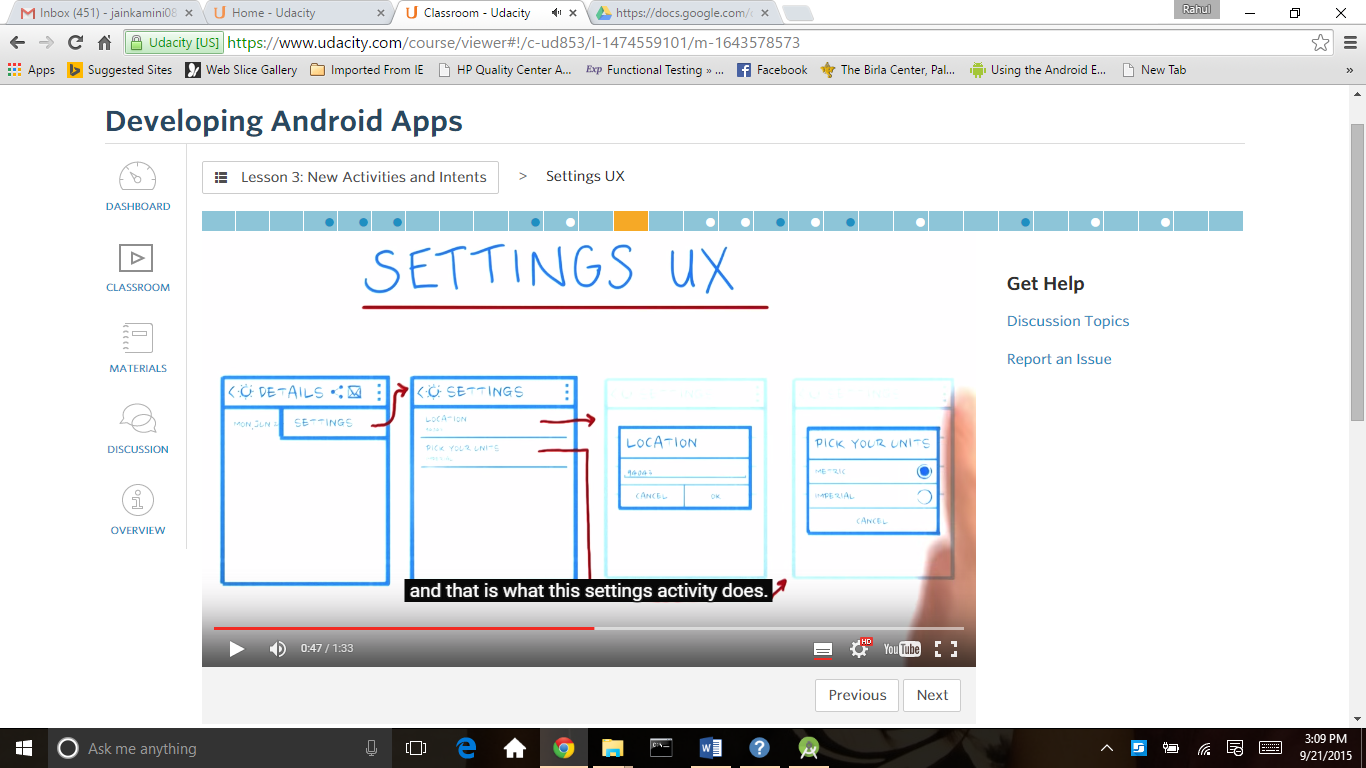
Setting Menu:-

How we set setting menu .what option is given in setting menu. What standard should we follows for this.



What is the setting and how we set setting according some standard is given is these links:

<https://www.google.com/design/spec/patterns/settings.html>

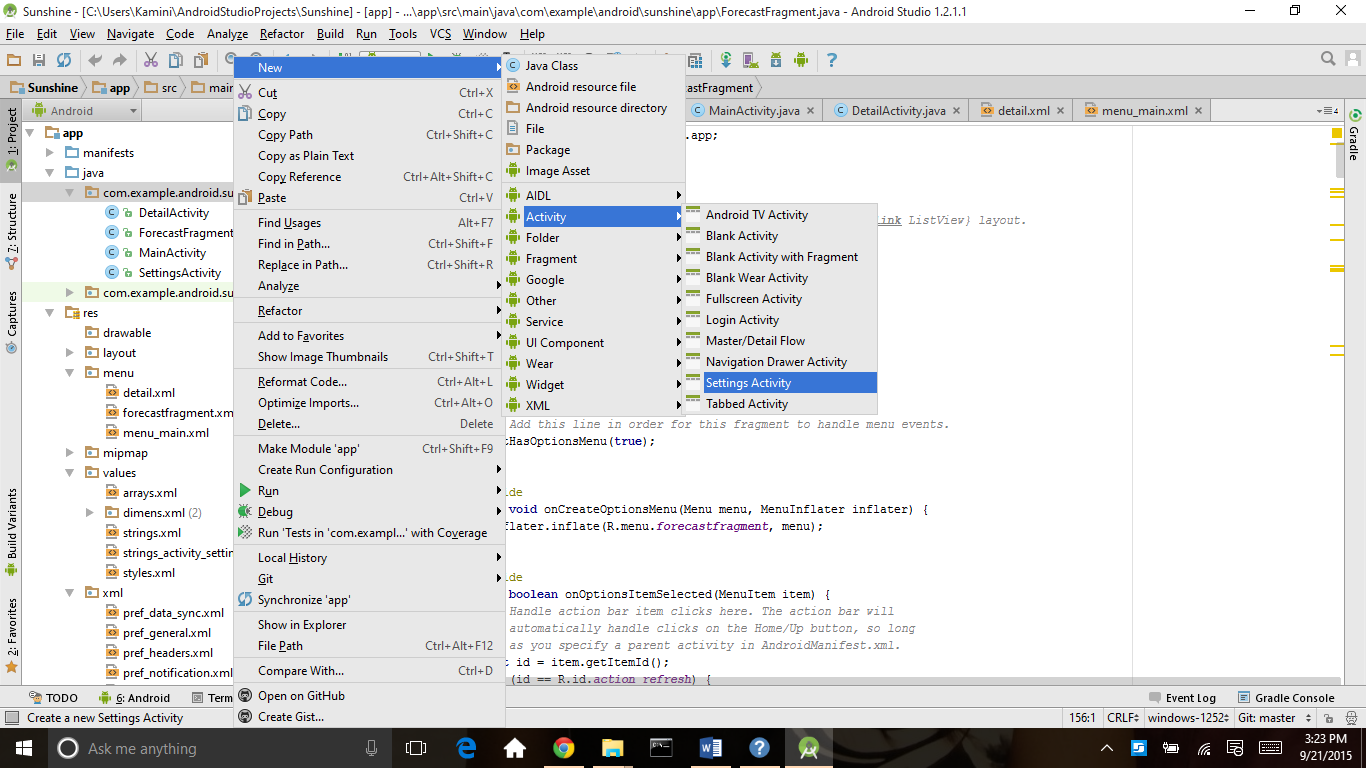
<http://developer.android.com/guide/topics/ui/settings.html>

<http://developer.android.com/reference/android/content/SharedPreferences.html>

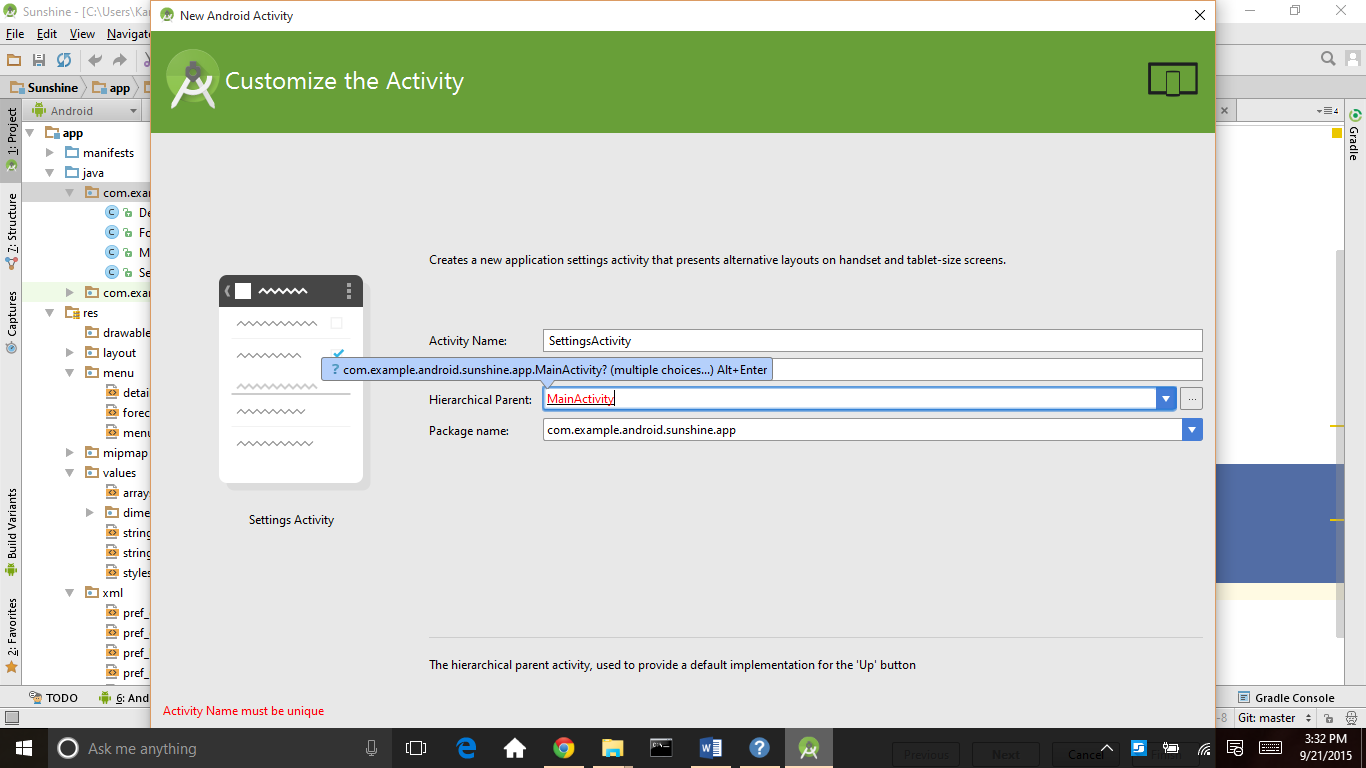
1 ) First create settings activity like screen 1 and then in hierarchical parent box write MainActivity like screen2 and then press finish button

So the mainActivity become parent for this activity(settings Activity)

Screen1

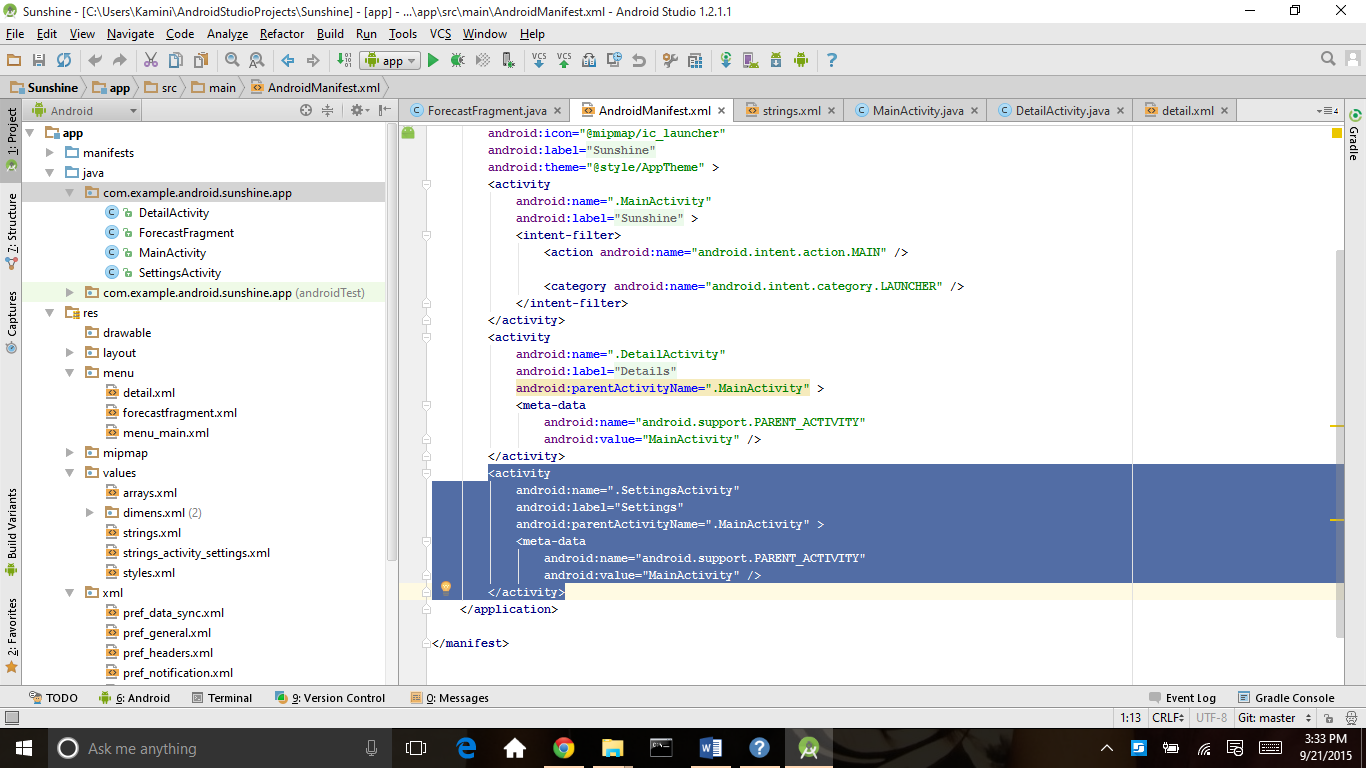


Screen2



In screen 3 Manifests.xml settinngsActivity is automatically added

Screen3



This is the code for settings Activity

package com.example.android.sunshine.app;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.preference.ListPreference;

import android.preference.Preference;

import android.preference.PreferenceActivity;

import android.preference.PreferenceManager;

import android.view.KeyEvent;

/\*\*

\* A {@link PreferenceActivity} that presents a set of application settings.

\* <p>

\* See <a href="http://developer.android.com/design/patterns/settings.html">

\* Android Design: Settings</a> for design guidelines and the <a

\* href="http://developer.android.com/guide/topics/ui/settings.html">Settings

\* API Guide</a> for more information on developing a Settings UI.

\*/

//some time activity should extend from fragments according to version

public class SettingsActivity extends PreferenceActivity

implements Preference.OnPreferenceChangeListener {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// Add 'general' preferences, defined in the XML file

// TODO: Add preferences from XML

// For all preferences, attach an OnPreferenceChangeListener so the UI summary can be

// updated when the preference changes.

// TODO: Add preferences

}

/\*\*

\* Attaches a listener so the summary is always updated with the preference value.

\* Also fires the listener once, to initialize the summary (so it shows up before the value

\* is changed.)

\*/

private void bindPreferenceSummaryToValue(Preference preference) {

// Set the listener to watch for value changes.

preference.setOnPreferenceChangeListener(this);

// Trigger the listener immediately with the preference's

// current value.

onPreferenceChange(preference,

PreferenceManager

.getDefaultSharedPreferences(preference.getContext())

.getString(preference.getKey(), ""));

}

@Override

public boolean onPreferenceChange(Preference preference, Object value) {

String stringValue = value.toString();

if (preference instanceof ListPreference) {

// For list preferences, look up the correct display value in

// the preference's 'entries' list (since they have separate labels/values).

ListPreference listPreference = (ListPreference) preference;

int prefIndex = listPreference.findIndexOfValue(stringValue);

if (prefIndex >= 0) {

preference.setSummary(listPreference.getEntries()[prefIndex]);

}

} else {

// For other preferences, set the summary to the value's simple string representation.

preference.setSummary(stringValue);

}

return true;

}

}

Then setting activity launch by main activity or maybe detail activity according to requirement we add following code in mainactivity or any other activity where we launch settings activity(green color code) in onoption item selected method

|  |
| --- |
|  |
|  | @Override  public boolean onOptionsItemSelected(MenuItem item) { |
|  | // Handle action bar item clicks here. The action bar will |
|  | // automatically handle clicks on the Home/Up button, so long |
|  | // as you specify a parent activity in AndroidManifest.xml. |
|  | int id = item.getItemId(); |
|  |  |
|  | //noinspection SimplifiableIfStatement |
|  | if (id == R.id.action\_settings) {  //this is a explicit intent to launch settingsactivity |
|  | startActivity(new Intent(this, SettingsActivity.class)); |
|  | return true; |
|  | } |
|  |  |
|  | return super.onOptionsItemSelected(item); |
|  | } |

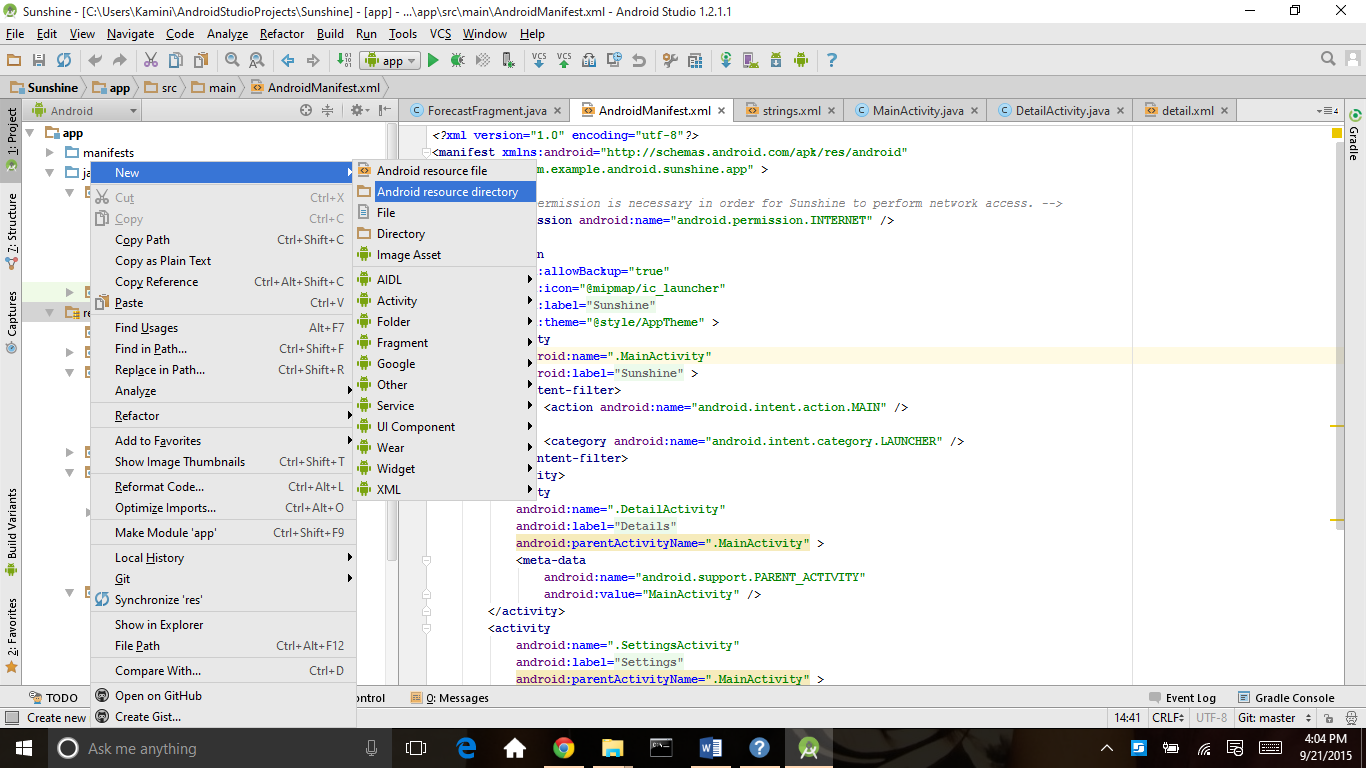
Now we have a blank settings activity so we add some preference in it according to our requirement

For preference see this link:

<https://developer.android.com/guide/topics/ui/settings.html#DefiningPrefs>

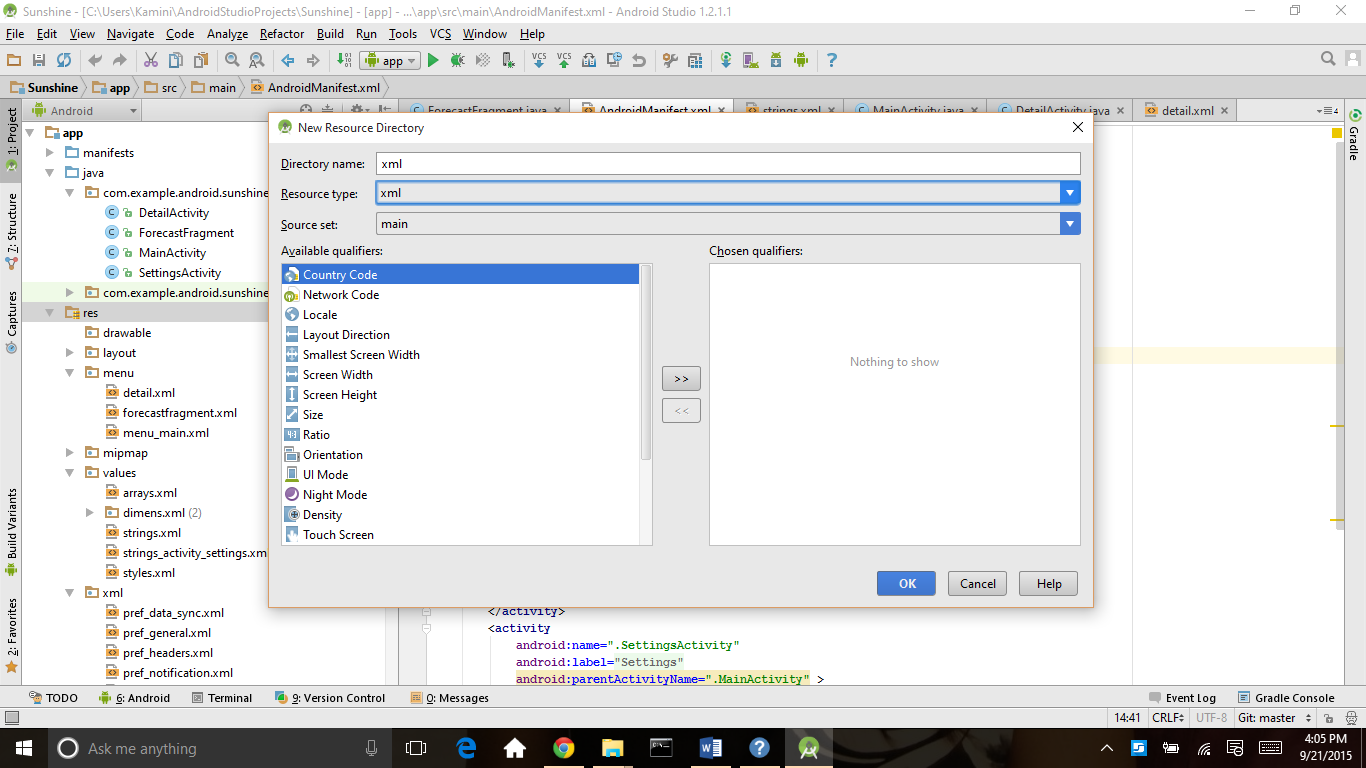
For preference we create a xml folder under res according to screen 4

Screen 4

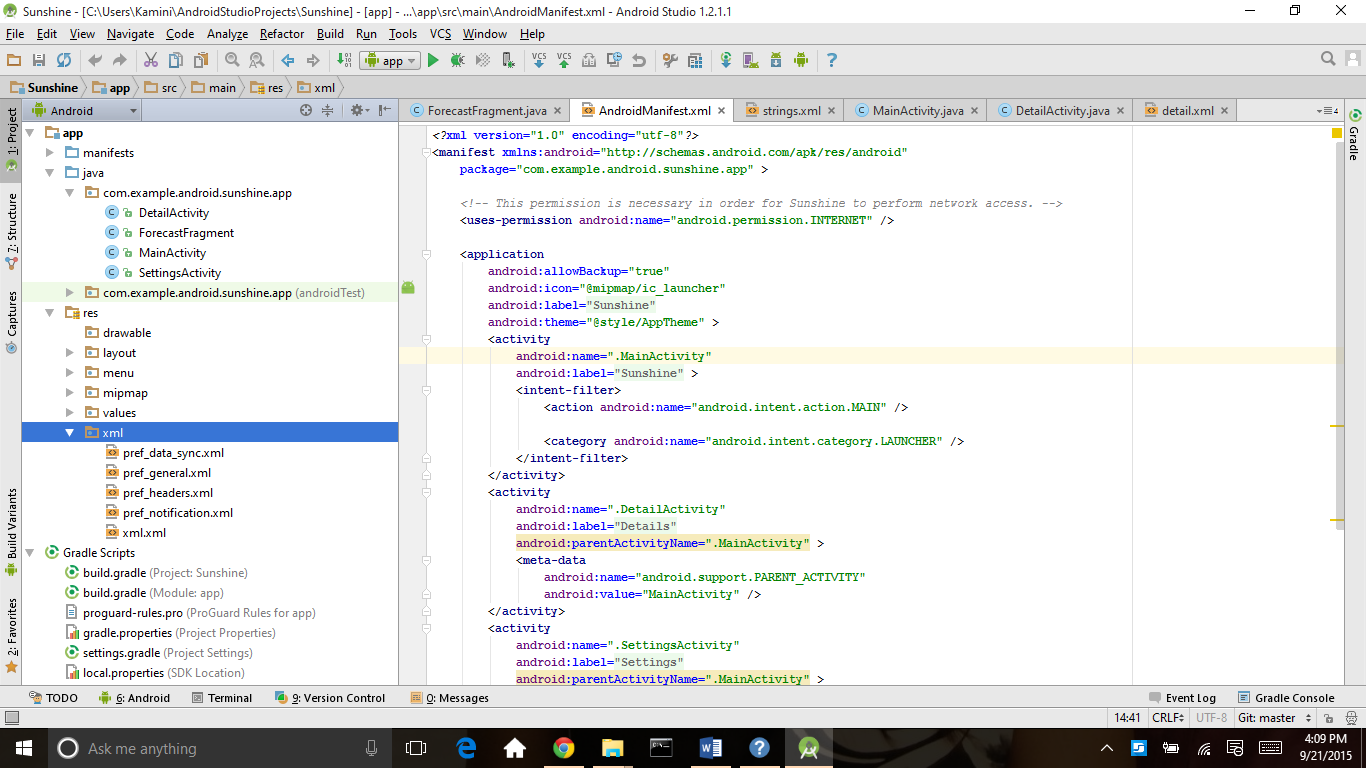


In screen 5 we type directory name as xml and we select resource type xml and then press ok

Screen5



In screen 6 all the preference related file will be automatically add Screen6



Then we add location preference in pref\_general.xml in our rquirment location should edit text so we add edit preference for location

We set some property of edit text preference like android:title, android:key, android:defaultValue etc.key is value for sharedpreference

|  |
| --- |
|  |
|  | <?xml version="1.0" encoding="utf-8"?>  <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" |
|  | android:layout\_width="match\_parent" |
|  | android:layout\_height="match\_parent"> |
|  |  |
|  | <EditTextPreference |
|  | android:title="@string/pref\_location\_label" |
|  | android:key="@string/pref\_location\_key" |
|  | android:defaultValue="@string/pref\_location\_default" |
|  | android:inputType="text" |
|  | android:singleLine="true" /> |
|  |  |
|  | </PreferenceScreen> |

|  |
| --- |
|  |
|  | String.xml  <string name="pref\_location\_label">Location</string> |
|  |  |
|  | <!-- Key name for storing location in SharedPreferences [CHAR LIMIT=NONE] --> |
|  | <string name="pref\_location\_key" translatable="false">location</string> |
|  |  |
|  | <!-- Default postal code for location preference [CHAR LIMIT=NONE] --> |
|  | <string name="pref\_location\_default" translatable="false">94043</string> |
|  |  |

Now add follwing code in settings activity(green color code )

Add general preference define in xml file.

For all preferences, attach an OnPreferenceChangeListener so the UI summary can be updated when the preference changes.

Load preferences XML layout in SettingsActivity with

addPreferencesFromResource(R.xml.pref\_general);

(Yes, we know that this function is deprecated, but it is actually the current best practice given that we're targeting Gingerbread!)

Bind preference summary to location preference

bindPreferenceSummaryToValue(findPreference(getString(R.string.pref\_location\_key)));

public class SettingsActivity extends PreferenceActivity

implements Preference.OnPreferenceChangeListener {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// Add 'general' preferences, defined in the XML file

addPreferencesFromResource(R.xml.pref\_general);

// For all preferences, attach an OnPreferenceChangeListener so the UI summary can be

// updated when the preference changes.

bindPreferenceSummaryToValue(findPreference(getString(R.string.pref\_location\_key)));

}

Then we add this code in our main activity where we set value of sharedpreference which are coming from setting (green code ) one is the location and other one is location value like77429

Location =77429

FetchWeatherTask weatherTask = new FetchWeatherTask();

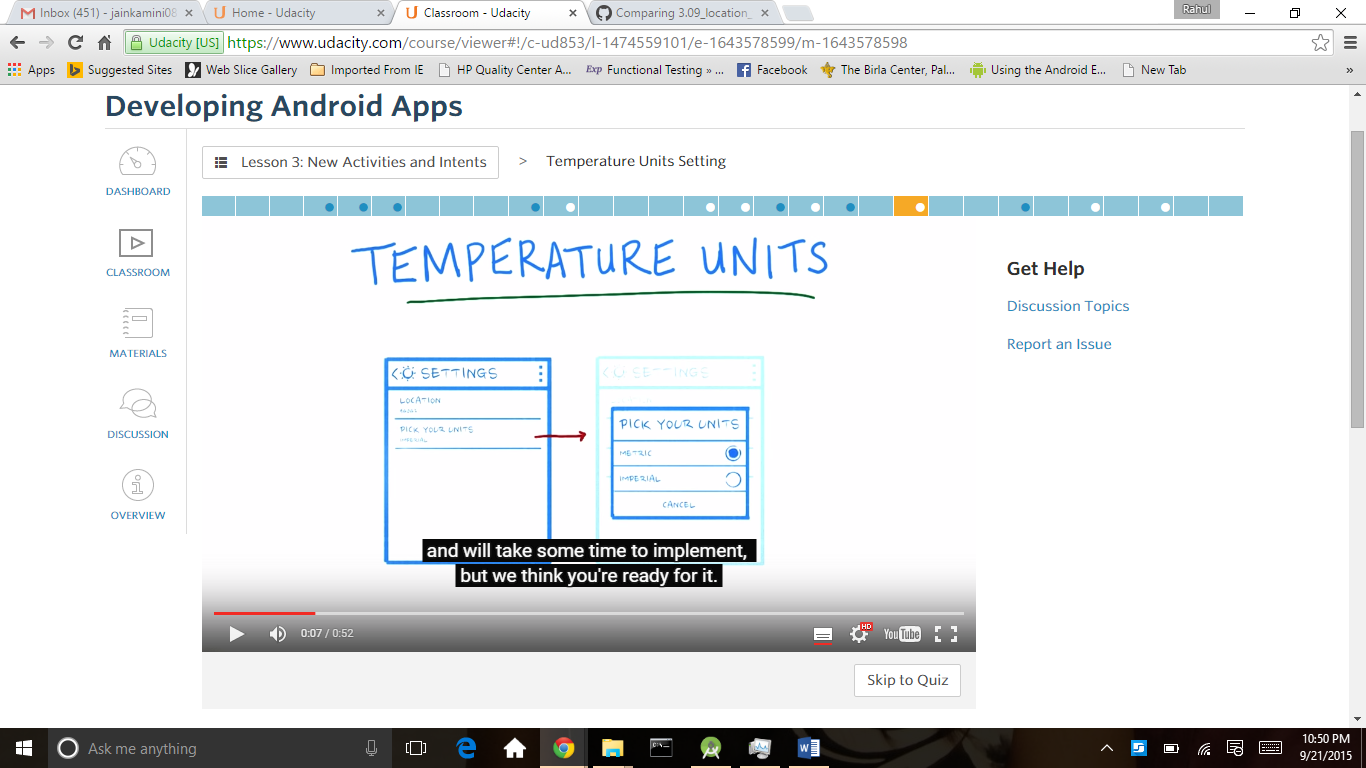
SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(getActivity());

String location = prefs.getString(getString(R.string.pref\_location\_key),

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

weatherTask.execute(location);

for listpreference



**In pref\_general.xml we add following code here android:title is title of heading** Temperature Units

<ListPreference

android:title="@string/pref\_units\_label"

android:key="@string/pref\_units\_key"

android:defaultValue="@string/pref\_units\_metric"

android:entryValues="@array/pref\_units\_values"

android:entries="@array/pref\_units\_options" /

string.xml

<!-- Label for the temperature units preference [CHAR LIMIT=30] -->

<string name="pref\_units\_label">Temperature Units</string>

<!-- Label for metric option in temperature unit preference [CHAR LIMIT=25] -->

<string name="pref\_units\_label\_metric">Metric</string>

<!-- Label for imperial option in temperature unit preference [CHAR LIMIT=25] -->

<string name="pref\_units\_label\_imperial">Imperial</string>

<!-- Key name for temperature unit preference in SharedPreferences [CHAR LIMIT=NONE] -->

<string name="pref\_units\_key" translatable="false">units</string>

<!-- Value in SharedPreferences for metric temperature unit option [CHAR LIMIT=NONE] -->

<string name="pref\_units\_metric" translatable="false">metric</string>

<!-- Value in SharedPreferences for imperial temperature unit option [CHAR LIMIT=NONE] -->

<string name="pref\_units\_imperial" translatable="false">imperial</string>

 app/src/main/res/values/arrays.xml add array.xml in values folder

<?xml version="1.0" encoding="utf-8"?>

<resources>

<string-array name="pref\_units\_options">

<item>@string/pref\_units\_label\_metric</item>

<item>@string/pref\_units\_label\_imperial</item>

</string-array>

<string-array name="pref\_units\_values">

<item>@string/pref\_units\_metric</item>

<item>@string/pref\_units\_imperial</item>

</string-array>

</resources>

bindPreferenceSummaryToValue(findPreference(getString(R.string.pref\_units\_key)));

this line is add in settings activity

add this code in activity where we get sharedprefrence value

SharedPreferences sharedPrefs =

PreferenceManager.getDefaultSharedPreferences(getActivity());

String unitType = sharedPrefs.getString(

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

getString(R.string.pref\_units\_metric));

then we add this for check different type of units pref\_units\_imperial or pref\_units\_metric and according this we do work in if else

if (unitType.equals(getString(R.string.pref\_units\_imperial))) {

high = (high \* 1.8) + 32;

low = (low \* 1.8) + 32;

} else if (!unitType.equals(getString(R.string.pref\_units\_metric))) {

Log.d(LOG\_TAG, "Unit type not found: " + unitType);

}