationBarPullDown){  
 **try** {  
 **windowManager**.addView(disableNotificationBarPullDown, **localLayoutParams**);  
 }**catch** (Exception e){  
 Log.*e*(***TAG***, **"disableNotificationBarPullDown already added to windowmanager"**);  
 }  
 **return** disableNotificationBarPullDown;  
}

**public void** enableNotificationBarPullDown(NoNotificationBarPullDownLayout noNotificationBarPullDownLayout){  
 **try** {  
 **windowManager**.removeView(noNotificationBarPullDownLayout);  
 }**catch** (Exception e){  
 Log.*e*(***TAG***, **"disableNotificationBarPullDown already removed from windowmanager"**);  
 }  
}

**private void** initializePullDownLocalLayoutParams(){  
 **localLayoutParams** = **new** WindowManager.LayoutParams();  
 **localLayoutParams**.**type** = WindowManager.LayoutParams.***TYPE\_SYSTEM\_ERROR***;  
 **localLayoutParams**.**gravity** = Gravity.***TOP***;  
 **localLayoutParams**.**flags** = WindowManager.LayoutParams.***FLAG\_NOT\_FOCUSABLE***|  
  
 *// this is to enable the notification to recieve touch events* WindowManager.LayoutParams.***FLAG\_NOT\_TOUCH\_MODAL*** |  
  
 *// Draws over status bar* WindowManager.LayoutParams.***FLAG\_LAYOUT\_IN\_SCREEN***;  
  
 **localLayoutParams**.**width** = WindowManager.LayoutParams.***MATCH\_PARENT***;  
 **localLayoutParams**.**height** = (**int**) (50 \* getApplicationContext().getResources()  
 .getDisplayMetrics().**scaledDensity**);  
 **localLayoutParams**.**format** = PixelFormat.***TRANSPARENT***;  
}

@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 **windowManager** = ((WindowManager) getApplicationContext()  
 .getSystemService(Context.***WINDOW\_SERVICE***));  
 initializePullDownLocalLayoutParams();  
  
  
  
 *//disableHomeButton();*}

2. Custom font:

-Remove the font creation from asset in these custom classes

HelveticaNeueRegular

HelveticaNeueMedium

- Make the typeface a variable

<http://stackoverflow.com/questions/13077525/slow-listview-when-set-custom-font>

(Answer by Scorchio and ASP)

<http://stackoverflow.com/questions/13077525/slow-listview-when-set-custom-font>

(Answer by Moises Olmedo)

<http://stackoverflow.com/questions/4576441/custom-font-in-android-listview>

3. Typeface/custom font on listview

<http://stackoverflow.com/questions/15338016/performance-issue-on-custom-font-textview>

4. Loading of images

<https://www.redfin.com/devblog/2015/10/using-fresco-to-load-images-efficiently-on-android.html>

5. Launcher

Removing my launcher app from being the default launcher then showing the launcher picker dialog

[*http://stackoverflow.com/questions/27991656/how-to-set-default-app-launcher-programmatically*](http://stackoverflow.com/questions/27991656/how-to-set-default-app-launcher-programmatically)[*http://stackoverflow.com/questions/28537576/launcher-app-chooser-dialog-is-not-being-displayed*](http://stackoverflow.com/questions/28537576/launcher-app-chooser-dialog-is-not-being-displayed)[*http://stackoverflow.com/questions/23110047/android-choose-default-launcher-programatically*](http://stackoverflow.com/questions/23110047/android-choose-default-launcher-programatically)

[*http://stackoverflow.com/questions/12594192/remove-activity-as-default-launcher*](http://stackoverflow.com/questions/12594192/remove-activity-as-default-launcher)

*//remove app from being the default launcher*

getPackageManager().clearPackagePreferredActivities(getPackageName());  
  
*//show the launcher picker*Intent intent = **new** Intent(Intent.***ACTION\_MAIN***);  
intent.addCategory(Intent.***CATEGORY\_HOME***);  
intent.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
startActivity(intent);

6. Show a dialog after 7 days of inactivity

7. Override home button

<http://stackoverflow.com/questions/4783960/call-method-when-home-button-pressed-on-android>

<http://stackoverflow.com/questions/8881951/detect-home-button-press-in-android/8883447#8883447>

<http://nisha113a5.blogspot.sg/2012/01/intercept-home-key-android.html> (no longer work on 4.0+)

<http://www.andreas-schrade.de/2015/02/16/android-tutorial-how-to-create-a-kiosk-mode-in-android/> (GOOD)

<https://groups.google.com/forum/#!topic/android-developers/trRI99-HszQ> (it can’t be done)

<https://groups.google.com/forum/embed/#!topic/android-developers/_Ex4GOfp4r4> (it can’t be done)

<http://uperones-blog.readthedocs.org/en/latest/_posts/2014-12-05-android-listen-home-button-click/>

<http://stackoverflow.com/questions/9972166/how-to-listener-homebutton-in-android> (it can’t be done)

<http://uperones-blog.readthedocs.org/en/latest/_posts/2014-12-05-android-listen-home-button-click/> (Listen to home button)

8. Triple tap on Home

- works only on devices with software navigation buttons

- not working on devices with hardware home buttons

Req: App is a launcher app

<**activity  
 android:name=".MainActivity"  
 android:launchMode="singleTask"  
 android:label="@string/app\_name"** >  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 *<!-- The following two intent-filters are the key to set homescreen -->* <**category android:name="android.intent.category.HOME"** />  
 <**category android:name="android.intent.category.DEFAULT"** />  
  
 </**intent-filter**>  
</**activity**>

Add these in your launcher activity:

**int homeClickCounter** = 0;  
**public final int HOME\_CLICK\_COUNTER** = 800;

@Override  
**protected void** onResume() {  
 **super**.onResume();  
 **homeClickCounter**++;  
 resetHomeTripleTapTimer(**HOME\_CLICK\_COUNTER**);  
}

@Override  
**protected void** onUserLeaveHint() {  
 **super**.onUserLeaveHint();  
 **homeClickCounter** = 0;  
  
}

**private** Handler **homeTripleTap** = **new** Handler() {  
 **public void** handleMessage(Message msg) {  
  
 }  
};  
**private** Runnable **tripleTapCallback** = **new** Runnable() {  
 @Override  
 **public void** run() {  
 Log.*d*(***TAG***, ***TAG***+**" tripleTapCallback run()"**);  
 *// Perform any required operation on disconnect* **if** (**homeClickCounter**==3){  
 **homeClickCounter** = 0;  
 Log.*d*(***TAG***, ***TAG***+**" tripleTapCallback run() triple tap"**);  
 }**else**{  
 **homeClickCounter** = 0;  
 Log.*d*(***TAG***, ***TAG***+**" tripleTapCallback run() no triple tap"**);  
 }  
 }  
};  
  
**public void** resetHomeTripleTapTimer(**long** timeout) {  
 **homeTripleTap**.removeCallbacks(**tripleTapCallback**);  
 **homeTripleTap**.postDelayed(**tripleTapCallback**, timeout);  
}

**ANDROID ALARM MANAGER NOT WORKING, NOT REPEATING**

<http://stackoverflow.com/questions/33053651/android-setrepeating-and-setinexactrepeating-do-not-fire>

**SET BACKGROUND OF ACTIVITY TO WHITE**

<http://stackoverflow.com/questions/4761686/how-to-set-background-color-of-activity-to-white-programmatically>

setContentView(R.layout.main); // Now get a handle to any View contained // within the main layout you are using View someView = findViewById(R.id.randomViewInMainLayout); // Find the root view View root = someView.getRootView() // Set the color root.setBackgroundColor(getResources().getColor(android.R.color.red));

**Robolectric Setup for Unit Testing**

1. On AndroidManifest.xml add the following

<**meta-data  
 android:name="AA\_DB\_NAME"  
 android:value="nameOfDatabaseHere.db"** />  
<**meta-data  
 android:name="AA\_DB\_VERSION"  
 android:value="1"** />  
<**meta-data android:name="AA\_MODELS"  
 android:value="com.example.models.User, com.example.models.Friends "** />

2. In the <application> tag add your Application Class there

3. On your application class,

- extend from **android.app.Application**

**-** create a method initializeDb(), initialize there what you have initialized in the AndroidManifest.xml like so

**private void** initializeDb(){  
 com.activeandroid.Configuration.Builder configurationBuilder = **new** com.activeandroid.Configuration.Builder(**this**);  
 configurationBuilder.setDatabaseName(**"nameOfDatabaseHere.db "**);  
 configurationBuilder.setDatabaseVersion(1);  
 configurationBuilder.addModelClasses(User.**class**);  
 configurationBuilder.addModelClasses(Friends.**class**);  
  
 ActiveAndroid.*initialize*(configurationBuilder.create());  
}

4. On your application class onCreate() call initializeDb(),

5. For unit testing, on build.gradle add the following

dependencies{

…

testCompile **"org.robolectric:robolectric:3.0"**

}

6. Also create another ApplicationTest class that extends the **com.activeandroid.app.Application,**  it is just the same as the Application class created from step 3

**public class** MyApplicationTest **extends** com.activeandroid.app.Application {  
 @Override  
 **public void** onCreate() {  
 **super**.onCreate();initializeDB();  
 }  
  
 **private void** initializeDB(){  
 com.activeandroid.Configuration.Builder configurationBuilder = **new** com.activeandroid.Configuration.Builder(**this**);  
 configurationBuilder.setDatabaseName(**"nameOfDatabaseHere.db "**);  
 configurationBuilder.setDatabaseVersion(1);  
 configurationBuilder.addModelClasses(User.**class**);  
 configurationBuilder.addModelClasses(Friends.**class**);  
 ActiveAndroid.*initialize*(configurationBuilder.create()); }  
}

7. Create your unit test class like so (UserTest.java)

@RunWith(RobolectricGradleTestRunner.**class**)  
@Config(

constants = BuildConfig.**class**,

sdk = 18,

application = MyApplicationTest.**class**,

shadows = {

ShadowEnvironment.**class**,

ShadowMultiDex.**class**}

)

**public class** UserTest {

@Before  
**public void** init(){

}

@Test  
**public void** addUserTest(){

}

}

**Notes:**

* the activeandroid jar library is built by me by downloading the activeandroid master branch in github (IMPORTANT: the .jar files that are available to download on ActiveAndroid github is not updated, you need to do ant build for the jar)

- the downloadable .jar files for ActiveAndroid does not work on Lollipop

see: <https://github.com/pardom/ActiveAndroid>

* I added ShadowEnvironment.class to make sure the file directories like getFilesDirectory() method gets shadowed
* I added ShadowMultidex.class because I have the below code in build.gradle

android{

…

multiDexEnabled **true**

**…**

**}**

Ref: <http://stackoverflow.com/questions/23907037/how-to-disable-scanning-for-model-in-activeandroid/>

<http://stackoverflow.com/questions/26933386/activeandroid-crashes-on-lollipop-noclassdeffounderror>

<https://github.com/pardom/ActiveAndroid/issues/291>

<https://github.com/pardom/ActiveAndroid/issues/109>

<https://github.com/pardom/ActiveAndroid/issues/299>

**ACTIVEANDROID**

**Save database file for debugging**

**public static void** copyAppDbToDownloadFolder(Context mContext) **throws** IOException {  
 File backupDB = **new** File(Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_DOWNLOADS*), **"herdhrBackup.db"**); *// for example "my\_data\_backup.db", "/storage/emulated/0/Download/herdhrBackup.db"* File currentDB = mContext.getApplicationContext().getDatabasePath(**"herdhr.db"**); *//databaseName=your current application database name, for example "my\_data.db"* **if** (currentDB.exists()) {  
 FileChannel src = **new** FileInputStream(currentDB).getChannel();  
 FileChannel dst = **new** FileOutputStream(backupDB).getChannel();  
 dst.transferFrom(src, 0, src.size());  
 src.close();  
 dst.close();  
 *// backupDB = {java.io.File@5502} "/storage/emulated/0/Download/herdhrBackup.db"  
 // currentDB = {java.io.File@5503} "/data/data/com.herdhr.operationapp\_dev/databases/herdhr.db"* }  
}

**Logging -**

1. <http://stackoverflow.com/questions/24475437/is-there-a-way-to-get-android-history-system-log>

2. <http://android.stackexchange.com/questions/14430/how-can-i-view-and-examine-the-android-log>

3. <http://stackoverflow.com/questions/4428643/examine-logs-from-the-past-with-logcat>

TO READ:

<http://forum.xda-developers.com/showthread.php?t=1726238>

<https://wiki.cyanogenmod.org/w/Doc:_debugging_with_logcat>

**UI DESIGN CHEATSHEET**

* <http://petrnohejl.github.io/Android-Cheatsheet-For-Graphic-Designers/>
* helps you find the equivalent of one dimension to ldpi, mdpi, hdpi, xhdpi and xxhdpi like
  + in mdpi – buttonHeight = “20dp”, the app will give you the values for ldpi, hdpi, xhdpi, xxhdpi

**SAMSUNG GALAXY TAB S WIFI ISSUE**

* the whole device can not access the internet, even browsers
* <https://www.ifixit.com/Wiki/Samsung_Galaxy_Tab_S_8.4_Troubleshooting>
* <http://thedroidguy.com/2015/01/samsung-galaxy-tab-s-8-4-problems-errors-glitches-solutions-part-3-101955>

**FONT SIZES**

* <https://www.google.com/design/spec/style/typography.html#typography-other-typographic-guidelines>

**CHEATSHEET**

* <https://drive.google.com/file/d/0B5XIkMkayHgRMVljUVIyZzNmQUU/view?pref=2&pli=1>

**REMOVE/DISABLE HIGHLIGHT ON LISTVIEW CLICK**

* <http://stackoverflow.com/questions/2907335/android-disabling-highlight-on-listview-click/12242564#12242564>
  + make sure to have this in your listview layout:

<ListView

android:listSelector="@android:color/transparent" android:cacheColorHint="@android:color/transparent"

/>

**PREVENT RXJAVA/RXANDROID FROM LEAKING**

* <http://www.philosophicalhacker.com/2015/03/24/how-to-keep-your-rxjava-subscribers-from-leaking/>

**getExternalFilesDir() returns null**

* <http://stackoverflow.com/questions/10114801/getexternalfilesdir-returns-null-not-a-permissions-issue> (sometimes you have to restart the device)

**Delay server response time**

* <http://fake-response.appspot.com/?sleep=10>

**TEST IF A FRAGMENT IS VISIBLE TO USER**

**- useful when populating a fragment when in a viewpager**

* [**http://stackoverflow.com/questions/9323279/how-to-test-if-a-fragment-view-is-visible-to-the-user**](http://stackoverflow.com/questions/9323279/how-to-test-if-a-fragment-view-is-visible-to-the-user)

**CANDIDATE LIBRARIES TO USE**

[**https://github.com/codepath/android\_guides/wiki/Must-Have-Libraries**](https://github.com/codepath/android_guides/wiki/Must-Have-Libraries)

**UI**

* [**https://github.com/81813780/AVLoadingIndicatorView**](https://github.com/81813780/AVLoadingIndicatorView)
* [**https://github.com/JoanZapata/android-iconify**](https://github.com/JoanZapata/android-iconify)
* [**https://github.com/Pkmmte/CircularImageView**](https://github.com/Pkmmte/CircularImageView)
* [**https://github.com/navasmdc/MaterialDesignLibrary**](https://github.com/navasmdc/MaterialDesignLibrary)
* [**https://fortawesome.github.io/Font-Awesome/icons/**](https://fortawesome.github.io/Font-Awesome/icons/)
* [**https://github.com/JakeWharton/ViewPagerIndicator**](https://github.com/JakeWharton/ViewPagerIndicator)
* [**https://github.com/p-v/FlexibleCalendar**](https://github.com/p-v/FlexibleCalendar)
* compile **'com.mikhaellopez:circularimageview:3.0.2'**
* [**https://github.com/emilsjolander/StickyListHeaders**](https://github.com/emilsjolander/StickyListHeaders)
* [**https://github.com/Pixplicity/EasyPreferences**](https://github.com/Pixplicity/EasyPreferences)
* [**http://loopj.com/android-async-http/**](http://loopj.com/android-async-http/)
* [**https://github.com/chrisjenx/Calligraphy**](https://github.com/chrisjenx/Calligraphy)
* [**https://github.com/orhanobut/logger**](https://github.com/orhanobut/logger)

**ISSUES**

* [**http://stackoverflow.com/questions/33841363/how-to-make-a-page-indicator-for-horizontal-recyclerview**](http://stackoverflow.com/questions/33841363/how-to-make-a-page-indicator-for-horizontal-recyclerview)

**SETUP of VIEWPAGERINDICATOR by JakeWharton**

**App’s build.gradle**repositories {  
 mavenCentral()  
 **maven { url "https://jitpack.io" }**  
}  
android {}

dependencies {

compile **'com.github.JakeWharton:ViewPagerIndicator:2.4.1@aar'**

}

**RETROFIT SETUP**

**Build.gradle (app)**

compile **'com.squareup.retrofit2:retrofit:2.0.0-beta4'**compile **'com.squareup.okhttp3:okhttp-urlconnection:3.2.0'**compile **'com.squareup.okhttp3:okhttp:3.2.0'**

compile **'com.squareup.okhttp3:logging-interceptor:3.2.0' //for retrofit logging**compile **'com.squareup.retrofit2:adapter-rxjava:2.0.0-beta4' //for rxjava**compile **'com.squareup.retrofit2:converter-gson:2.0.0-beta4' //for the converter**

**API MANAGER CLASS**

**public class** ApiManager {  
 **public static final** String ***BASE\_URL*** = **""**;  
 Retrofit **retrofit**;  
 ApiInterface **api**;  
  
 **public** ApiManager() {  
 **retrofit** = **new** Retrofit.Builder()  
 .baseUrl(***BASE\_URL***)  
 .addConverterFactory(GsonConverterFactory.*create*())

.addCallAdapterFactory(RxJavaCallAdapterFactory.*create*()) //very important for RXJAVA  
 .build();  
 **api** = **retrofit**.create(ApiInterface.**class**);  
 }  
}

**SAMPLE POST API:**

**public interface** ApiInterface {

**…**

@Headers({  
 **"X-XSRF-TOKEN: 'Ej3NQ4mLspTosE2o9Aili7gp6Q7p5Yn1hfQi6E34'"**,  
 **"X-HOSTNAME: 'demo'"**})  
@FormUrlEncoded  
@POST(***OAUTH\_URL***)  
Observable<Authorize> authorizeObservable(  
 @Field(**"host"**) String host,  
 @Field(**"username"**) String username,  
 @Field(**"password"**) String password,  
 @Field(**"client\_id"**) String client1id,  
 @Field(**"client\_secret"**) String clientSecret,  
 @Field(**"grant\_type"**) String grantType,  
 @Field(**"gcm"**) String gcmid  
);

**…**

}

**GETTING THE HEADER INFORMATION OF THE RESPONSE**

* [**http://stackoverflow.com/questions/26851459/getting-header-information-with-rxjava-and-retrofit**](http://stackoverflow.com/questions/26851459/getting-header-information-with-rxjava-and-retrofit)
* [**http://stackoverflow.com/questions/33774940/get-response-status-code-using-retrofit-2-0-and-rxjava**](http://stackoverflow.com/questions/33774940/get-response-status-code-using-retrofit-2-0-and-rxjava)

**sample:**

@Headers({  
 **"X-XSRF-TOKEN: "**+***HEADER\_XSRFTOKEN***,  
 **"X-HOSTNAME: "**+***HEADER\_HOSTNAME***})  
@FormUrlEncoded  
@POST(***OAUTH\_URL***)  
Observable<**Response<**Authorize**>**> authorizeResponseObservable(

//Authorize is your pojo/class that represents the return json String  
 @Field(**"host"**) String host,  
 @Field(**"username"**) String username,  
 @Field(**"password"**) String password,  
 @Field(**"client\_id"**) String client1id,  
 @Field(**"client\_secret"**) String clientSecret,  
 @Field(**"grant\_type"**) String grantType,  
 @Field(**"gcm"**) String gcmid  
);

**to use it:**

**to get the response:**

ApiManager apiManager = **new** ApiManager();  
  
apiManager.authorizeResponseObservable(**"demo"**, **"admin@admin.com"**, **"herd4008hr"**, **"mobile-android"**, **"28001a630ec3b1d53eaac5463fcc4563"**, **"password"**, **"3424"**)  
 .subscribe(**new** Subscriber<Response<Authorize>>() {  
 @Override  
 **public void** onCompleted() {  
  
 }  
  
 @Override  
 **public void** onError(Throwable e) {  
  
 }  
  
 @Override  
 **public void** onNext(Response<Authorize> authorizeResponse) {  
 okhttp3.Response raw = authorizeResponse.raw(); **//how to get the Header Response**  
Authorize body = authorizeResponse.body(); //**how to get the converted object**

}  
 });

**RETROFIT ADDING TIMEOUT CONFIGURATION**

**1. create the client**

okhttp3.OkHttpClient **client** = **new** okhttp3.OkHttpClient().newBuilder()  
 .readTimeout(60, TimeUnit.***SECONDS***)  
 .connectTimeout(60, TimeUnit.***SECONDS***)  
 .build();

**2. add the client to the retrofit object**

retrofit = **new** Retrofit.Builder()  
 .baseUrl(ApiInterface.***BASE\_URL***)  
 .client(**client**)  
 .addConverterFactory(GsonConverterFactory.*create*())  
 .addCallAdapterFactory(RxJavaCallAdapterFactory.*create*())  
 .build();

**RETROFIT, PARSE SERVER DATA IF STATUSCODE != 200**

[**https://github.com/square/retrofit/issues/1235**](https://github.com/square/retrofit/issues/1235)

**RETROFIT, ADDING LOGGING TO RETROFIT CALLS**

<http://stackoverflow.com/questions/21886313/how-to-log-request-and-response-body-with-retrofit-android>

On your build.gradle make sure to have

compile **'com.squareup.okhttp3:logging-interceptor:3.2.0'**

then on your instance in creating the retrofit instance

HttpLoggingInterceptor logging = **new** HttpLoggingInterceptor();  
*// set your desired log level*logging.setLevel(HttpLoggingInterceptor.Level.BODY);  
  
  
okhttp3.OkHttpClient client = **new** okhttp3.OkHttpClient().newBuilder()  
 .readTimeout(60, TimeUnit.***SECONDS***)  
 .connectTimeout(60, TimeUnit.***SECONDS***)

**.addInterceptor(logging)**  
 .build();  
**retrofit** = **new** Retrofit.Builder()  
 .baseUrl(ApiInterface.***BASE\_URL***)  
 .client(**client**)  
 .addConverterFactory(GsonConverterFactory.create())  
 .addCallAdapterFactory(RxJavaCallAdapterFactory.create())  
 .build();  
**api** = **retrofit**.create(ApiInterface.**class**);

**RETROFIT/RXJAVA DOWNLOADING A FILE**

**In your interface:**

@GET  
@Streaming  
Observable<ResponseBody> downloadFileObservable(  
 @Url String url, //the url to download  
 @Header(***HEADER\_KEY\_NAME***) String accessToken, //header variables  
 @Header(***HEADER\_KEY\_NAME2***) String hostname //header variables  
);

**Instantiate Retrofit**

**public** Observable<ResponseBody> downloadFileObservable(String accessToken, String host, String url){  
 **return retrofitApi**.downloadFileObservable(url, accessToken, host)  
 .subscribeOn(Schedulers.*io*())  
 .observeOn(AndroidSchedulers.*mainThread*());  
}

**Then download**

**apiManager**.downloadFileObservable(accessToken, host, url)  
 .subscribe(**new** Subscriber<ResponseBody>() {  
 @Override  
 **public void** onCompleted() {  
  
 }  
  
 @Override  
 **public void** onError(Throwable e) {  
 Log.*d*(***TAG***, **"downloadNow2() downloadFileObservable() onError() file not downloaded "**,e);  
  
 }  
  
 @Override  
 **public void** onNext(ResponseBody responseBody) {  
 **savePayslipObservable**(responseBody, month, year)  
 .subscribe(**new** Subscriber<File>() {  
 @Override  
 **public void** onCompleted() {  
  
 }  
  
 @Override  
 **public void** onError(Throwable e) {  
 Log.*d*(***TAG***, **"onError() called with: "** + **"e = ["** + e + **"]"**);  
 Toast.*makeText*(WebViewActivity.**this**, getString(R.string.***label\_payslip\_error\_retrieval***), Toast.***LENGTH\_SHORT***).show();  
 hideLoadingDialog();  
 }  
  
 @Override  
 **public void** onNext(File file) {  
 Log.*d*(***TAG***, **"onNext() called with file "**);  
 **if** (file!=**null** && file.exists()==**true**){  
 Log.*d*(***TAG***, **"onNext() file downloaded"**);  
 *//Toast.makeText(WebViewActivity.this, getString(R.string.label\_download\_success), Toast.LENGTH\_SHORT).show();* hideLoadingDialog();  
 checkFabVisibility(month, year);  
 showPayslipDownloadedSnackbar(file);  
 }**else**{  
 Log.*d*(***TAG***, **"onNext() called with file not downloaded"**);  
 Toast.*makeText*(WebViewActivity.**this**, getString(R.string.***label\_payslip\_error\_retrieval***), Toast.***LENGTH\_SHORT***).show();  
 hideLoadingDialog();  
 }  
 }  
 });  
 }  
 });

**//THE OBSERVABLE TO SAVE THE FILE FROM ResponseBody**

**public** Observable<File> savePayslipObservable(**final** ResponseBody responseBody, **int** month, **int** year){  
 Log.*d*(***TAG***, **"savePayslipObservable() called with: month = ["** + month + **"], year = ["** + year + **"]"**);  
 **return** Observable.*just*(**savePayslip**(responseBody, month, year))  
 .subscribeOn(Schedulers.*computation*())  
 .observeOn(AndroidSchedulers.*mainThread*());  
}

**//SAVING FILE**

*/\*\*  
 \* savePayslip  
 \** ***@param responseBody*** *ResponseBody  
 \** ***@return*** *int File - the downloaded file, otherwise null (if not downloaded)  
 \*/***public** File savePayslip(ResponseBody responseBody, **int** month, **int** year){  
 File retVal = **null**;  
 Log.*d*(***TAG***, **"savePayslip()"**);  
 InputStream inputStream = **null**;  
  
 *//String name = generateFileName(month, year);  
 //Log.d(TAG, "savePayslip() destination file:"+name);  
  
 //MyApplication application = (MyApplication)getApplication();  
 //File file = new File(application.getFileList().get(1), name);* File file = generateFile(month, year); //creates a new file  
 **if** (file.exists())  
 file.delete();  
  
 **try** {  
 inputStream = responseBody.byteStream();  
 FileUtils.*copyInputStreamToFile*(inputStream, file);  
 Log.*d*(***TAG***, **"savePayslip() downloadFileObservable() onNext() file downloaded"**);  
 retVal = file;  
 }**catch** (Exception e){  
 Log.*d*(***TAG***, **"savePayslip() downloadFileObservable() onNext() file not downloaded"**);  
 e.printStackTrace();  
 }  
 **return** retVal;  
}

**SORT A LIST OF OBJECTS Using the ‘name’ attribute**

Class Person{

String name = “”;

}

//the actual sort,

ArrayList<Person> list = **new** ArrayList< Person >();

**if** (list!=**null** && list.size()>1) {  
 Collections.*sort*(list, **new** Comparator< Person >() {  
 **public int** compare(Person o1, Person o2) {  
 **return** o1.**name**.toLowerCase().compareTo(o2.**name**.toLowerCase());  
 }  
 });  
}

//the **list** variable will now be **sorted alphabetically**

**RETROFIT CHANGE BASE URL AT RUNTIME**

<https://futurestud.io/tutorials/retrofit-2-how-to-change-api-base-url-at-runtime-2>

**RXJAVA LIVESEARCH**

* this live search uses SearchView, it will trigger the search functionality when the user has stopped pressing keys only the debounce will determine when the search will perform,
* sample: (debounce = 1000)
  + the user will input text to the SearchView, after the user has stopped pressing keys for 1000ms (the debounce amount) only then the search will trigger

…

@Override  
**public void** onViewCreated(View view, Bundle savedInstanceState) {  
 **super**.onViewCreated(view, savedInstanceState);  
 *//add searchview livesearch* getLiveSearchObservable()  
 .debounce(DirectoryListingListPresenter.***DEBOUNCE***, TimeUnit.***MILLISECONDS***)  
 .observeOn(AndroidSchedulers.*mainThread*())  
 .subscribe(**new** Subscriber<String>() {  
 @Override  
 **public void** onCompleted() {}  
 @Override  
 **public void** onError(Throwable e) {}  
 @Override  
 **public void** onNext(String s) {  
 Toast.*makeText*(getActivity(), **"Search triggered:"**+s, Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
}  
  
**public** Observable<String> getLiveSearchObservable(){  
 **return** Observable.*create*(**new** Observable.OnSubscribe<String>() {  
 @Override  
 **public void** call(**final** Subscriber<? **super** String> subscriber) {  
 **searchViewCompat**.setOnQueryTextListener(**new** SearchView.OnQueryTextListener() {  
 @Override  
 **public boolean** onQueryTextSubmit(String query) {  
 **return false**;  
 }  
 @Override  
 **public boolean** onQueryTextChange(String newText) {  
 subscriber.onNext(newText);  
 **return false**;  
 }  
 });  
 }  
 });  
}

Can also be done using edittexts, the getLiveSearchObservable() method will change as follows

**private** Observable<String> getLiveSearchEditTextObservable(**final** EditText editText){  
 **return** Observable.*create*(**new** Observable.OnSubscribe<String>() {  
 @Override  
 **public void** call(**final** Subscriber<? **super** String> subscriber) {  
 editText.addTextChangedListener(**new** TextWatcher() {  
 @Override  
 **public void** beforeTextChanged(CharSequence s, **int** start, **int** count, **int** after) {}  
 @Override  
 **public void** onTextChanged(CharSequence s, **int** start, **int** before, **int** count) {  
 subscriber.onNext(s.toString());  
 }  
 @Override  
 **public void** afterTextChanged(Editable s) {}  
 });  
 }  
 });  
}

**UNIVERSAL IMAGELOADER IMAGE CACHE**

<http://stackoverflow.com/questions/25968252/how-to-use-universal-image-loader-offline-caching>

<http://stackoverflow.com/questions/15703389/correct-usage-of-universal-image-loader>

**RETROFIT PARAMETERIZED HEADER**

<http://stackoverflow.com/questions/18478258/android-retrofit-parameterized-headers>

**GSON OBJECT ARRAY TO JSON, and VICE VERSA**

[**http://stackoverflow.com/questions/9186806/gson-turn-an-array-of-data-objects-into-json-android**](http://stackoverflow.com/questions/9186806/gson-turn-an-array-of-data-objects-into-json-android)

**JODATIME DATETIME**

**Parse a UTC String**

String time = **"2016-03-10T10:34:55Z"**;  
DateTime parsed = DateTime.*parse*(time).withZone(DateTimeZone.***UTC***);

**Apply timezone to the parsed**

DateTimeZone zone = DateTimeZone.*forID*(timezone);

//timeZone = “Asia/Singapore”  
DateTime zonedDateTime = parsed.withZone(zone);

**Create a formatte with timezone**

DateTimeFormatter fmt = DateTimeFormat.*forPattern*(dateFormat).withZone(zone);

**Create a datetime object with timezone then convert to UTC format**

int day = 1;

int month = 1;

int year = 2016;

**public static final** String ***DATEFORMAT\_UTC*** = **"yyyy-MM-dd'T'HH:mm:ss'Z'"**;

//get the timezone of the location, here we just explicitly set to Asia/Singapore

DateTimeZone timezone = DateTimeZone.*forID*(**"Asia/Singapore"**);

//create the datetiime object

DateTime dateTime = **new** DateTime(year,month,day,0,0,timezone);

//create a formatter with zone set to UTC

DateTimeFormatter fmt = DateTimeFormat.*forPattern*(dateFormat).withZoneUTC();

//get the formatted datetime

String dateStr = fmt.print(dateTime);

**Create a datetime object with utc timezone and format of**

**yyyy-MM-dd'T'HH:mm:ss'Z'**

**final** String DATEFORMAT\_UTC = **"yyyy-MM-dd'T'HH:mm:ss'Z'"**;  
DateTimeFormatter fmt = DateTimeFormat.*forPattern*(DATEFORMAT\_UTC).withZone(DateTimeZone.***UTC***);

DateTime dateTime = DateTime.*now*();  
String dateString = fmt.print(dateTime); //DISPLAYABLE STRING

**Create a datetime object with utc timezone and format of**

**yyyy-MM-dd'T'HH:mm:ss'Z'**

**without the hours, minutes, seconds**

**final** String DATEFORMAT\_UTC = **"yyyy-MM-dd'T'HH:mm:ss'Z'"**;  
DateTimeFormatter fmt = DateTimeFormat.*forPattern*(DATEFORMAT\_UTC).withZone(DateTimeZone.***UTC***);

LocalDate localDate = LocalDate.*now*();  
DateTime now = localDate.toDateTimeAtStartOfDay();  
String dateStr = fmt.print(now); //DISPLAYABLE STRING

**TIME SINCE/AGO Library**

1. <http://stackoverflow.com/questions/13018550/time-since-ago-library-for-android-java>

2. <http://stackoverflow.com/questions/11665404/simplest-way-to-get-local-milliseconds-in-a-time-zone-with-joda-time>

**ex. Given date = “2016-03-10T10:34:55Z”**

**Get the time since from the given date: e.g. 23 days ago**

**1. create a reference to Jan 1, 1970**

**private static final** LocalDateTime ***JAN\_1\_1970*** = **new** LocalDateTime(1970, 1, 1, 0, 0);

**2. create a reference to the timezone of the current location**

DateTimeZone **timezone** = DateTimeZone.*forID*(timezoneStr);

//timezoneStr can be like “Asia/Singapore”

**3. get the current time in millis**

**long** now = System.*currentTimeMillis*();

**4. get the millis value of the parsed Given time** (with reference from Jan 1, 1970)

DateTime parsed = DateTime.*parse*(**“2016-03-10T10:34:55Z”**);  
**long** parsedMs = **new** Duration(***JAN\_1\_1970***.toDateTime(**timezone**), parsed).getMillis();

**5. Use the following code to get the time since/ago**

*/\*\*  
 \** ***@param time*** *long timestamp from Date or DateTime  
 \** ***@param ctx*** *\** ***@param now*** *System.currentTimeMillis()  
 \** ***@return*** *\*/***public static** String getTimeAgo(**long** time, Context ctx, **long** now) {  
 **if** (time < 1000000000000L) {  
 *// if timestamp given in seconds, convert to millis* time \*= 1000;  
 }  
  
 **if** (time > now || time <= 0) {  
 **return null**;  
 }  
**final long** diff = now - time;  
 **if** (diff < ***MINUTE\_MILLIS***) {  
 **return "just now"**;  
 } **else if** (diff < 2 \* ***MINUTE\_MILLIS***) {  
 **return "a minute ago"**;  
 } **else if** (diff < 50 \* ***MINUTE\_MILLIS***) {  
 **return** diff / ***MINUTE\_MILLIS*** + **" minutes ago"**;  
 } **else if** (diff < 90 \* ***MINUTE\_MILLIS***) {  
 **return "an hour ago"**;  
 } **else if** (diff < 24 \* ***HOUR\_MILLIS***) {  
 **return** diff / ***HOUR\_MILLIS*** + **" hours ago"**;  
 } **else if** (diff < 48 \* ***HOUR\_MILLIS***) {  
 **return "yesterday"**;  
 } **else** {  
 **return** diff / ***DAY\_MILLIS*** + **" days ago"**;  
 }  
}

**//To get the time ago/since**

String temp = DateUtil.*getTimeAgo*(parsedMs, **this**.**context**, now);

**ADD CUSTOM HEADER VALUES WHEN LOADING A URL in WebView**

<http://stackoverflow.com/questions/7610790/add-custom-headers-to-webview-resource-requests-android>

String token = **“myToken”**;  
String username = **“myName”**;

String password = **“myPassword;**  
Map<String, String> **extraHeaders** = **new** HashMap<String, String>();  
extraHeaders.put(**"X-XSRF-TOKEN"**, token);  
extraHeaders.put(**"X-USERNAME"**, username);

extraHeaders.put(**"X-PASSWORD"**, password);  
**webView**.loadUrl(url, **extraHeaders**);

**PUSH NOTIFICATION**

<http://blog.android-develop.com/2014/03/gcm-push-notification-testing.html>

**GET THE FIRST AND LAST DATE OF THE MONTH (JODATIME)**

[**https://gist.github.com/marti1125/7405008**](https://gist.github.com/marti1125/7405008)

DateTimeFormatter formatterFecha = DateTimeFormat.forPattern("dd/MM/yyyy");

DateTime primerDiaDelMes = new DateTime().dayOfMonth().withMinimumValue();

String desde = new LocalDate(primerDiaDelMes).toString(formatterFecha);

DateTime ultimoDiaDelMes = new DateTime().dayOfMonth().withMaximumValue();

String hasta = new LocalDate(ultimoDiaDelMes).toString(formatterFecha);

**RXJAVA UNSUBSCRIBE**

* **do it in onFinish()**

**if** (**sendPushReadSubscription**!=**null** && **sendPushReadSubscription**.isUnsubscribed()==**false**){  
 **sendPushReadSubscription**.unsubscribe();  
}

**RXJAVA 20second TIMER**

**- Reset**

**public void resetTimer**(){  
 **if** (**disconnectTimerSubscription**!=**null** && **disconnectTimerSubscription**.isUnsubscribed()==**false**){  
 **disconnectTimerSubscription**.unsubscribe();  
 }  
 **disconnectTimerSubscription** =  
 Observable  
 .*interval*(1000, TimeUnit.***MILLISECONDS***)  
 .subscribeOn(Schedulers.*io*())  
 .observeOn(AndroidSchedulers.*mainThread*())  
 .subscribe(getTimerTickSubscriber());  
}

**public** Action1<Long> **getTimerTickSubscriber**(){  
 **return new** Action1<Long>() {  
 @Override  
 **public void** call(Long aLong) {  
 Log.*d*(***TAG***, **"getTimerTickSubscriber() call() finish timer:"**+aLong);  
 **if** (aLong==19){ //20 second   
 stopTimer();  
 }  
 }  
 }  
 };  
}

**- Stop**

**public void stopTimer**(){  
 Log.*d*(***TAG***, **"stopTimer() "**);  
 **if** (**disconnectTimerSubscription**!=**null** && **disconnectTimerSubscription**.isUnsubscribed()==**false**){  
 **disconnectTimerSubscription**.unsubscribe();  
 Log.*d*(***TAG***, **"stopTimer() 2"**);  
 }  
}

**SOFTWARE LICENSEs**

[**http://www.gnu.org/licenses/gpl-faq.en.html#GPLCommercially**](http://www.gnu.org/licenses/gpl-faq.en.html#GPLCommercially)

**CHECKING SERVER**

<http://stackoverflow.com/questions/8919083/checking-host-reachability-availability-in-android>

**CHECK IF DEVICE IS TABLET**

<http://stackoverflow.com/questions/9279111/determine-if-the-device-is-a-smartphone-or-tablet>

**PERFORMANCE IMPROVEMENTS**

<http://www.curious-creature.com/2012/12/01/android-performance-case-study/>

**MAPS**

Drawing custome icons, show title of all markers

<https://code.google.com/p/gmaps-api-issues/issues/detail?id=5240>

<https://www.youtube.com/watch?v=nb2X9IjjZpM>

<https://developers.google.com/maps/documentation/android-api/views#updating_the_camera_view>

<http://googlemaps.github.io/android-maps-utils/>

<https://www.youtube.com/watch?v=_oZiK_NJuG8&feature=youtu.be>

Draw custom icons on map using xml layout

<http://stackoverflow.com/questions/14811579/how-to-create-a-custom-shaped-bitmap-marker-with-android-map-api-v2>

<https://github.com/googlemaps/android-maps-utils>

**HANDLING LAYOUTS FOR DIFFERENT DEVICES**

<https://github.com/intuit/sdp>

**CHANGE TEXTVIEW TEXTSIZE PROGRAMMATICALLY**

myTextView.setTextSize(

TypedValue.***COMPLEX\_UNIT\_PX***, getResources().getDimensionPixelSize(R.dimen.***txt\_size65***)

);

where dimens.xml

<**dimen name="txt\_size65"**>65sp</**dimen**>

**ORIENTATION CHANGES**

<http://code.hootsuite.com/orientation-changes-on-android/>

**FORM SUBMIT ON ENTER/DONE**

**MOVE TO NEXT EDITTEXT ON ‘next’ click in softkeyboard**

<http://stackoverflow.com/questions/19217582/implicit-submit-after-hitting-done-on-the-keyboard-at-the-last-edittext>

**XML LAYOUT**

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/template\_forget\_password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"**

**android:orientation="vertical">**

**>**

< **EditText  
 android:id="@+id/firstname"  
 style="@style/edtStyle"   
 android:imeOptions="actionNext" 🡨 Important: this allows the next button to transfer focus to the next edittext  
 android:hint="@string/label\_signin\_tenant"  
 android:maxLines="1" 🡨 Important: needed for the next button to function**

**android:inputType="text" 🡨 Important: needed for the next button to function**

/>

< **EditText  
 android:id="@+id/lastname"  
 style="@style/edtStyle"  
 android:imeOptions="actionNext"   
 android:hint="@string/label\_signin\_tenant"  
 android:maxLines="1"**

**android:inputType="text"** />

< **EditText  
 android:id="@+id/age"  
 style="@style/edtStyle"  
 android:imeOptions="actionDone" 🡨 Important: this is the last edit text, the next button will become DONE, this should correspond to the one in .java file  
 android:hint="@string/label\_signin\_tenant"  
 android:maxLines="1"**

**android:inputType="text"** />

</**LinearLayout>**

**TO LISTEN To the DONE button**

**…**

@Bind(R.id.***age***)

EditText **mAge**;

**…**

**mAge**.setOnEditorActionListener(**new** TextView.OnEditorActionListener() {  
 @Override  
 **public boolean** onEditorAction(TextView v, **int** actionId, KeyEvent event) {  
 **if** (actionId == EditorInfo.***IME\_ACTION\_DONE***) { **<- this is the Done button from softkeyboard**  
 **mLoginButton**.performClick(); **<- this is the submit button**  
 **return true**;  
 }  
 **return false**;  
 }  
});

**TO LISTEN To the KEYBOARD SHOW HIDE**

**XML**

* add an id to the root view of your xml e.g. **R.id.content**

**ACTIVITY**

* add the following java code

**int keyboardHeight** = 0;  
**int oldKeyboardHeight** = 0 + **keyboardHeight**;  
  
**private void** checkKeyboardHeight() {  
 **final** View rootLayout = findViewById(R.id.***content***);  
 **if** (rootLayout != **null**) {  
 rootLayout.getViewTreeObserver().addOnGlobalLayoutListener(**new** ViewTreeObserver.OnGlobalLayoutListener() {  
 @Override  
 **public void** onGlobalLayout() {  
 Rect r = **new** Rect();  
  
 rootLayout.getWindowVisibleDisplayFrame(r);  
  
 **int** screenHeight = rootLayout.getRootView().getHeight();  
 **keyboardHeight** = screenHeight - (r.**bottom**);  
  
 **if** (**oldKeyboardHeight**!=**keyboardHeight**){  
 **oldKeyboardHeight** = 0 + **keyboardHeight**;  
 **if** (**keyboardHeight**>0){  
 *//KEYBOARD IS SHOWING* Log.*d*(***TAG***, **"checkKeyboardHeight() keyboard showing"**);  
 onKeyboardShow();  
 }**else**{  
 *//KEYBOARD IS NOT SHOWING* Log.*d*(***TAG***, **"checkKeyboardHeight() keyboard is not showing"**);  
 onKeyboardHide();  
 }  
 }**else**{  
 Log.*d*(***TAG***, **"checkKeyboardHeight() no change in keyboard visibility"**);  
 }  
  
 *//Log.d(TAG, "checkKeyboardHeight() onGlobalLayout() keyboardHeight:" + keyboardHeight);* }  
 });  
 }  
}

* call the above code in onCreate(),
* **onKeyboardShow()** will be called when the keyboard shows, **onKeyboardHide()** will be called when the keyboard is hidden

**RxJAVA RxAndroid listen to click listener**

**TAG: double click, triple click, quadruple click**

**Create an Observable from onClick() of View**

**public** Observable<View> getOnClickObservable(**final** View view){  
 **return** Observable.*create*(**new** Observable.OnSubscribe<View>() {  
 @Override  
 **public void** call(**final** Subscriber<? **super** View> subscriber) {  
 view.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Log.*d*(***TAG***, **"onClick: "**);  
 *//if (subscriber.isUnsubscribed()) return;* subscriber.onNext(v);  
 }  
 });  
 }  
 });  
}

**The 4th click subscriber**

@NonNull  
**private** Subscriber<List<View>> getQuadroupleClickSubscriber() {  
 **return new** Subscriber<List<View>>() {  
 @Override  
 **public void** onCompleted() {  
 Log.*d*(***TAG***, **"onCompleted: "**);  
 }  
  
 @Override  
 **public void** onError(Throwable e) {  
 Log.*d*(***TAG***, **"onError: "**);  
 }  
  
 @Override  
 **public void** onNext(List<View> views) {  
 Log.*d*(***TAG***, **"onNext: "**+views.size());  
 **//CODE HERE TO DO AFTER THE 4th CLICK HAPPENS**  
 }  
 };  
}

**The filter to get the 4 clicks**

Observable<View> loadingObservable = getOnClickObservable (**mLoading**);  
**loadingFourClickSubscription** = loadingObservable  
 .buffer(1000, TimeUnit.***MILLISECONDS***) //THE BUFFER ENSURES THE 4 CLICKS HAPPENS IN A SPAN OF 1 SECOND  
 .filter(**new** Func1<List<View>, Boolean>() {  
 @Override  
 **public** Boolean call(List<View> views) {  
 //RETURNS **true** IF IT IS ALREADY THE 4TH CLICK

**return** views.size() == 4;

}  
 }).observeOn(AndroidSchedulers.*mainThread*())  
 .subscribeOn(Schedulers.*io*())  
 .subscribe(getQuadroupleClickSubscriber());

**Listen to clock changes, time change**

<http://stackoverflow.com/questions/8500900/how-to-know-if-the-system-clock-has-changed>

**Section on recyclerview**

[**https://gist.github.com/gabrielemariotti/4c189fb1124df4556058**](https://gist.github.com/gabrielemariotti/4c189fb1124df4556058)

**PLAYSTORE multiple APKS**

[**https://developer.android.com/google/play/publishing/multiple-apks.html#Rules**](https://developer.android.com/google/play/publishing/multiple-apks.html#Rules)

[**https://developer.android.com/guide/practices/screens-distribution.html**](https://developer.android.com/guide/practices/screens-distribution.html)

**Playstore keywords, playstore tags**

<http://stackoverflow.com/questions/15836577/can-i-add-tags-to-an-application-in-google-play-console>

<https://support.google.com/googleplay/android-developer/answer/4448378?hl=en>

**SharedPref util, SharedPreference Util**

**public class** SharedPrefUtil {  
  
 **public static** SharedPreferences getPreferences(Context context){  
 **return** ((MyApplication) context.getApplicationContext())  
 .getApplicationPreferences();  
 }  
  
 **public static void** updatePref(String prefKey, String prefValue, Context context){  
 SharedPreferences prefs = *getPreferences*(context);  
 SharedPreferences.Editor editor = prefs.edit();  
 editor.putString(prefKey, prefValue);  
 **if** (prefs.contains(prefKey) ==**false**) {  
 editor.commit();  
 }**else** {  
 editor.apply();  
 }  
 }  
 **public static** String getPrefValue(String prefKey, Context context){  
 SharedPreferences prefs = *getPreferences*(context);  
 **return** prefs.getString(prefKey, **null**);  
  
 }  
  
 **public static void** updatePref(String prefKey, **int** prefValue, Context context){  
 SharedPreferences prefs = *getPreferences*(context);  
 SharedPreferences.Editor editor = prefs.edit();  
 editor.putInt(prefKey, prefValue);  
 **if** (prefs.contains(prefKey) ==**false**) {  
 editor.commit();  
 }**else** {  
 editor.apply();  
 }  
 }  
  
 */\*\*  
 \*  
 \** ***@param prefKey*** *\** ***@param context*** *\** ***@return*** *int -1 if not found  
 \*/* **public static int** getIntPrefValue(String prefKey, Context context){  
 SharedPreferences prefs = *getPreferences*(context);  
 **return** prefs.getInt(prefKey, -1);  
  
 }  
}

**PLAYSTORE SHOWING**

**App Is Designed for phones**

My app is restricted for tablet devices only as of the moment. I have added the <support-screens/> tag to my manifest and added layout supports for sw600dp and sw720dp. I do not intend the app to be released for phones yet. Please kindly do manual checking of my app again. Thanks.

<http://stackoverflow.com/questions/26719442/android-app-for-tablets-still-showing-designed-for-phones>

**BUTTON PRESS, PRESSED, DEPRESSED STATE**

*<?***xml version="1.0" encoding="UTF-8"***?>  
<!--  
<shape xmlns:android="http://schemas.android.com/apk/res/android" >  
 <solid android:color="@color/button\_color1" />  
 <corners android:radius="5dp" />  
</shape>  
-->*<**selector xmlns:android="http://schemas.android.com/apk/res/android"**>  
 <**item android:state\_pressed="true" android:drawable="@drawable/button\_round\_dark"**/>  
 <**item android:drawable="@drawable/button\_round\_light"**/>  
</**selector**>

**Check if app is in background**

<http://stackoverflow.com/questions/4414171/how-to-detect-when-an-android-app-goes-to-the-background-and-come-back-to-the-fo>

**OPACITY PERCENTAGE TO HEX**

[**http://stackoverflow.com/questions/15852122/hex-transparency-in-colors**](http://stackoverflow.com/questions/15852122/hex-transparency-in-colors)

* 100% — FF
* 95% — F2
* 90% — E6
* 85% — D9
* 80% — CC
* 75% — BF
* 70% — B3
* 65% — A6
* 60% — 99
* 55% — 8C
* 50% — 80
* 45% — 73
* 40% — 66
* 35% — 59
* 30% — 4D
* 25% — 40
* 20% — 33
* 15% — 26
* 10% — 1A
* 5% — 0D
* 0% — 00

**GOOGLE MAP LIBRARY**

<https://github.com/googlemaps/android-maps-utils>

**GPS ACCURACY**

<http://www.directionsmag.com/entry/10-things-you-need-to-know-about-indoor-positioning/324602>

\* accuracy depends on:

- GPS, Wi-Fi, cellular data, and device hardware.  If you’re experiencing location issues on your device, please give these following steps a try:

\* <http://stackoverflow.com/questions/34037764/mobile-better-gps-accuracy-sensor-vs-google-maps-provided-location>

- google maps accuracy is 10-30meters. GPS needs a clear sky view to be very accurate,

- If the mobile phone is inside a building and can not retrieve the correct location, the GPS will automatically point to the nearest street as its new location.

GEOFENCE

<https://developer.android.com/training/location/geofencing.html>

Google maps also provides geofence - a circular perimeter that indicates the users location if the phone is offline, this gives 0.7-1.5 miles accuracy.

* minimum radius of 100m

GPS Accuracy study

<https://communityhealthmaps.nlm.nih.gov/2014/06/30/how-accurate-is-the-gps-on-my-smartphone/>

<https://communityhealthmaps.nlm.nih.gov/2014/07/07/how-accurate-is-the-gps-on-my-smart-phone-part-2/>

When Wi-Fi is available location accuracy is usually between 20 - 50 meters.

When indoor location is available, the accuracy range can be as small as 5 meters.

<https://www.quora.com/How-accurate-is-Android-GPS-on-Google-Map>

<https://support.google.com/maps/answer/2839911?co=GENIE.Platform%3DAndroid&hl=en>

**GET DIMENSION FROM XML THEN SET TO TEXTVIEW AT RUNTIME**

<http://stackoverflow.com/questions/14540293/get-dimension-from-xml-and-set-text-size-in-runtime>

dimens.xml  
<dimen name="text\_medium">18sp</dimen>

then

textView.setTextSize(TypedValue.COMPLEX\_UNIT\_PX, getResources().getDimension(R.dimen.text\_medium));

**CHECK IF APP WENT TO BACKGROUND**

<http://stackoverflow.com/questions/4414171/how-to-detect-when-an-android-app-goes-to-the-background-and-come-back-to-the-fo>

**PREVENT APP FROM BEING KILLED BY BACK BUTTON**

<http://stackoverflow.com/questions/6514657/prevent-back-button-from-closing-my-application>

/\* Prevent app from being killed on back, Prevent back button from closing application

\*/

@Override

public boolean onKeyDown(int keyCode, KeyEvent event) {

// Back?

if (keyCode == KeyEvent.KEYCODE\_BACK) {

// Back

Log.d(TAG, "onKeyDown: moved to background");

moveTaskToBack(true);

return true;

}

else {

// Return

return super.onKeyDown(keyCode, event);

}

}