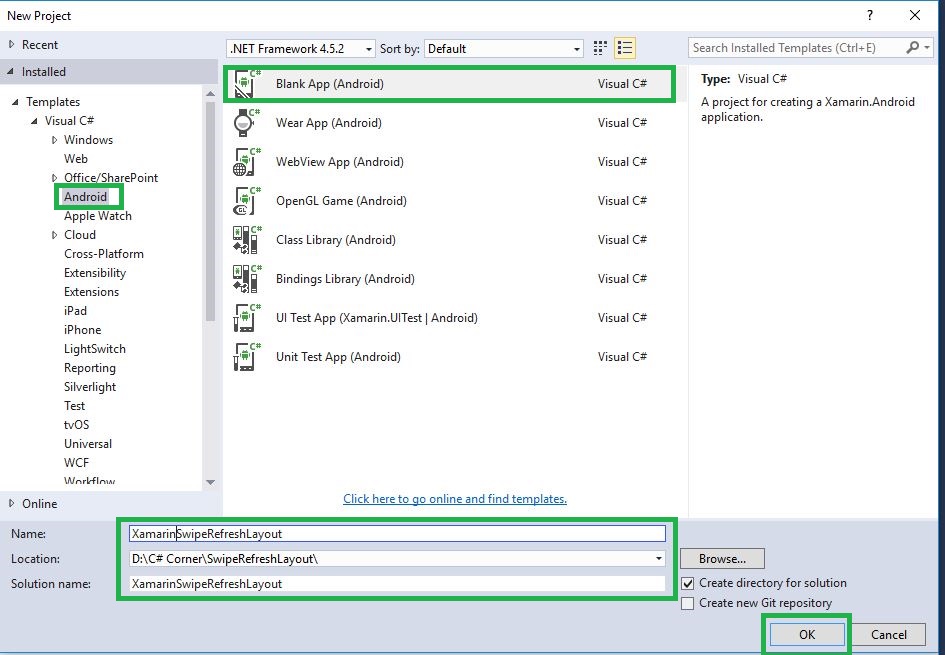
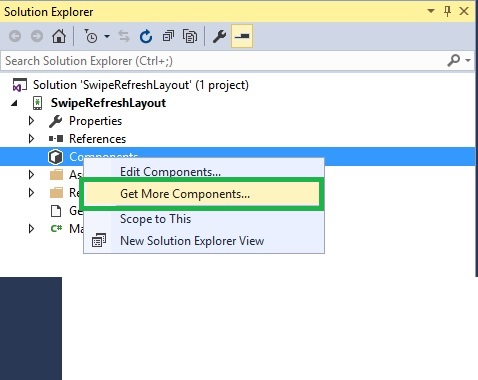
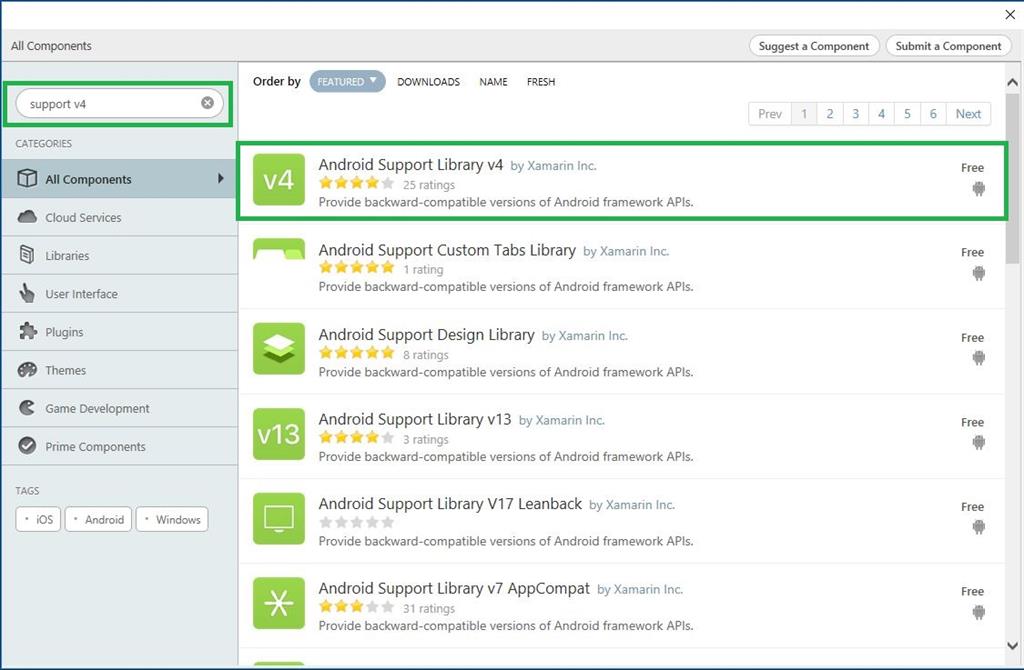
# Creating Xamarin Android SwipeRefreshLayout Using ListView Items

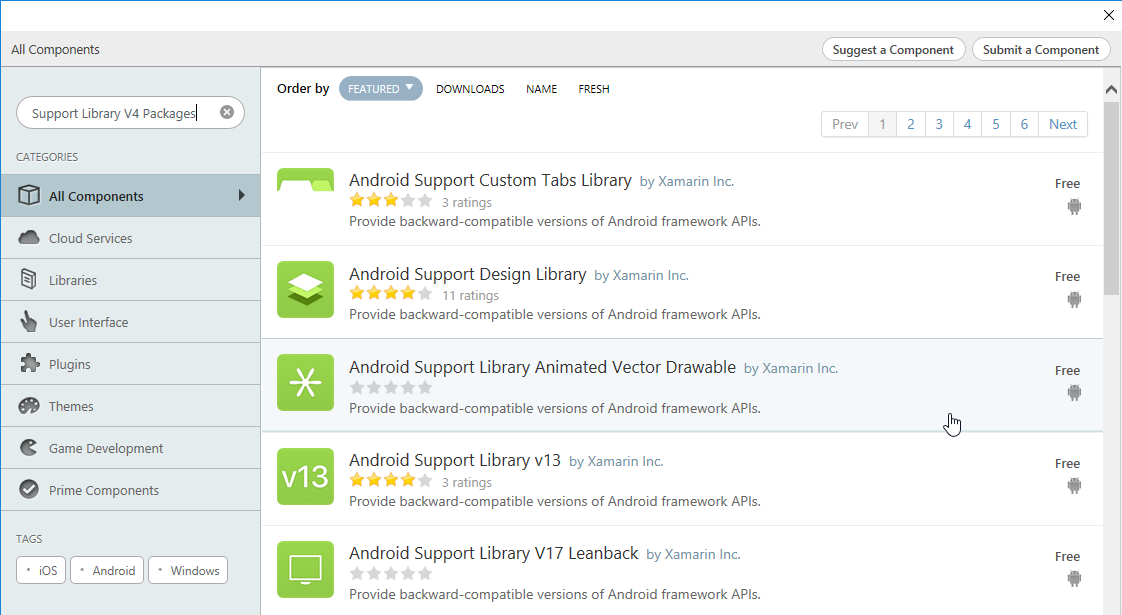
**Let’s start,**

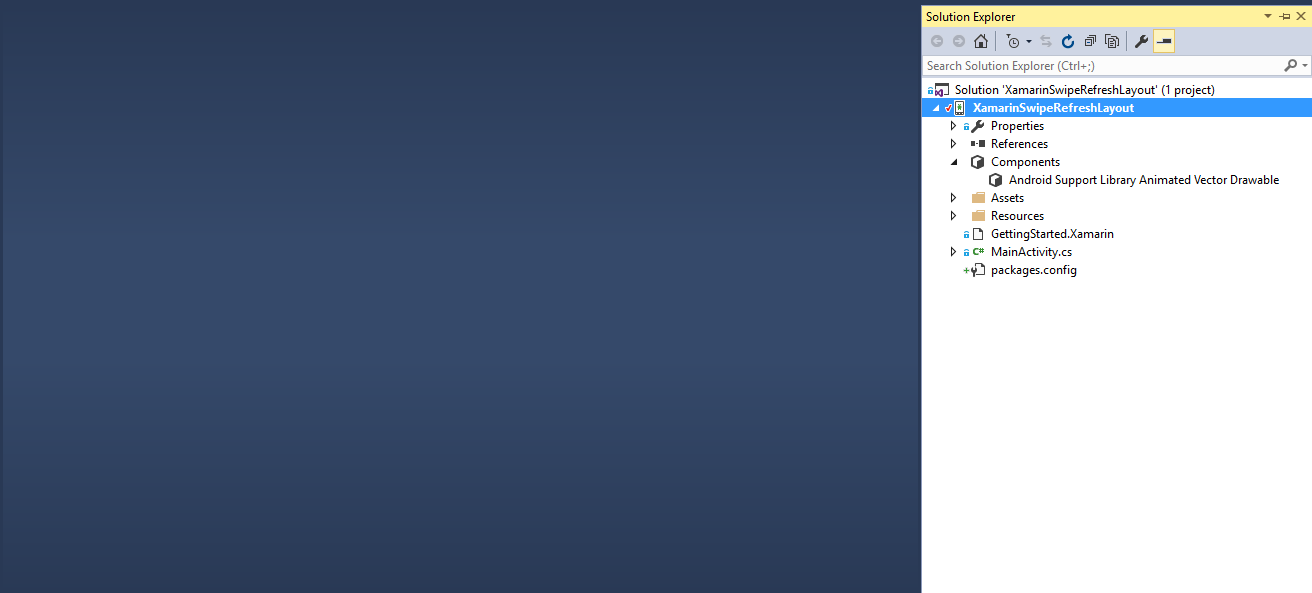
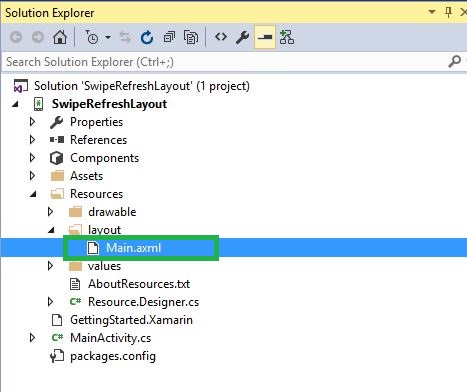
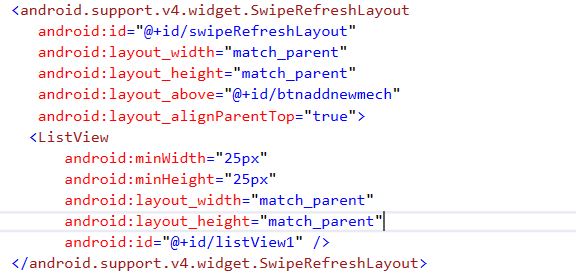
**Step 1:** Open Visual Studio->New Project->Templates->Visual C#->Android->Blank App

Select Blank App. Give the Project Name and Project Location.  
  
  
 **Step 2**Go to Solution Explorer-> Project Name-> Components, right click to Get More Components and the new dialog box opens. This dialog box is required to search the Support V4, add the Android Support Library V4 Packages.  
  


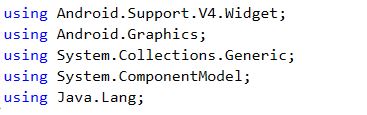
 

Or



  
 **Step 3**: To open Solution Explorer, go to -> Project Name->Resources->layout ->Main.axml, click to open Design View and the code, given below:  
  
  


1. <android.support.v4.widget.SwipeRefreshLayout
2. android:id="@+id/swipeRefreshLayout"
3. android:layout\_width="match\_parent"
4. android:layout\_height="match\_parent"
5. android:layout\_above="@+id/btnaddnewmech"
6. android:layout\_alignParentTop="true">
7. <ListView
8. android:minWidth="25px"
9. android:minHeight="25px"
10. android:layout\_width="match\_parent"
11. android:layout\_height="match\_parent"
12. android:id="@+id/listView1" />
13. </android.support.v4.widget.SwipeRefreshLayout>

**Step 4**: After Design view creation, open Solution Explorer-> Project Name->MainActivity.cs, add the namespaces, given below:  
  


1. **using** Android.Support.V4.Widget;
2. **using** Android.Graphics;
3. **using** System.Collections.Generic;
4. **using** System.ComponentModel;
5. **using** Java.Lang;

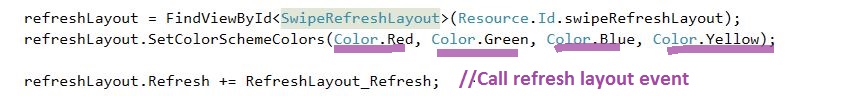
**Step 5:** To create the new variables for RefreshLayout and ListView, one can use the code, given below:  
  
 http://csharpcorner.mindcrackerinc.netdna-cdn.com/article/creating-xamarin-android-swiperefreshlayout-using-listview-items/Images/6.JPG

1. **private** SwipeRefreshLayout refreshLayout;
2. **private** ListView listitem;

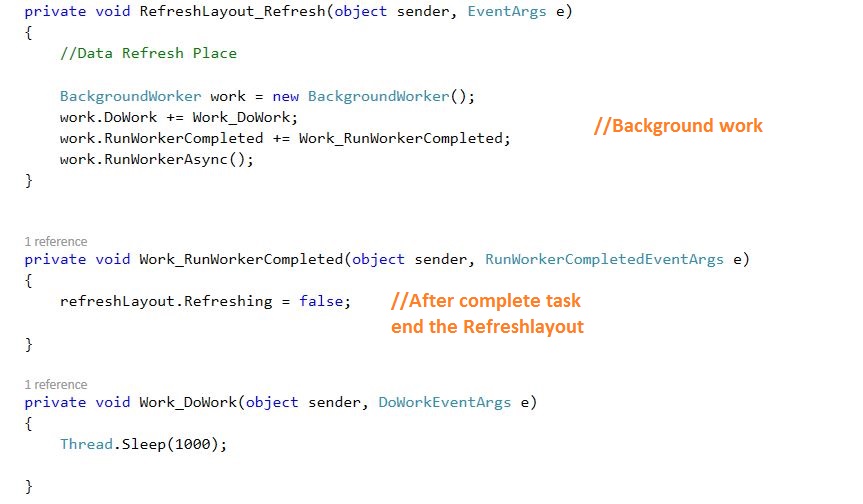
**Step 6:** Go to Oncreate() to declare the list view id and Listview<string>Variable.

**Step 7:** Next step is to create one List String Item and then add the list view adapter.  
  
  
**C# Code**

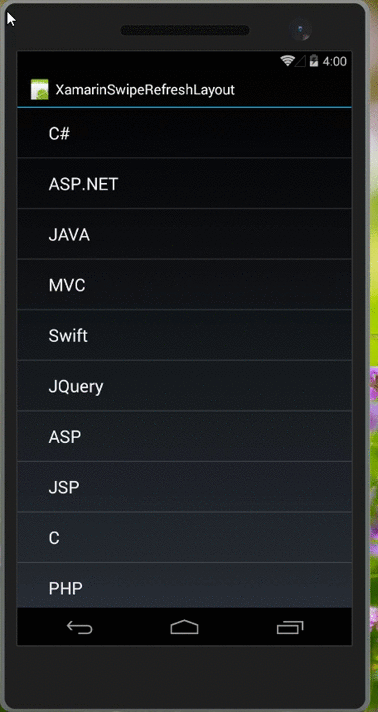
1. listitem = FindViewById < ListView > (Resource.Id.listView1);
2. List < **string** > items = **new** List < **string** > ();
3. items.Add("C#");
4. items.Add("ASP.NET");
5. items.Add("JAVA");
6. items.Add("MVC");
7. items.Add("Swift");
8. items.Add("JQuery");
9. items.Add("ASP");
10. items.Add("JSP");
11. items.Add("C");
12. items.Add("PHP");
13. items.Add("Ruby");
14. ArrayAdapter < **string** > liststring = **new** ArrayAdapter < **string** > (**this**, Android.Resource.Layout.SimpleExpandableListItem1, items);
15. listitem.Adapter = liststring;

**Step 8**After Listview declaration, assign Refresh Layout variables and colors. Here, create one more method for Refresh event. This event is declared after Oncreate().  
   
 **C# Code**

1. refreshLayout = FindViewById < SwipeRefreshLayout > (Resource.Id.swipeRefreshLayout);
2. refreshLayout.SetColorSchemeColors(Color.Red, Color.Green, Color.Blue, Color.Yellow);
3. refreshLayout.Refresh += RefreshLayout\_Refresh;

**Step 9**Declare RefreshLayout\_refresh method. This method is working in the background process of the layout. Once the work completes, it will terminate the work.  
   
**C# Code**

1. **private** **void** RefreshLayout\_Refresh(**object** sender, EventArgs e)
2. {
3. //Data Refresh Place
4. BackgroundWorker work = **new** BackgroundWorker();
5. work.DoWork += Work\_DoWork;
6. work.RunWorkerCompleted += Work\_RunWorkerCompleted;
7. work.RunWorkerAsync();
8. }
9. **private** **void** Work\_RunWorkerCompleted(**object** sender, RunWorkerCompletedEventArgs e) {
10. refreshLayout.Refreshing = **false**;
11. }
12. **private** **void** Work\_DoWork(**object** sender, DoWorkEventArgs e) {
13. Thread.Sleep(1000);
14. }

**Step 10:** Press F5 or build and run the Application.  
  
  
  
Finally, we successfully created Xamarin Android SwipeRefreshLayout with ListView items**.**