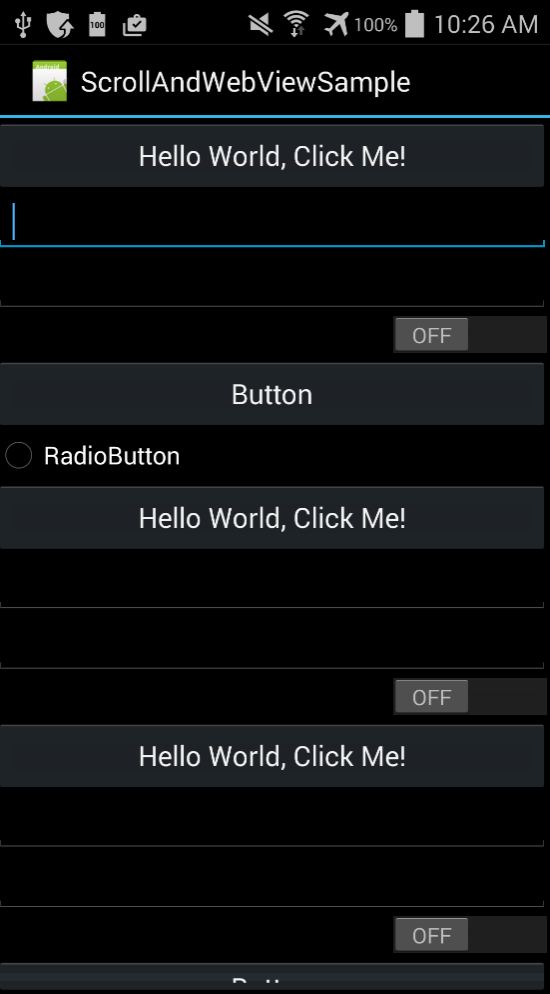
Day 22: ScrollView and WebView

Today, we will explore couple more of commonly used views in Xamarin.Android: ScrollView, WebView.

# ScrollView

ScrollView in Android is used in scenarios where the number of views on the screen overflow and present a need for a scroll bar to see the Views that are towards the end of the screen. Let’s see the problem at hand first. I took a layout file and filled it with bunch of Views so the full screen is not seen on the on presented screen height –

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent">  <Button  android:id="@+id/MyButton"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:text="Navigate to WebView" />  <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText1" />  <EditText  android:inputType="textMultiLine"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText2" />  <Switch  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/switch1" />  <Button  android:text="Button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/button1" />  <RadioButton  android:text="RadioButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/radioButton1" />  <Button  android:id="@+id/MyButton"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:text="@string/Hello" />  <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText4" />  <EditText  android:inputType="textMultiLine"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText3" />  <Switch  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/switch2" />  <Button  android:id="@+id/MyButton"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:text="@string/Hello" />  <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText6" />  <EditText  android:inputType="textMultiLine"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText5" />  <Switch  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/switch3" />  <Button  android:text="Button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/button3" />  <RadioButton  android:text="RadioButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/radioButton3" />  <Button  android:text="Button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/button3" />  <RadioButton  android:text="RadioButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/radioButton3" />  </LinearLayout> |

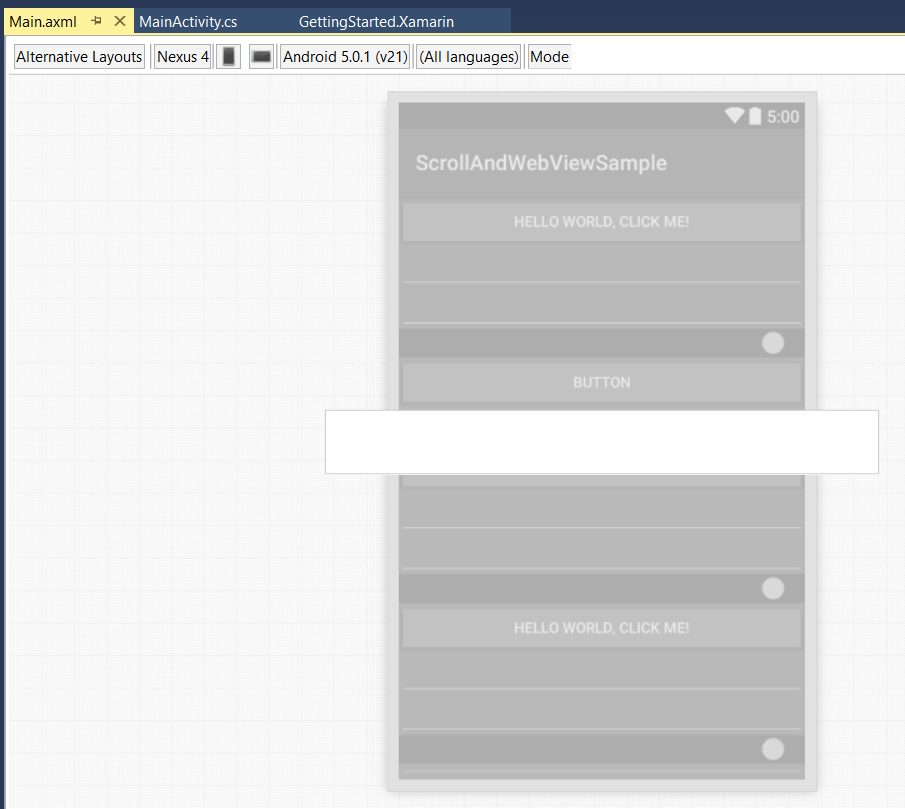


If you observe carefully, there is a button (and few more control underneath) that gets cutoff and we cannot really get to it. To solve this issue, we would be using a ScrollView.

If you are familiar with ScrollView on WindowsPhone controls, your natural instinct would be wrap all these controls in a ScrollView, something like –

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent">  <Button  android:id="@+id/MyButton"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:text="Navigate to WebView" />  <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText1" />  <EditText  android:inputType="textMultiLine"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText2" />  <Switch  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/switch1" />  <Button  android:text="Button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/button1" />  <RadioButton  android:text="RadioButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/radioButton1" />  <Button  android:id="@+id/MyButton"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:text="@string/Hello" />  <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText4" />  <EditText  android:inputType="textMultiLine"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText3" />  <Switch  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/switch2" />  <Button  android:id="@+id/MyButton"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:text="@string/Hello" />  <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText6" />  <EditText  android:inputType="textMultiLine"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText5" />  <Switch  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/switch3" />  <Button  android:text="Button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/button3" />  <RadioButton  android:text="RadioButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/radioButton3" />  <Button  android:text="Button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/button3" />  <RadioButton  android:text="RadioButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/radioButton3" />  </ScrollView> |

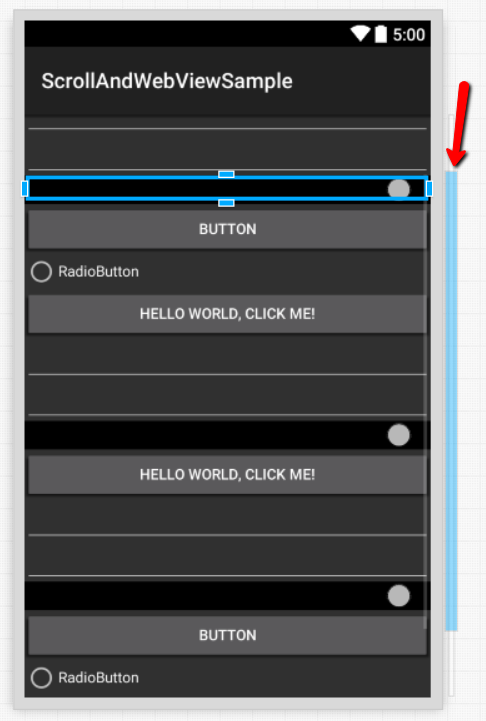
The moment you do this and switch to the Xamarin.Android designer, you will see a blocked view like below –



Even if you were to run this application in its current state, the application would crash. The reason for this behavior is ScrollView in android can only hold one direct child, and cannot hold bunch of controls like we have above. To solve this issue, we would have to wrap all our controls in any standard layouts like LinearLayout, TableLayout or RelativeLayout. Let’s wrap out sample in LinearLayout –

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent">  <LinearLayout  android:orientation="vertical"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent">  <Button  android:id="@+id/MyButton"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:text="Navigate to WebView" />  <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText1" />  <EditText  android:inputType="textMultiLine"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText2" />  <Switch  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/switch1" />  <Button  android:text="Button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/button1" />  <RadioButton  android:text="RadioButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/radioButton1" />  <Button  android:id="@+id/MyButton"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:text="@string/Hello" />  <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText4" />  <EditText  android:inputType="textMultiLine"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText3" />  <Switch  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/switch2" />  <Button  android:id="@+id/MyButton"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:text="@string/Hello" />  <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText6" />  <EditText  android:inputType="textMultiLine"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/editText5" />  <Switch  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/switch3" />  <Button  android:text="Button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/button3" />  <RadioButton  android:text="RadioButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/radioButton3" />  <Button  android:text="Button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/button3" />  <RadioButton  android:text="RadioButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/radioButton3" />  </LinearLayout>  </ScrollView> |

And when you go back to the designer now, you will start seeing a ScrollBar in the designer, which lets you scroll through the views. And a similar scroll bar would appear even when the application is run



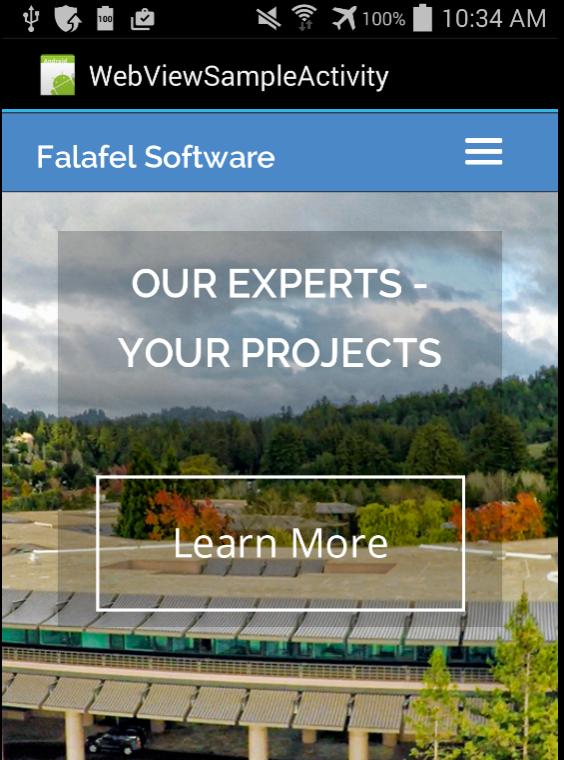
# WebView

WebView in Android is used to show a Web Page either from an external URL or from a HTML block. Let’s see it in action. To get started with it, let’s add a WebView to your AXML file –

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent"  android:minWidth="25px"  android:minHeight="25px">  <WebView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/webView1" />  </LinearLayout> |

And in our Activity file, after we find the WebView using the FindViewById, we will just be calling LoadUrl method –

|  |
| --- |
| using Android.App;  using Android.OS;  using Android.Webkit;  namespace ScrollAndWebViewSample  {  [Activity(Label = "WebViewSampleActivity")]  public class WebViewSampleActivity : Activity  {  protected override void OnCreate(Bundle bundle)  {  base.OnCreate(bundle);  SetContentView(Resource.Layout.webviewsample);  var webView = FindViewById<WebView>(Resource.Id.webView1);  webView.LoadUrl("http://falafel.com");  }  }  } |



That’s it for today, tomorrow, we will explore Tabbed Layouts in Xamarin.Android.