WebView控件:

[Activity(Label = "No5测试App", Icon = "@drawable/icon")]

public class MainActivity : Activity

{

WebView webviewMain;

protected override void OnCreate(Bundle bundle)

{

base.OnCreate(bundle);

//隐藏标题栏(须放在SetContentView函数之前)

this.RequestWindowFeature(WindowFeatures.NoTitle);

//设置全屏

this.Window.SetFlags(WindowManagerFlags.Fullscreen, WindowManagerFlags.Fullscreen);

// Set our view from the "main" layout resource

SetContentView(Resource.Layout.Main);

//获取web视图控件

webviewMain = FindViewById<WebView>(Resource.Id.webViewMain);

//启动js

webviewMain.Settings.JavaScriptEnabled = true;

//载入网址

webviewMain.LoadUrl("http://www.no5.com.cn/");

//设置支持缩放(前提是网页自身支持缩放)

webviewMain.Settings.SetSupportZoom(true);

webviewMain.Settings.BuiltInZoomControls = true;

//显示缩放控件(放大/缩小按钮)

webviewMain.Settings.DisplayZoomControls = false;

//自适应屏幕

webviewMain.Settings.SetLayoutAlgorithm(WebSettings.LayoutAlgorithm.SingleColumn);

webviewMain.Settings.LoadWithOverviewMode = true;

var btn = FindViewById<Button>(Resource.Id.BtnOpenNo5);

btn.Click += Btn\_Click;

webviewMain.SetWebViewClient(new CustWebViewClient());

}

private void Btn\_Click(object sender, System.EventArgs e)

{

webviewMain.LoadUrl("https://www.baidu.com/");

}

/// <summary>

/// 巢状class,继承webviewclient

/// </summary>

private class CustWebViewClient : WebViewClient

{

/// <summary>

/// 响应超链接事件，通过改变Load加载方式，使在WebView控件中的连接在当前视图中打开，而不是新窗口中打开

/// </summary>

/// <param name="view"></param>

/// <param name="url"></param>

/// <returns></returns>

public override bool ShouldOverrideUrlLoading(WebView view, string url)

{

view.LoadUrl(url);

return true;

}

}

/// <summary>

/// 重写使返回键可以返回上一页,并非离开app

/// </summary>

/// <param name="keyCode"></param>

/// <param name="e"></param>

/// <returns></returns>

public override bool OnKeyDown(Android.Views.Keycode keyCode, Android.Views.KeyEvent e)

{

if (keyCode == Keycode.Back && webviewMain.CanGoBack())

{

webviewMain.GoBack();

return true;

}

return base.OnKeyDown(keyCode, e);

}

}

使用权限设置: UsesPermissionAttribute

使用WebView,，因为用处到了网络，所以我们必须在AndroidManifset.xml文件中进行权限设置<uses-permission android:name="android.permission.INTERNET" />

使用或跳转某个页面:SetContentView(Resource.Layout.Main);

**Assets：** 放置在Assets资料夹中的档案, 将会一起被封装进Android的封装档中(建置动作要设定为"AndroidAsset"). 之后便可以透过如下的陈述式来存取Assets的资源。

[view source](http://www.dotblogs.com.tw/taihon64/archive/2013/08/26/115518.aspx#viewSource)

[print](http://www.dotblogs.com.tw/taihon64/archive/2013/08/26/115518.aspx#printSource)[?](http://www.dotblogs.com.tw/taihon64/archive/2013/08/26/115518.aspx#about)

|  |  |  |
| --- | --- | --- |
| 1 | public class ReadAsset : Activity | |
| 2 |  |

|  |  |
| --- | --- |
| 3 | { |
| 4 |  | |

|  |  |  |
| --- | --- | --- |
| 5 | protected override void OnCreate (Bundle bundle) { | |
| 6 |  |

|  |  |  |
| --- | --- | --- |
| 7 | base.OnCreate (bundle); | |
| 8 |  |

|  |  |
| --- | --- |
| 9 | InputStream input = Assets.Open ("my\_asset.txt");}} |

**Resources：**包含Drawable, Layout以及Values资料夹. Drawable用来放置图片. 依照装置的解析度不同, 还可以新增drawable-hdpi, drawable-mdpi, drawable-ldpi等资料夹来存放不同解析度的档案. Layout资料夹则是存放使用者界面档(副档名为.axml). 而Value资料夹则是可以存放不同型別的XML对应档, 例如styles.xml, colors.xml… 针对Resources底下的档案, 建置动作请设定为”AndroidResource”

若您开启预设的Main.axml, 会看到如同底下的XML描述



* LinearLayout: 主要的页面框架, 以垂直或水平的方式排列页面上的物件, 相当于Silverlight 中的stack panel
* @+id/[物件名称]: 告诉Android parser, 为物件建立一个resource id
* @string/[名称]: 在String.xml中建立一个字串资源, 后续可供Resource类別存取.

上述的@string则会对应到资料夹Resources\Values\String.xml

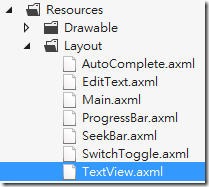


* 名称Hello对应到UI中Button的Text属性
* 名称ApplicationName对应到专案属性中的应用程式名称
* 名称Hello2为自行定义的字串资源.

有了以上的基本概念后, 接下来我们来介绍Android的基本控制项。

**TextView**

1. 开启Lab03-BasicControls 专案并开启Layout资料夹下的TextView.axml



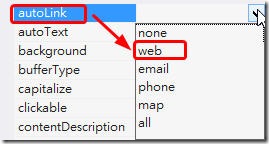
2. 从左边的工具列将TextView拖放到画面中, 双击TextView并编辑文字



3. 接着拖拉一个TextView, 并在右边的属性视窗设定textcolor为#2A3748, textsize为24dip



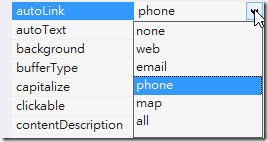
4. 再拖拉一个TextView并输入文字, 包含一个超链接. 在属性中将autolink的属性值改为web.



结果如下:链接文字会自动变成超链接.



5. 最后拖拉一个TextView并输入文字, 包含超过5位数的数字, 在属性中将autolink的属性值改为phone



结果如下: 数字被更改为超链接

http://img.blog.csdn.net/20140303112723562?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQveGFtYXJpbg==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast

6. 开启TextViewScreen.cs 並在OnCreate 事件中载入Layout中的TextView

SetContentView(Resource.Layout.TextView);

7. 执行专案并检视及操作有链接的TextView內容.

添加app需要的权限

