**Important links for intents:**

<https://developer.android.com/guide/components/intents-common.html>

**In** [app](https://github.com/udacity/Sunshine-Version-2/tree/745a01e91230db498ae77adae3241f2edee83570/app)/[src](https://github.com/udacity/Sunshine-Version-2/tree/745a01e91230db498ae77adae3241f2edee83570/app/src)/[main](https://github.com/udacity/Sunshine-Version-2/tree/745a01e91230db498ae77adae3241f2edee83570/app/src/main)/[res](https://github.com/udacity/Sunshine-Version-2/tree/745a01e91230db498ae77adae3241f2edee83570/app/src/main/res)/[menu](https://github.com/udacity/Sunshine-Version-2/tree/745a01e91230db498ae77adae3241f2edee83570/app/src/main/res/menu)/**main.xml file we add this menu item for map which is available in overflow menu we also define string in string.xml file**

<item android:id="@+id/action\_map"

android:title="@string/action\_map"

app:showAsAction="never" />

**string.xml**

<string name="action\_map">Map Location</string>

**This menu will be inflated in main activity class**

**MainActivity we add this code for find location**

private void openPreferredLocationInMap() {

SharedPreferences sharedPrefs =

PreferenceManager.getDefaultSharedPreferences(this);

String location = sharedPrefs.getString(

getString(R.string.pref\_location\_key),

getString(R.string.pref\_location\_default));

// Using the URI scheme for showing a location found on a map. This super-handy

// intent can is detailed in the "Common Intents" page of Android's developer site:

// http://developer.android.com/guide/components/intents-common.html#Maps

Uri geoLocation = Uri.parse("geo:0,0?").buildUpon()

.appendQueryParameter("q", location)

.build();

Intent intent = new Intent(Intent.ACTION\_VIEW);

intent.setData(geoLocation);

if (intent.resolveActivity(getPackageManager()) != null) {

startActivity(intent);

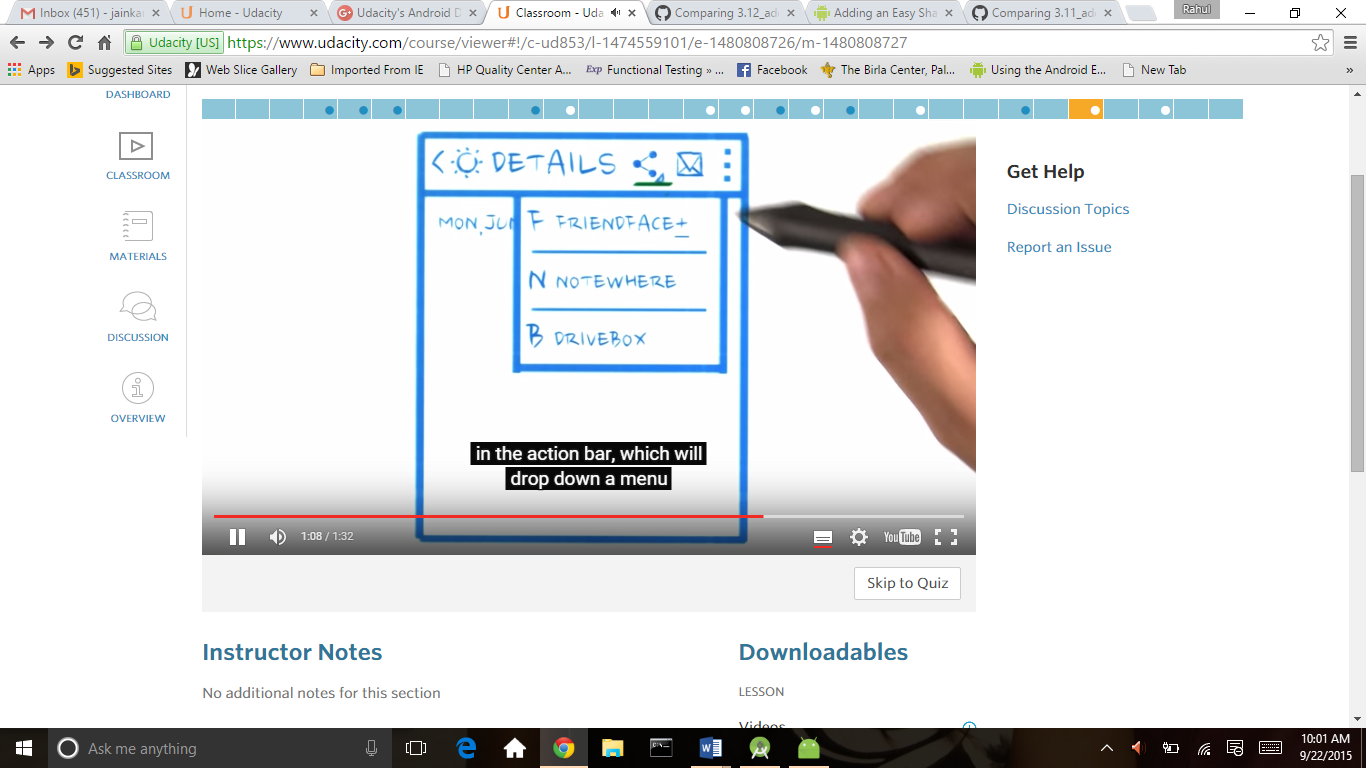
} else {

Log.d(LOG\_TAG, "Couldn't call " + location + ", no receiving apps installed!");

}

}

**Share Action provider**



<http://developer.android.com/training/sharing/shareaction.html>

for this first we add this line in string.xml

<string name="action\_share">Share</string>

Then add resource file in menu detailfragment

app:actionProviderClass="android.support.v7.widget.ShareActionProvider"

it is use for android support libeary according to our framework

<**menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"**>  
 <**item android:id="@+id/action\_share"  
 android:title="@string/action\_share"  
 app:showAsAction="always"  
 app:actionProviderClass="android.support.v7.widget.ShareActionProvider"** />  
  
</**menu**>

Then in detailactivity because detailactivity launch share action so we add following code

shareIntent.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_WHEN\_TASK\_RESET***);  
 shareIntent.setType(**"text/plain"**);

this flage is use for cleare the stack when thi is return in app

**public static class** DetailFragment **extends** Fragment {  
 **private static final** String ***LOG\_TAG*** = DetailFragment.**class**.getSimpleName();  
 **private static final** String ***FORECAST\_SHARE\_HASHTAG*** = **" #SunshineApp"**;  
 **private** String **mForecastStr**;  
  
 **public** DetailFragment() {  
 setHasOptionsMenu(**true**);  
 }  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
  
 View rootView = inflater.inflate(R.layout.***fragment\_detail***, container, **false**);  
  
 Intent intent = getActivity().getIntent();  
 **if** (intent != **null** && intent.hasExtra(Intent.***EXTRA\_TEXT***)) {  
 String forecastStr = intent.getStringExtra(Intent.***EXTRA\_TEXT***);  
 ((TextView) rootView.findViewById(R.id.***detail\_text***))  
 .setText(forecastStr);  
 }  
  
 **return** rootView;  
 }  
 @Override  
 **public void** onCreateOptionsMenu(Menu menu, MenuInflater inflater) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* inflater.inflate(R.menu.***detailfragment***, menu);  
  
 *// Retrieve the share menu item* MenuItem menuItem = menu.findItem(R.id.***action\_share***);  
  
 *// Get the provider and hold onto it to set/change the share intent.* ShareActionProvider mShareActionProvider =  
 (ShareActionProvider) MenuItemCompat.*getActionProvider*(menuItem);  
  
 *// Attach an intent to this ShareActionProvider. You can update this at any time,  
 // like when the user selects a new piece of data they might like to share.* **if** (mShareActionProvider != **null** ) {  
 mShareActionProvider.setShareIntent(createShareForecastIntent());  
 } **else** {  
 Log.*d*(***LOG\_TAG***, **"Share Action Provider is null?"**);  
 }  
 }  
 **private** Intent createShareForecastIntent() {  
 Intent shareIntent = **new** Intent(Intent.***ACTION\_SEND***);  
 shareIntent.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_WHEN\_TASK\_RESET***);  
 shareIntent.setType(**"text/plain"**);  
 shareIntent.putExtra(Intent.***EXTRA\_TEXT***,  
 **mForecastStr** + ***FORECAST\_SHARE\_HASHTAG***);  
 **return** shareIntent;  
 }  
}