Developed your First Xamarin.Forms App

Prepared by Eng Soon Cheah, Microsoft MVP

Objective

Build your first mobile app for iOS,Android, and Windows with Xamarin.Forms

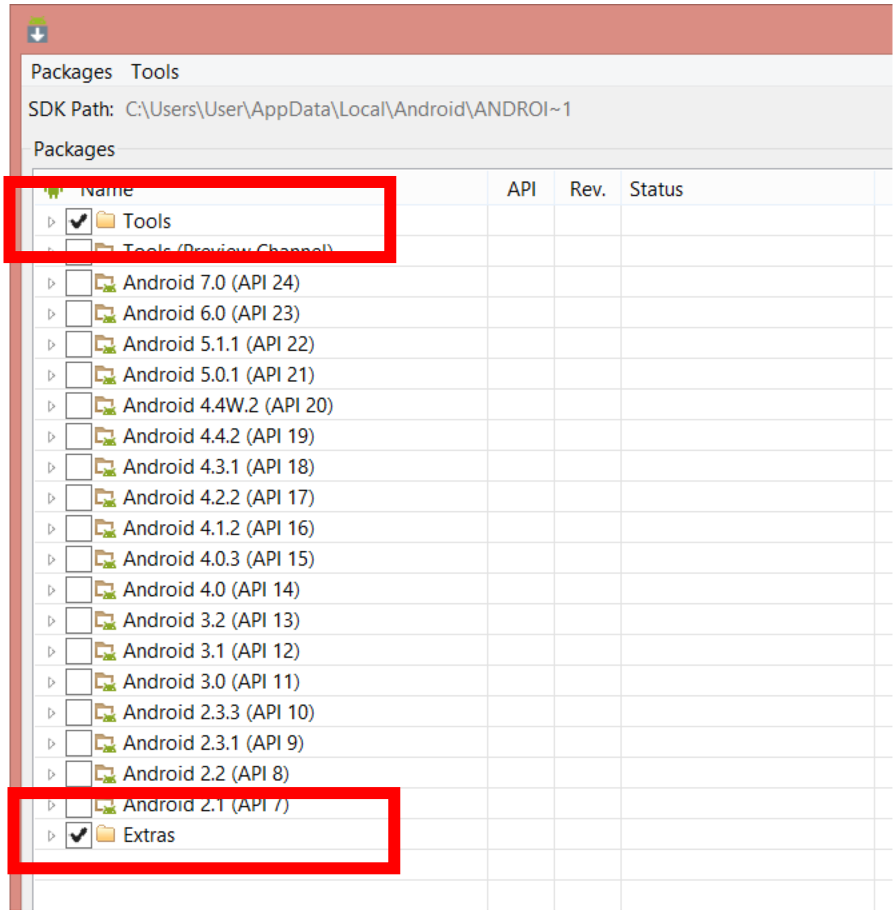
Requirements

1. Install Visual Studio 2015 Enterprise

2. Install Visual Studio Emulator for Android

3. Install Windows Standalone SDK for Windows 10 & Enable Developer Mode

4. Install Android SDK Manager for Tools and Extras



SDK Path: C:\Users\User\AppData\Local\Android\ANDROI~1

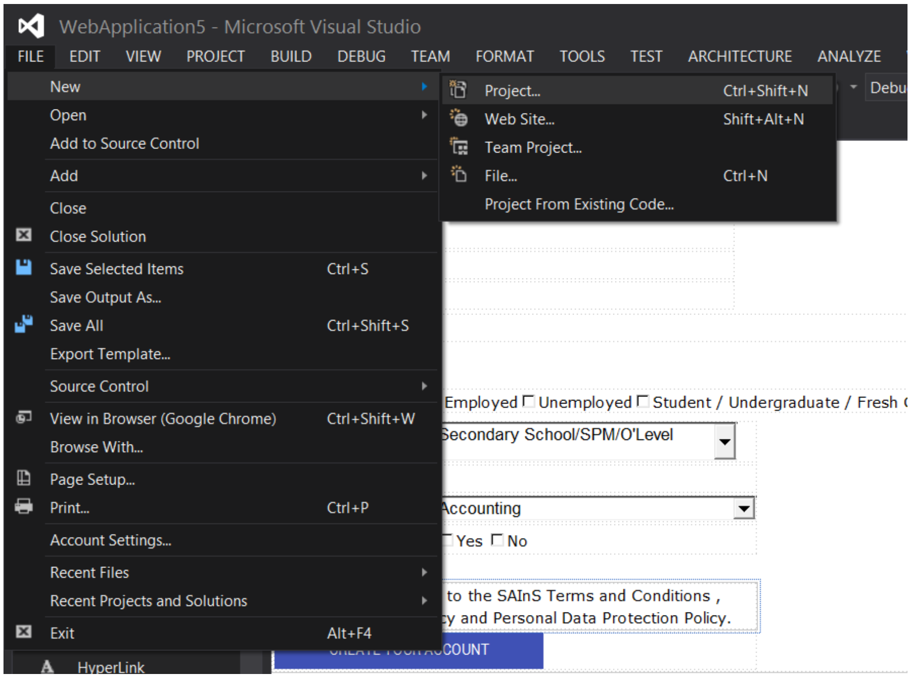
Packages

Name	API	Rev.	Status
☑ Tools			
□ Tools (Preview Channel)			
□ Android 7.0 (API 24)			
□ Android 6.0 (API 23)			
□ Android 5.1.1 (API 22)			
□ Android 5.0.1 (API 21)			
□ Android 4.4W.2 (API 20)			
□ Android 4.4.2 (API 19)			
□ Android 4.3.1 (API 18)			
□ Android 4.2.2 (API 17)			
□ Android 4.1.2 (API 16)			
□ Android 4.0.3 (API 15)			
□ Android 4.0 (API 14)			
□ Android 3.2 (API 13)			
□ Android 3.1 (API 12)			
□ Android 3.0 (API 11)			
□ Android 2.3.3 (API 10)			
□ Android 2.3.1 (API 9)			
□ Android 2.2 (API 8)			
□ Android 2.1 (API 7)			
☑ Extras			

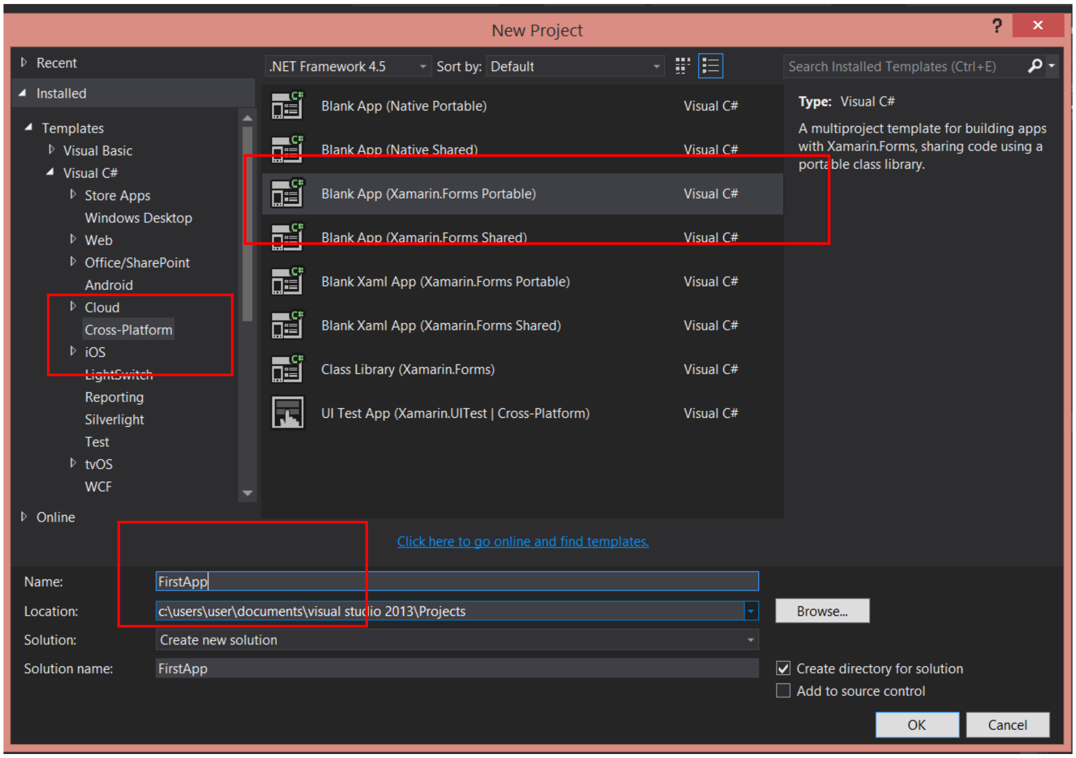
Content

Chapter 1: Create Your First Project

1. Open your Visual Studio 2015/ Visual Studio Enterprise.
2. **Go to File > New > Project**



3. Select “**Cross-Platform**” > Select “ **Blank App (Xamarin.Forms Portable)** “& Name Your Project (App Name)


New Project
Recent
Installed
Templates
Visual Basic
Visual C#
Store Apps
Windows Desktop
Web
Office/SharePoint
Android
Cloud
Cross-Platform
iOS
LightSwitch
Reporting
Silverlight
Test
tvOS
WCF
Online
.NET Framework 4.5
Sort by: Default
Search Installed Templates (Ctrl+E)
Blank App (Native Portable) Visual C#
Blank App (Native Shared) Visual C#
Blank App (Xamarin.Forms Portable) Visual C#
Blank App (Xamarin.Forms Shared) Visual C#
Blank Xaml App (Xamarin.Forms Portable) Visual C#
Blank Xaml App (Xamarin.Forms Shared) Visual C#
Class Library (Xamarin.Forms) Visual C#
UI Test App (Xamarin.UITest | Cross-Platform) Visual C#
Click here to go online and find templates.
Browse...
OK
Cancel

Type: Visual C#

A multiproject template for building apps with Xamarin.Forms, sharing code using a portable class library.

Name: FirstApp

Location: c:\users\user\documents\visual studio 2013\Projects

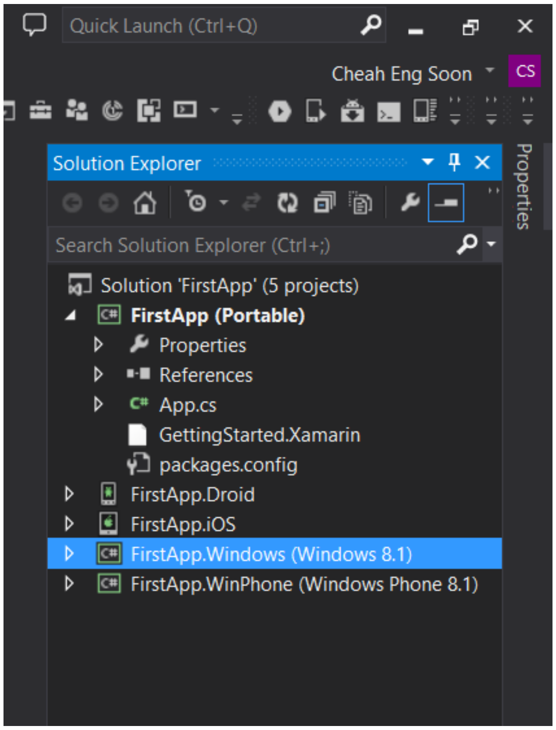
Solution: Create new solution

Solution name: FirstApp

☑ Create directory for solution

□ Add to source control

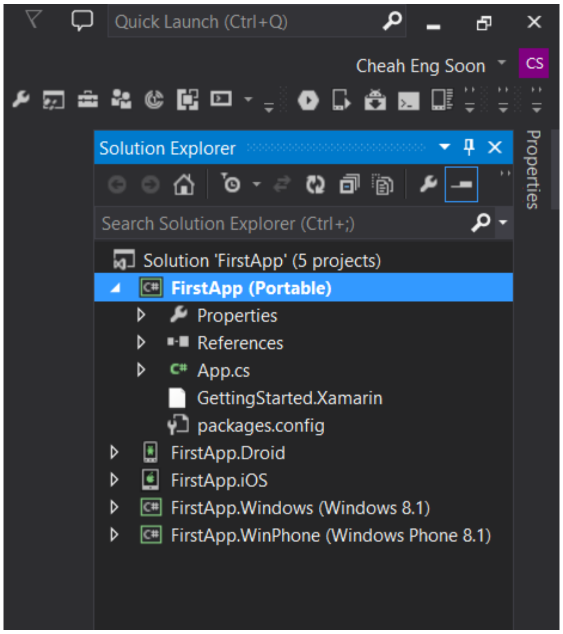
4. Your Xamarin.Forms App Solutions Successful Created.



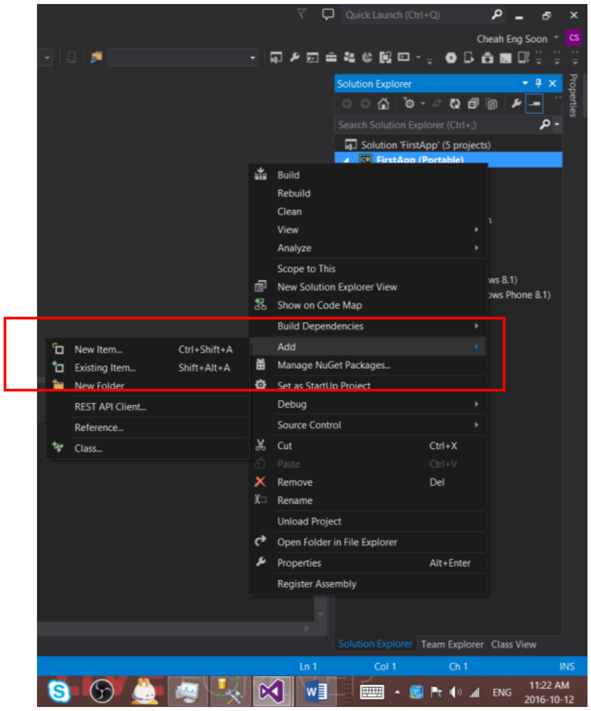
```
Solution 'FirstApp' (5 projects)
  FirstApp (Portable)
    Properties
    References
    App.cs
    GettingStarted.Xamarin
    packages.config
  FirstApp.Droid
  FirstApp.iOS
  FirstApp.Windows (Windows 8.1)
  FirstApp.WinPhone (Windows Phone 8.1)
```

* If you use Windows 10 & install Windows 10 SDK, You will saw the Universal Windows Platform (UWP) App.

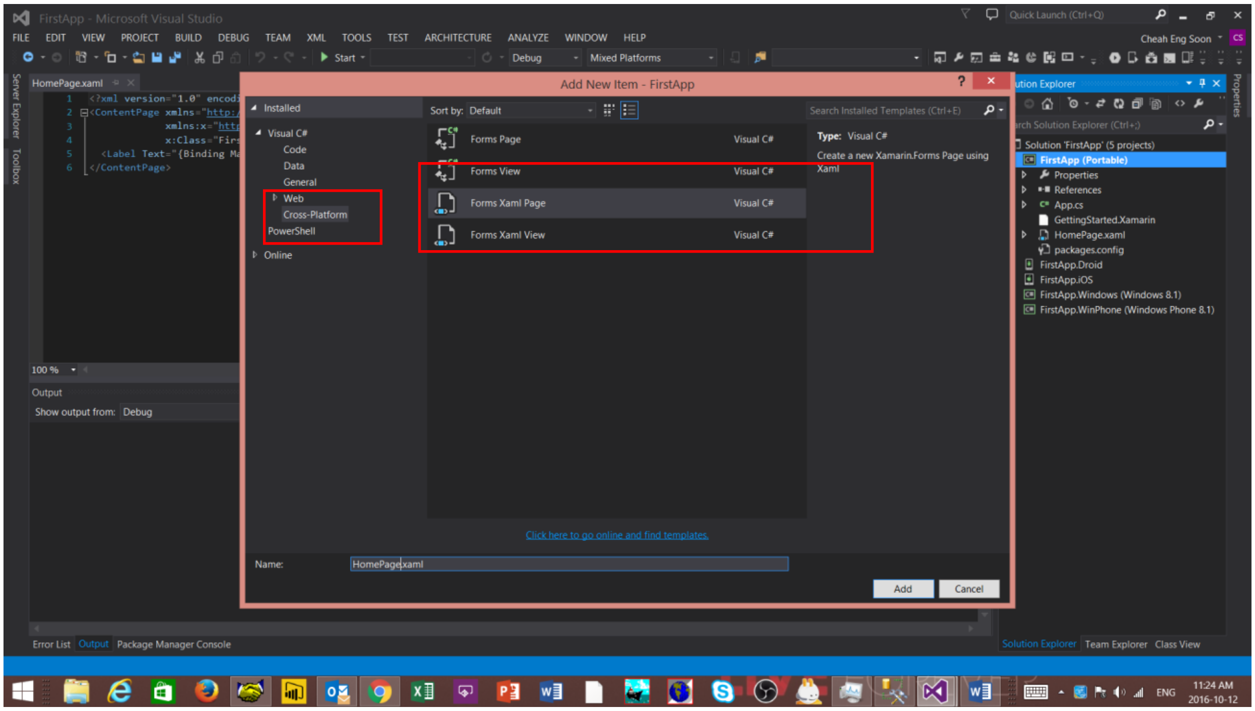
5. Select the Solution "FirstApp(Portable)"



6. Right- Click the Solution "**FirstApp(Portable)**" > Select "**Add**" > Select "**New Item**"



7. Select “**Cross-Platform**” > Select “**Forms Xaml Page**” > Name the Page “**HomePage.xaml**”



Add New Item - FirstApp

- Installed
  - Visual C#
    - Code
    - Data
    - General
    - Web
    - Cross-Platform
  - PowerShell
- Online

Sort by: Default

Forms Page	Visual C#
Forms View	Visual C#
Forms Xaml Page	Visual C#
Forms Xaml View	Visual C#

Type: Visual C#

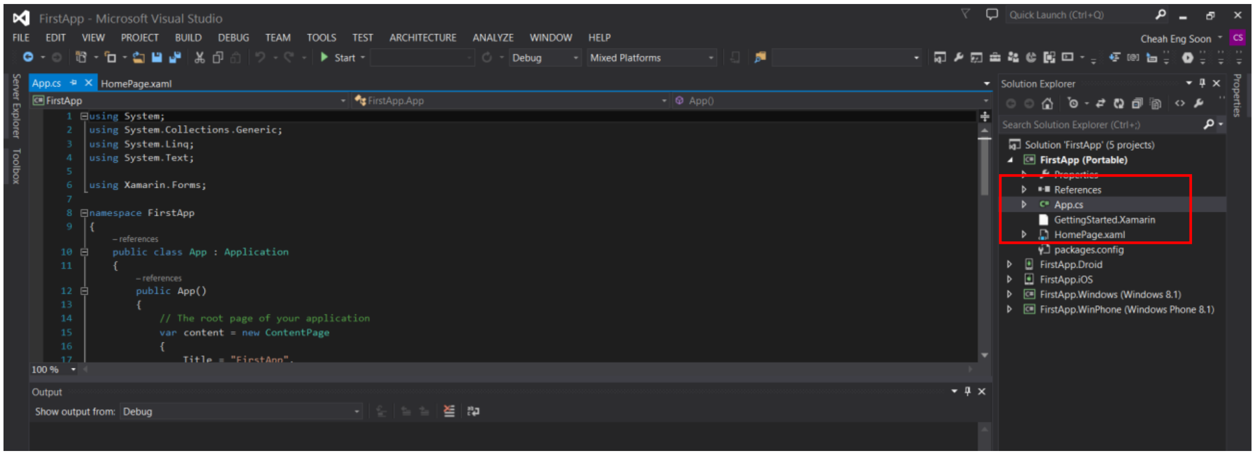
Create a new Xamarin.Forms Page using Xaml

Name: HomePage.xaml

```
<?xml version="1.0" encodi
<ContentPage xmlns="http:/
             xmlns:x="http
             x:Class="Firs
  <Label Text="{Binding Ma
</ContentPage>
```

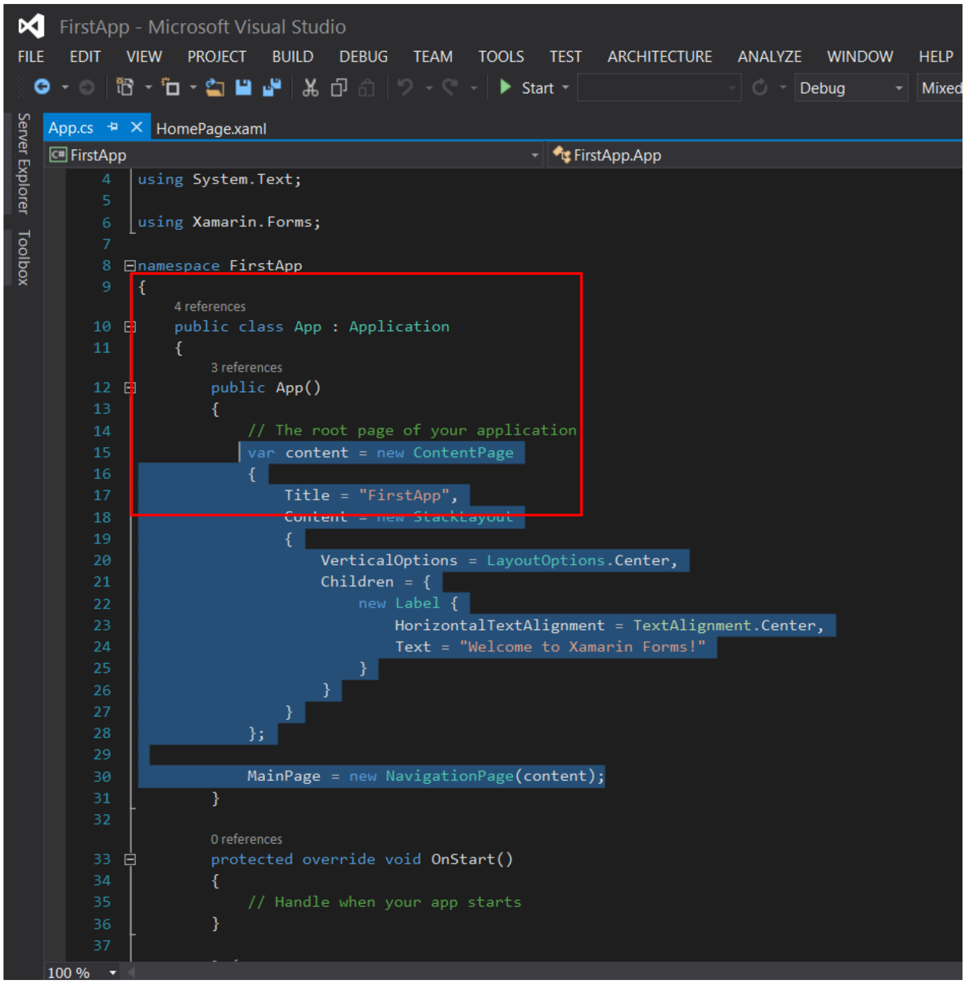
FirstApp - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM XML TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Start
Debug
Mixed Platforms
Quick Launch (Ctrl+Q)
Cheah Eng Soon
HomePage.xaml
Search Installed Templates (Ctrl+E)
Click here to go online and find templates.
Add
Cancel
Solution Explorer
Solution 'FirstApp' (5 projects)
FirstApp (Portable)
Properties
References
App.cs
GettingStarted.Xamarin
HomePage.xaml
packages.config
FirstApp.Droid
FirstApp.iOS
FirstApp.Windows (Windows 8.1)
FirstApp.WinPhone (Windows Phone 8.1)
100 %
Output
Show output from: Debug
Error List Output Package Manager Console
Solution Explorer Team Explorer Class View
ENG
11:24 AM
2016-10-12

8. Go to “**FirstApp (Portable)**” > Select “**App.cs**”



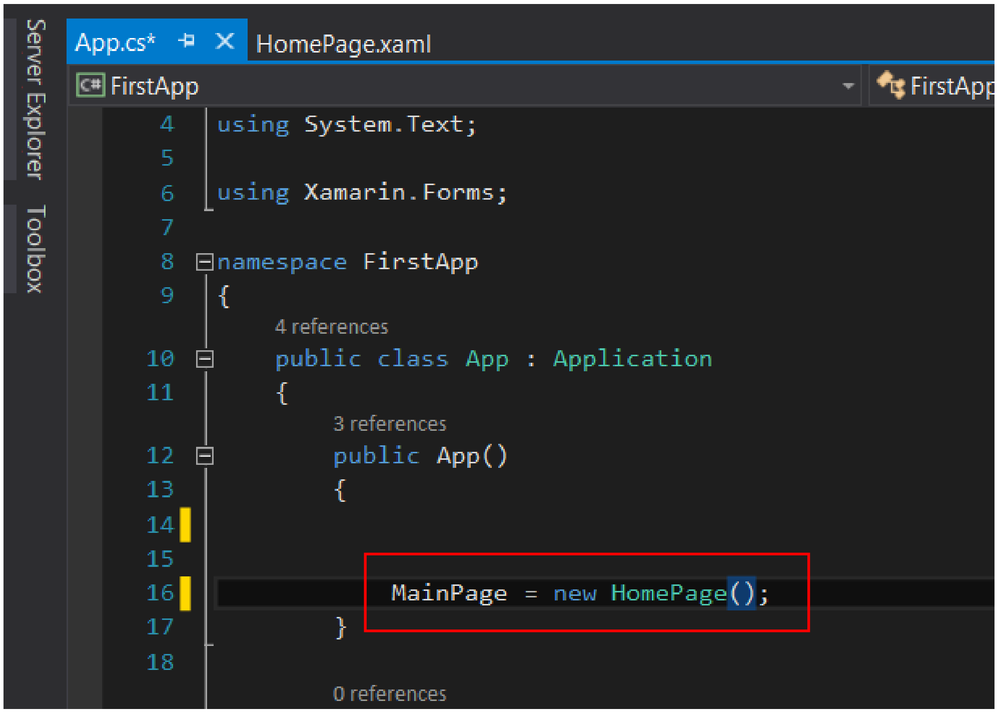
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Xamarin.Forms;

namespace FirstApp
{
    public class App : Application
    {
        public App()
        {
            // The root page of your application
            var content = new ContentPage
            {
                Title = "FirstApp",
```

9. Highlight and Remove the Source Code in **App()**


FirstApp - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Start
Debug
Mixed
Server Explorer
Toolbox
App.cs
HomePage.xaml
FirstApp
FirstApp.App
100 %

```
using System.Text;

using Xamarin.Forms;

namespace FirstApp
{
    4 references
    public class App : Application
    {
        3 references
        public App()
        {
            // The root page of your application
            var content = new ContentPage
            {
                Title = "FirstApp",
                Content = new StackLayout
                {
                    VerticalOptions = LayoutOptions.Center,
                    Children = {
                        new Label {
                            HorizontalTextAlignment = TextAlignment.Center,
                            Text = "Welcome to Xamarin Forms!"
                        }
                    }
                }
            };

            MainPage = new NavigationPage(content);
        }

        0 references
        protected override void OnStart()
        {
            // Handle when your app starts
        }
```

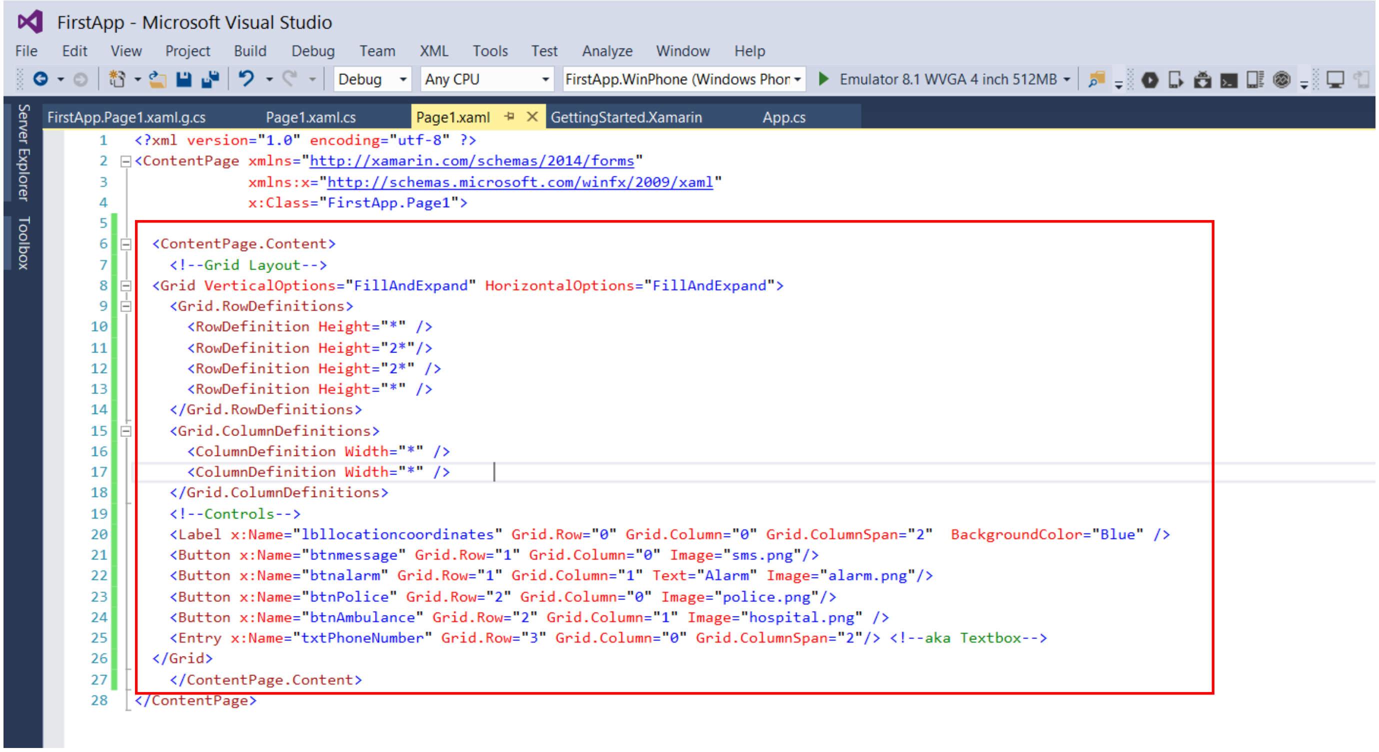
10. And Replace with "MainPage = new HomePage();"



```
using System.Text;

using Xamarin.Forms;

namespace FirstApp
{
        4 references
        public class App : Application
        {
            3 references
            public App()
            {


                MainPage = new HomePage();
            }

            0 references
```

Chapter 2: Layout and Control

1. Go to “**HomePage.xaml**” add Layout and Control



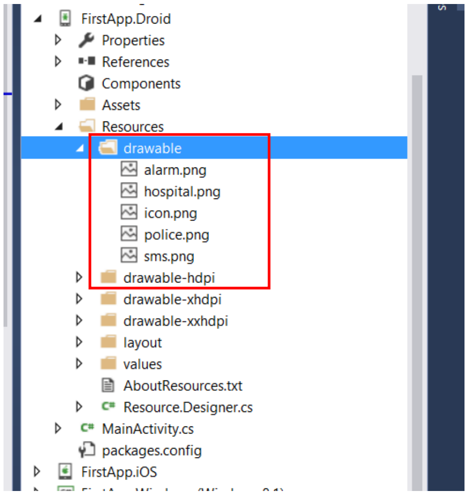
```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="FirstApp.Page1">

  <ContentPage.Content>
    <!--Grid Layout-->
  <Grid VerticalOptions="FillAndExpand" HorizontalOptions="FillAndExpand">
    <Grid.RowDefinitions>
      <RowDefinition Height="*" />
      <RowDefinition Height="2*"/>
      <RowDefinition Height="2*" />
      <RowDefinition Height="*" />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="*" />
      <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>
    <!--Controls-->
    <Label x:Name="lbllocationcoordinates" Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2"  BackgroundColor="Blue" />
    <Button x:Name="btnmessage" Grid.Row="1" Grid.Column="0" Image="sms.png"/>
    <Button x:Name="btnalarm" Grid.Row="1" Grid.Column="1" Text="Alarm" Image="alarm.png"/>
    <Button x:Name="btnPolice" Grid.Row="2" Grid.Column="0" Image="police.png"/>
    <Button x:Name="btnAmbulance" Grid.Row="2" Grid.Column="1" Image="hospital.png" />
    <Entry x:Name="txtPhoneNumber" Grid.Row="3" Grid.Column="0" Grid.ColumnSpan="2"/> <!--aka Textbox-->
  </Grid>
    </ContentPage.Content>
</ContentPage>
```

2. Add Image to each Platform (iOS,Android,Windows/UWP)

- iOS – Add Image to “ **Resources**” Folder , **Build Action: BundleResource** in Properties Tab.
- Android – Add Image to “**Resources/drawable**” Folder , **Build Action:AndroidResource**.
- Windows /UWP – Place images in root directory , **Build Action: Content**.

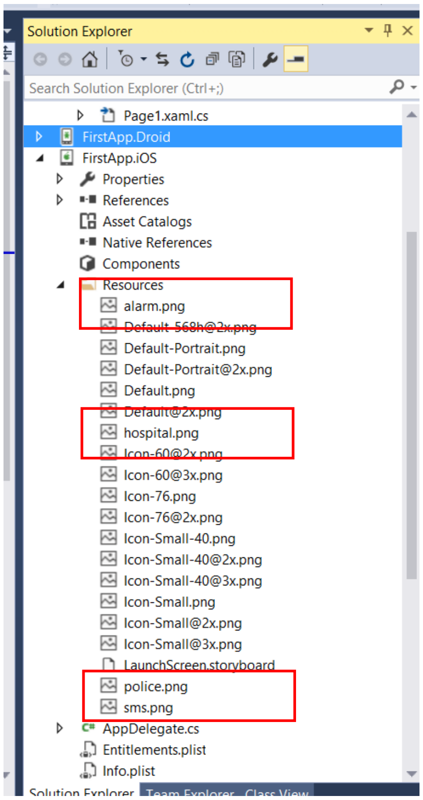
Download the image and MP3 file from:

https://github.com/cheahengsoon/FirstXamarinFormsApp



```
FirstApp.Droid
  Properties
  References
  Components
  Assets
  Resources
    drawable
      alarm.png
      hospital.png
      icon.png
      police.png
      sms.png
    drawable-hdpi
    drawable-xhdpi
    drawable-xxhdpi
    layout
    values
    AboutResources.txt
    Resource.Designer.cs
  MainActivity.cs
  packages.config
FirstApp.iOS
```

Android Platform



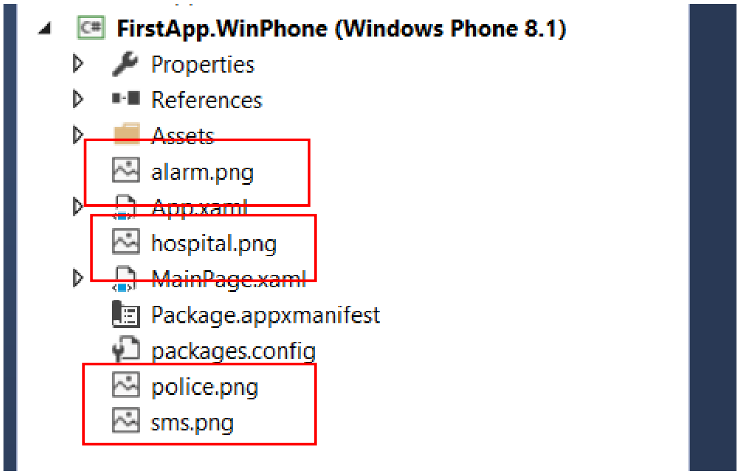
Solution Explorer

Search Solution Explorer (Ctrl+;)

```
    Page1.xaml.cs
FirstApp.Droid
FirstApp.iOS
  Properties
  References
  Asset Catalogs
  Native References
  Components
  Resources
    alarm.png
    Default-568h@2x.png
    Default-Portrait.png
    Default-Portrait@2x.png
    Default.png
    Default@2x.png
    hospital.png
    Icon-60@2x.png
    Icon-60@3x.png
    Icon-76.png
    Icon-76@2x.png
    Icon-Small-40.png
    Icon-Small-40@2x.png
    Icon-Small-40@3x.png
    Icon-Small.png
    Icon-Small@2x.png
    Icon-Small@3x.png
    LaunchScreen.storyboard
    police.png
    sms.png
  AppDelegate.cs
  Entitlements.plist
  Info.plist
```

Solution Explorer   Team Explorer   Class View

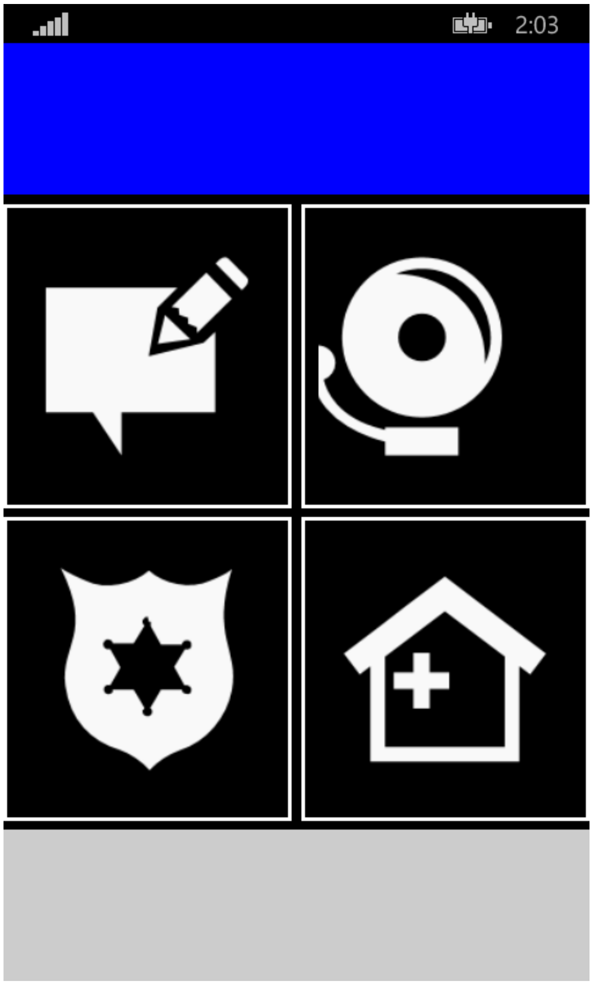
iOS Platform



```
◢ FirstApp.WinPhone (Windows Phone 8.1)
    ▷ Properties
    ▷ References
    ▷ Assets
      alarm.png
    ▷ App.xaml
      hospital.png
    ▷ MainPage.xaml
      Package.appxmanifest
      packages.config
      police.png
      sms.png
```

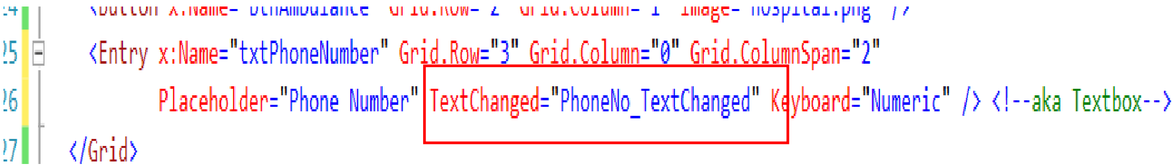
Windows Platform

Result:



Layout Done. But still not had any event or functions.

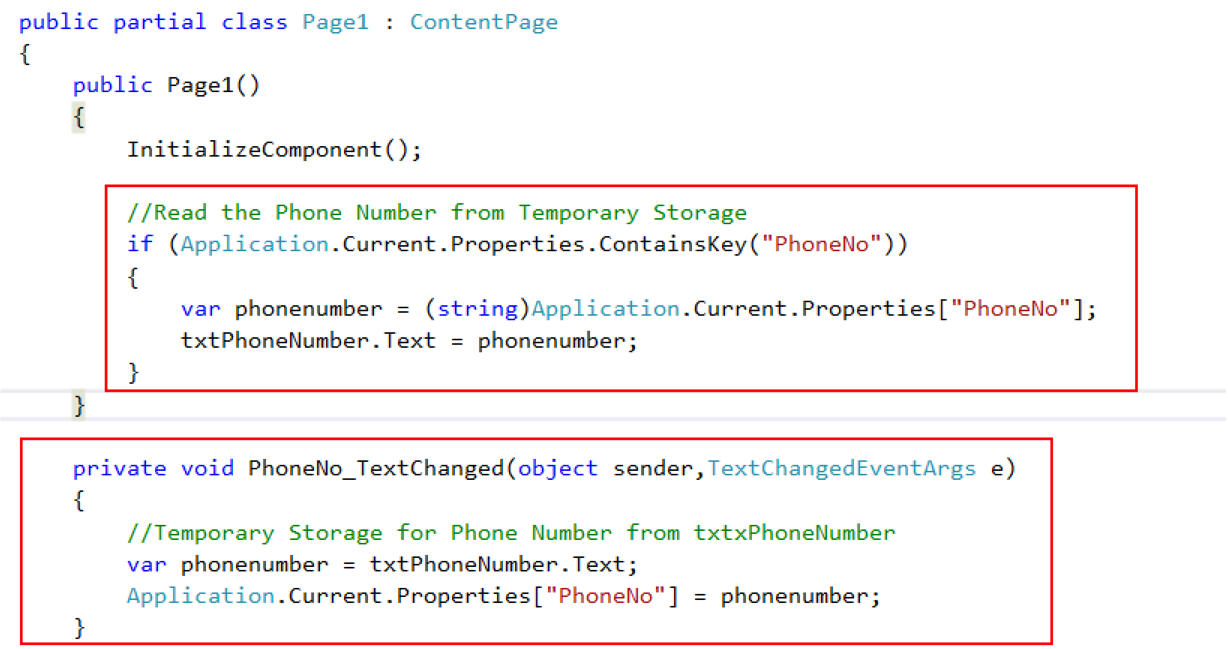
3. Add Event handler to the Entry for PhoneNumber.



```
    <Button x:Name="btnAmbulance" Grid.Row="2" Grid.Column="1" Image="hospital.png" />
    <Entry x:Name="txtPhoneNumber" Grid.Row="3" Grid.Column="0" Grid.ColumnSpan="2"
           Placeholder="Phone Number" TextChanged="PhoneNo_TextChanged" Keyboard="Numeric" /> <!--aka Textbox-->
  </Grid>
```

And

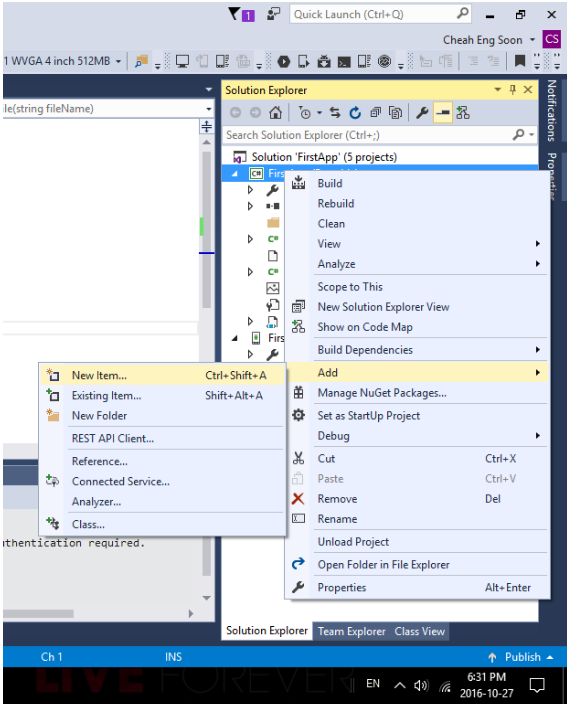
Go to HomePage.xaml.cs , add the C# Code



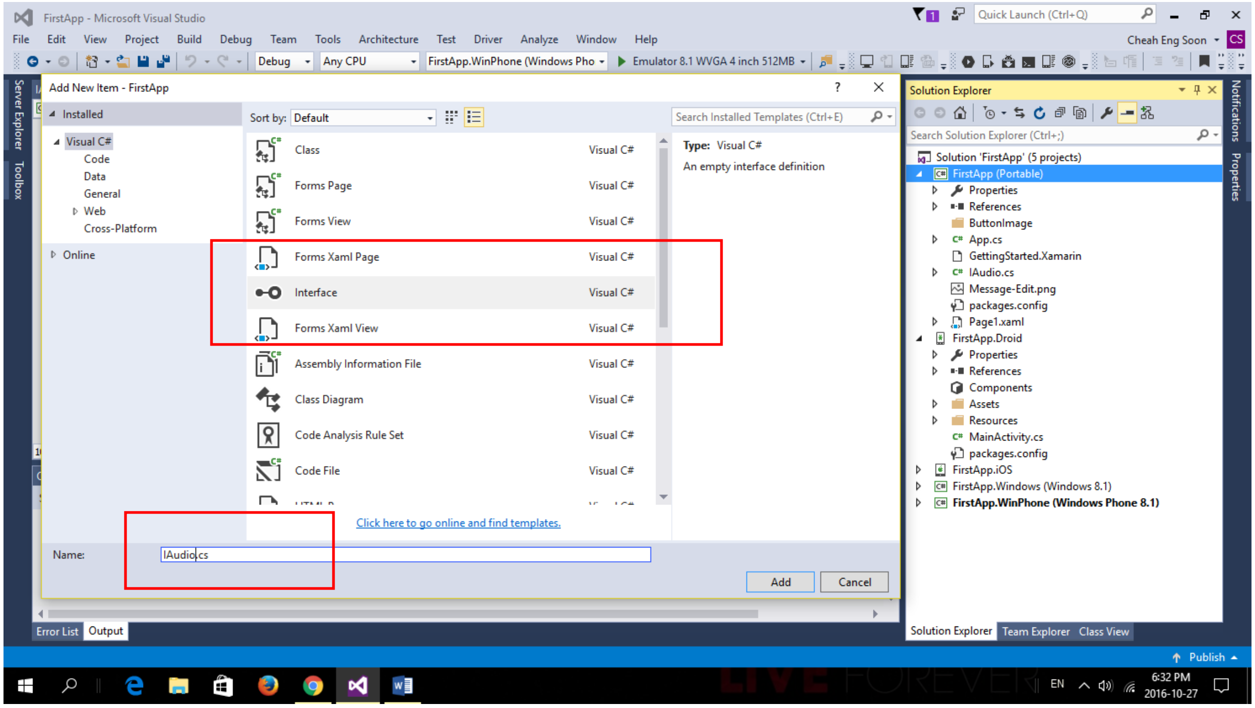
```
public partial class Page1 : ContentPage
{
    public Page1()
    {
        InitializeComponent();

        //Read the Phone Number from Temporary Storage
        if (Application.Current.Properties.ContainsKey("PhoneNo"))
        {
            var phonenumber = (string)Application.Current.Properties["PhoneNo"];
            txtPhoneNumber.Text = phonenumber;
        }
    }

    private void PhoneNo_TextChanged(object sender,TextChangedEventArgs e)
    {
        //Temporary Storage for Phone Number from txtxPhoneNumber
        var phonenumber = txtPhoneNumber.Text;
        Application.Current.Properties["PhoneNo"] = phonenumber;
    }
```

4. Play Audio for Alarm

1. Right Click “**FirstApp(Portable)**” Solution, select “**Add**”, and “**New item**”.



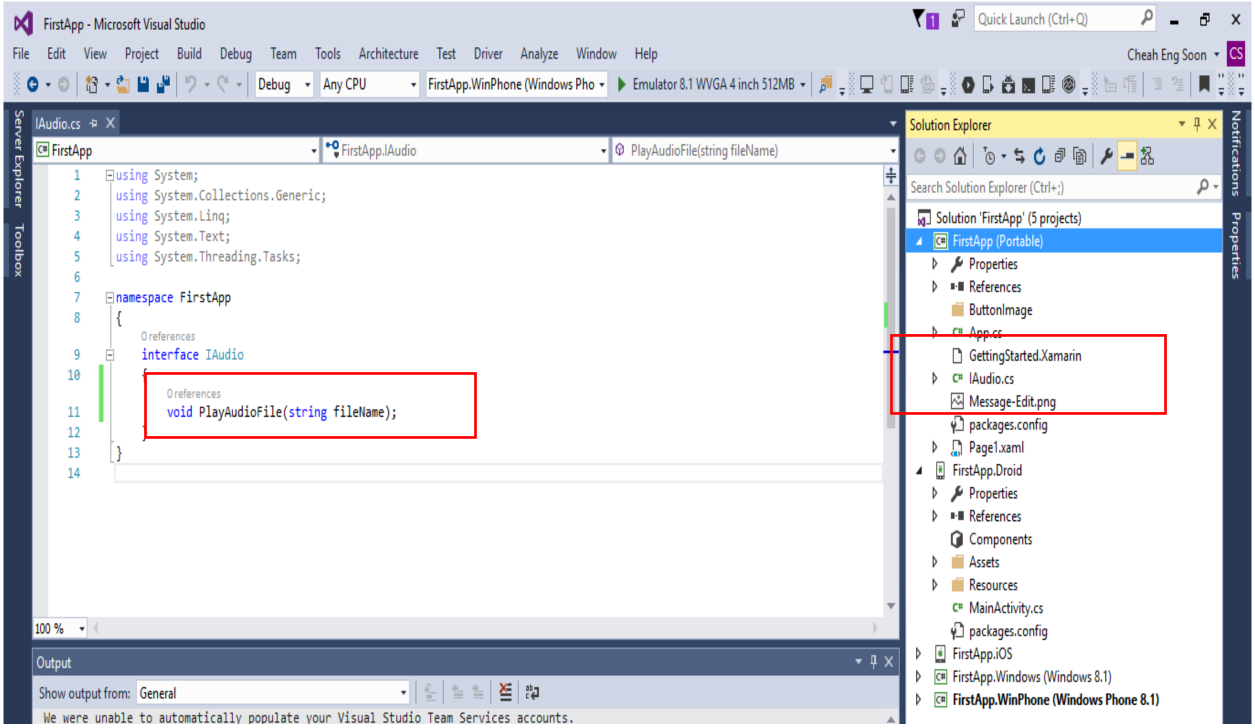
2. Select “**Interface**” and name the class as “**IAudio.cs**”



Name: IAudio.cs

3. “**IAudio.cs**” Success created, And Insert this line of code as below.

   void PlayAudioFile(string filename);



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace FirstApp
{
    0 references
    interface IAudio
    {
        0 references
        void PlayAudioFile(string fileName);
    }
}
```

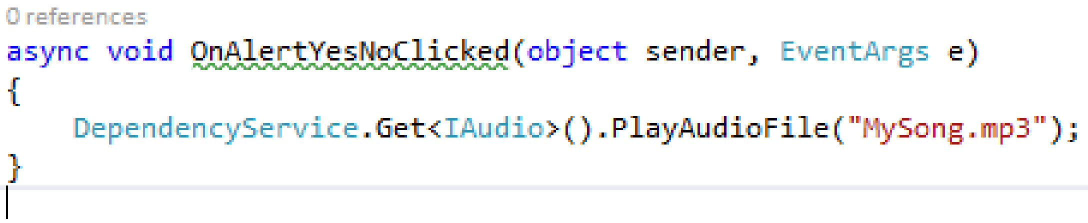
```
We were unable to automatically populate your Visual Studio Team Services accounts.
```

4. Go to “**HomePage.xaml**” add event handler for “**btnalarm**”

```
<Button x:Name="btnalarm" Grid.Row="1" Grid.Column="1" Text="Alarm" Image="alarm.png"
Clicked="OnAlertYesNoClicked" />
```

And

Add the backend function to play the alarm sound.



```
0 references
async void OnAlertYesNoClicked(object sender, EventArgs e)
{
    DependencyService.Get<IAudio>().PlayAudioFile("MySong.mp3");
}
```

5. Now add the “policeSiren.mp3” to iOS & Android.

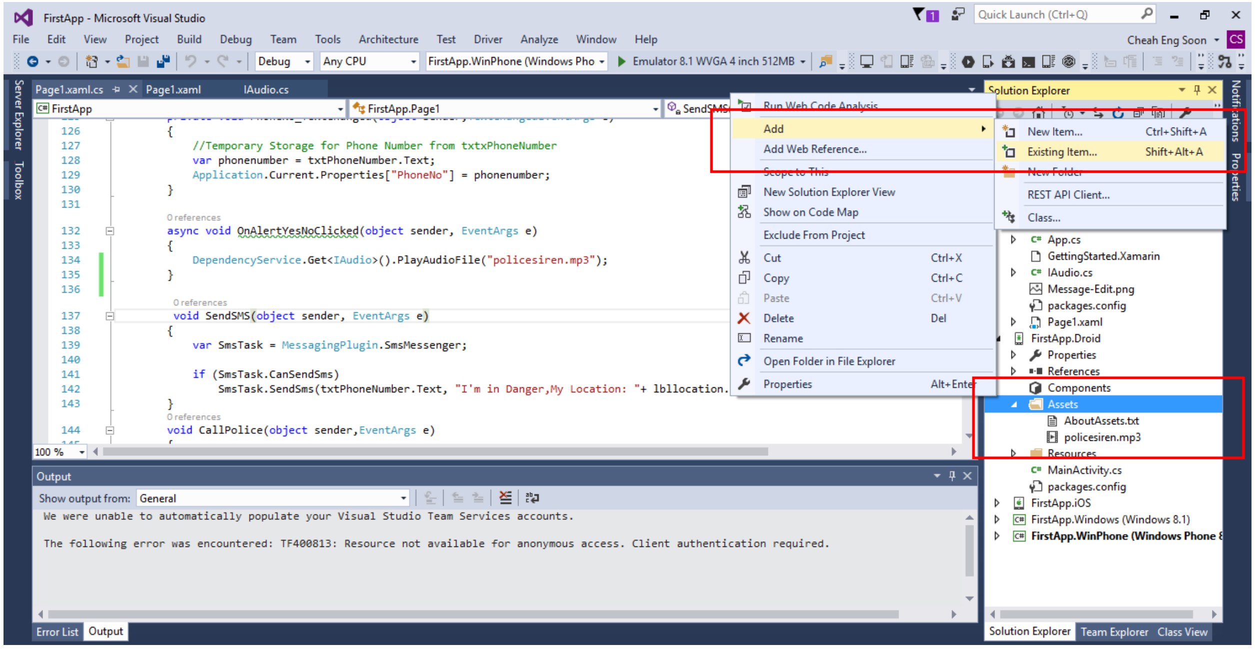
**iOS: file should be added to the Resources folder

**Android: file should be added to the Assets folder

For Android

==========

Go to “**FirstApp.Droid**”, Select “**Assets**” folder and select “**Add**” and “**Existing item**”. Add “**policesiren.mp3**”

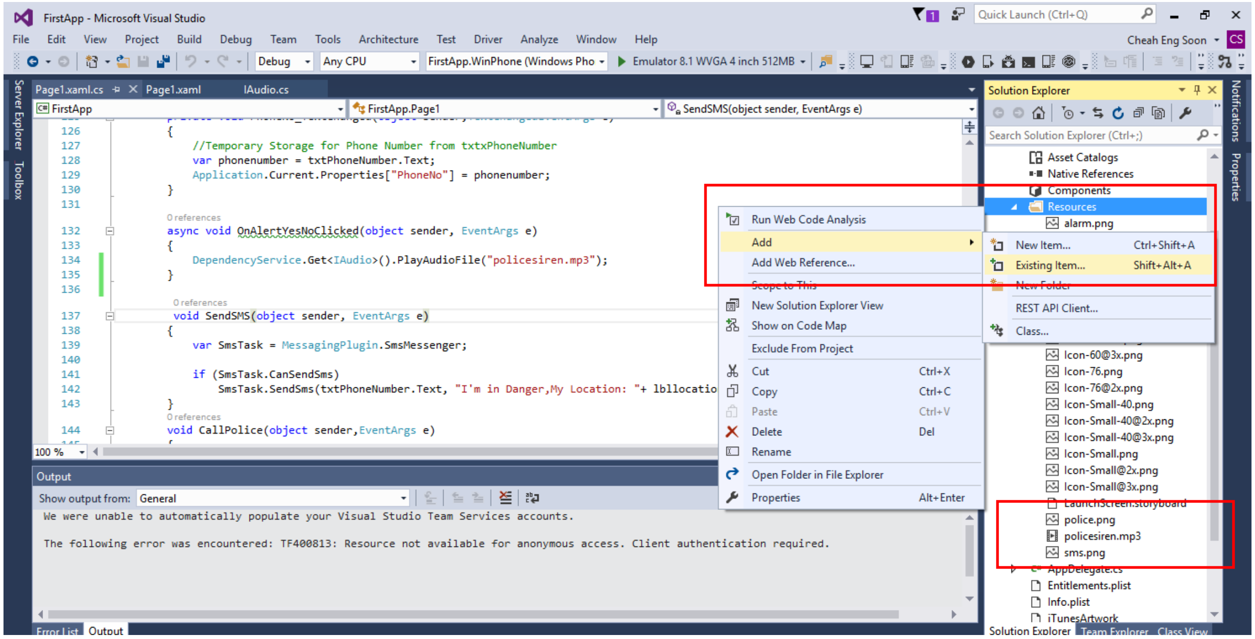


```
        {
            //Temporary Storage for Phone Number from txtxPhoneNumber
            var phonenumber = txtPhoneNumber.Text;
            Application.Current.Properties["PhoneNo"] = phonenumber;
        }

        0 references
        async void OnAlertYesNoClicked(object sender, EventArgs e)
        {
            DependencyService.Get<IAudio>().PlayAudioFile("policesiren.mp3");
        }

         0 references
         void SendSMS(object sender, EventArgs e)
        {
            var SmsTask = MessagingPlugin.SmsMessenger;

            if (SmsTask.CanSendSms)
                SmsTask.SendSms(txtPhoneNumber.Text, "I'm in Danger,My Location: "+ lbllocation.
        }
        0 references
        void CallPolice(object sender,EventArgs e)
```

```
We were unable to automatically populate your Visual Studio Team Services accounts.

The following error was encountered: TF400813: Resource not available for anonymous access. Client authentication required.
```

For iOS

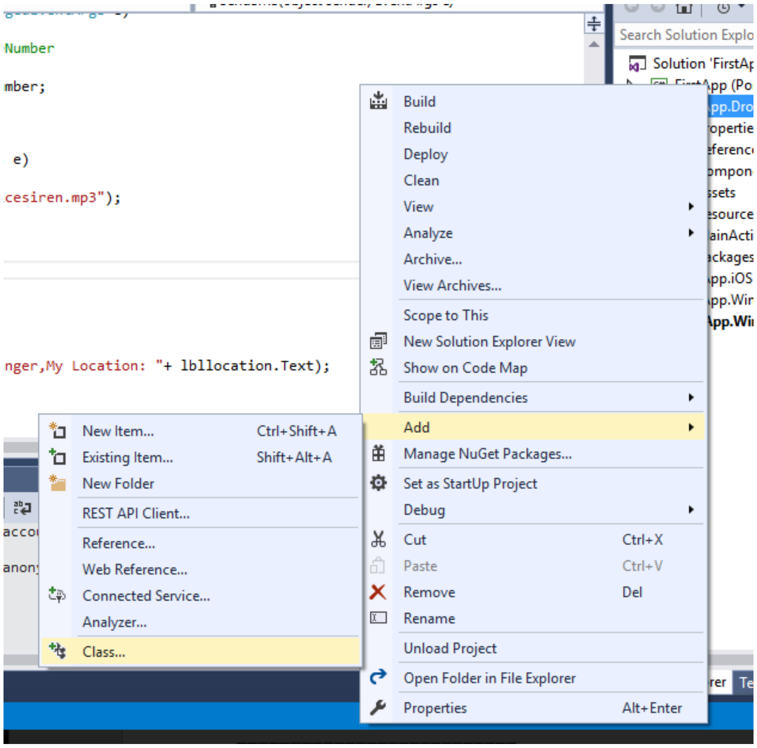
Go to “**FirstApp.iOS**”, Select “**Resources**” folder, select “**Add**” and “**Existing Item**”. Add “**policesiren.mp3**”

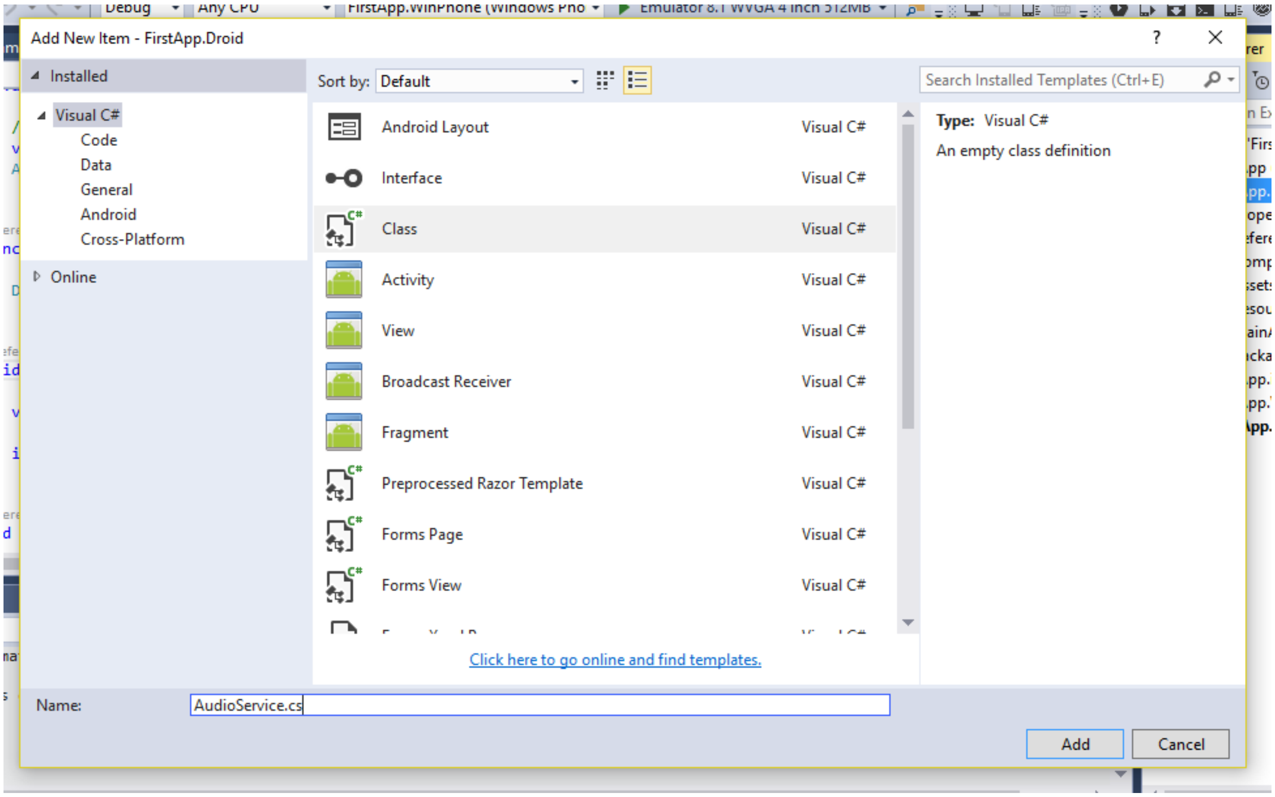


```
        {
            //Temporary Storage for Phone Number from txtxPhoneNumber
            var phonenumber = txtPhoneNumber.Text;
            Application.Current.Properties["PhoneNo"] = phonenumber;
        }

        0 references
        async void OnAlertYesNoClicked(object sender, EventArgs e)
        {
            DependencyService.Get<IAudio>().PlayAudioFile("policesiren.mp3");
        }

         0 references
         void SendSMS(object sender, EventArgs e)
        {
            var SmsTask = MessagingPlugin.SmsMessenger;

            if (SmsTask.CanSendSms)
                SmsTask.SendSms(txtPhoneNumber.Text, "I'm in Danger,My Location: "+ lbllocatio
        }
        0 references
        void CallPolice(object sender,EventArgs e)
```

```
We were unable to automatically populate your Visual Studio Team Services accounts.

The following error was encountered: TF400813: Resource not available for anonymous access. Client authentication required.
```

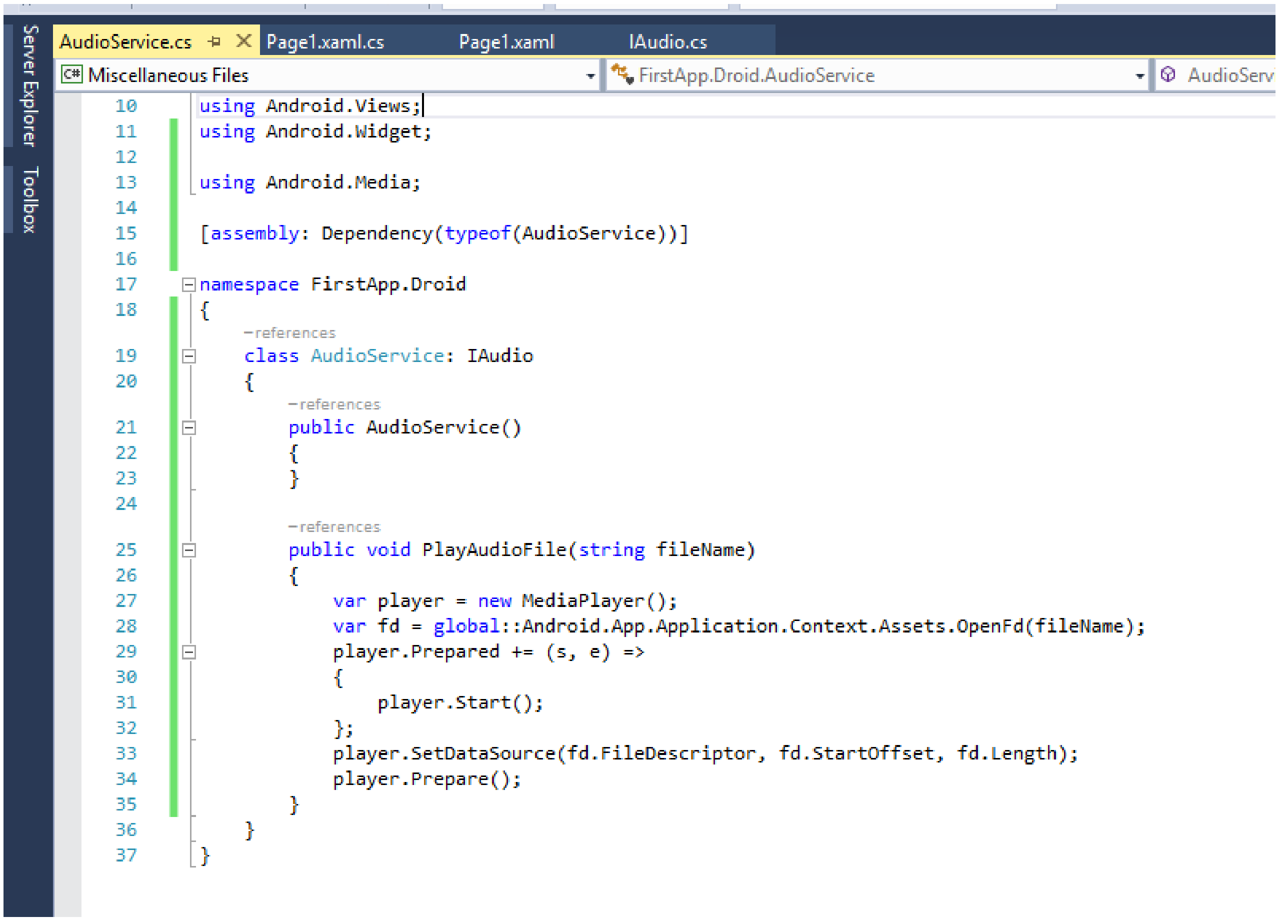
6. Add class “**AudioService.cs**” to **FirstApp.Droid**.





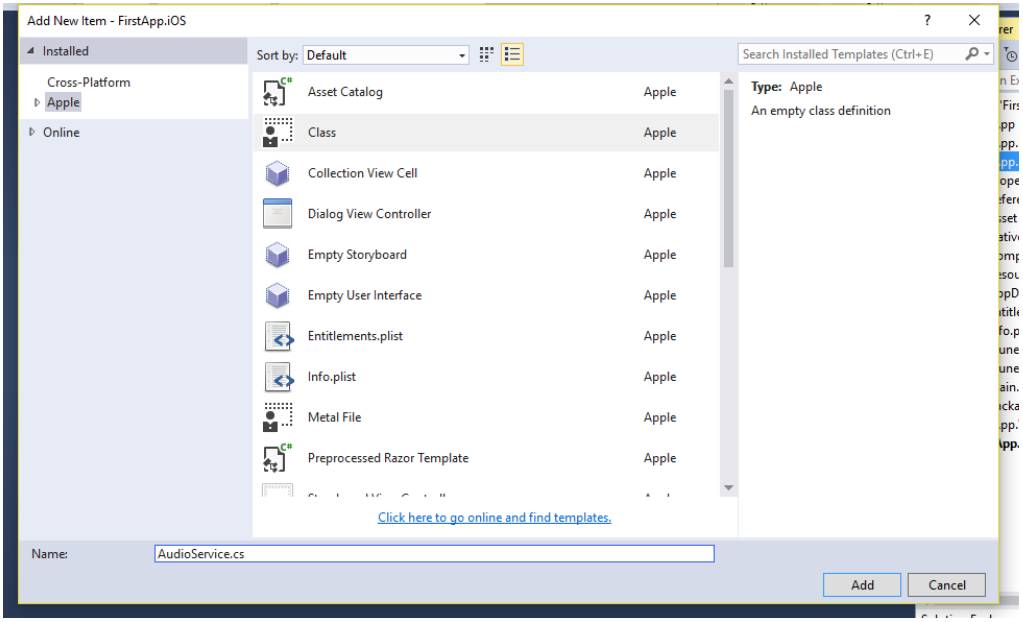
Name: AudioService.cs

And add the Code to “**AudioServices.cs**”



```
using Android.Views;
using Android.Widget;

using Android.Media;

[assembly: Dependency(typeof(AudioService))]

namespace FirstApp.Droid
{
    -references
    class AudioService: IAudio
    {
        -references
        public AudioService()
        {
        }

        -references
        public void PlayAudioFile(string fileName)
        {
            var player = new MediaPlayer();
            var fd = global::Android.App.Application.Context.Assets.OpenFd(fileName);
            player.Prepared += (s, e) =>
            {
                player.Start();
            };
            player.SetDataSource(fd.FileDescriptor, fd.StartOffset, fd.Length);
            player.Prepare();
        }
    }
}
```

7. In iOS still same steps with Android add the “**AudioServices.cs**”

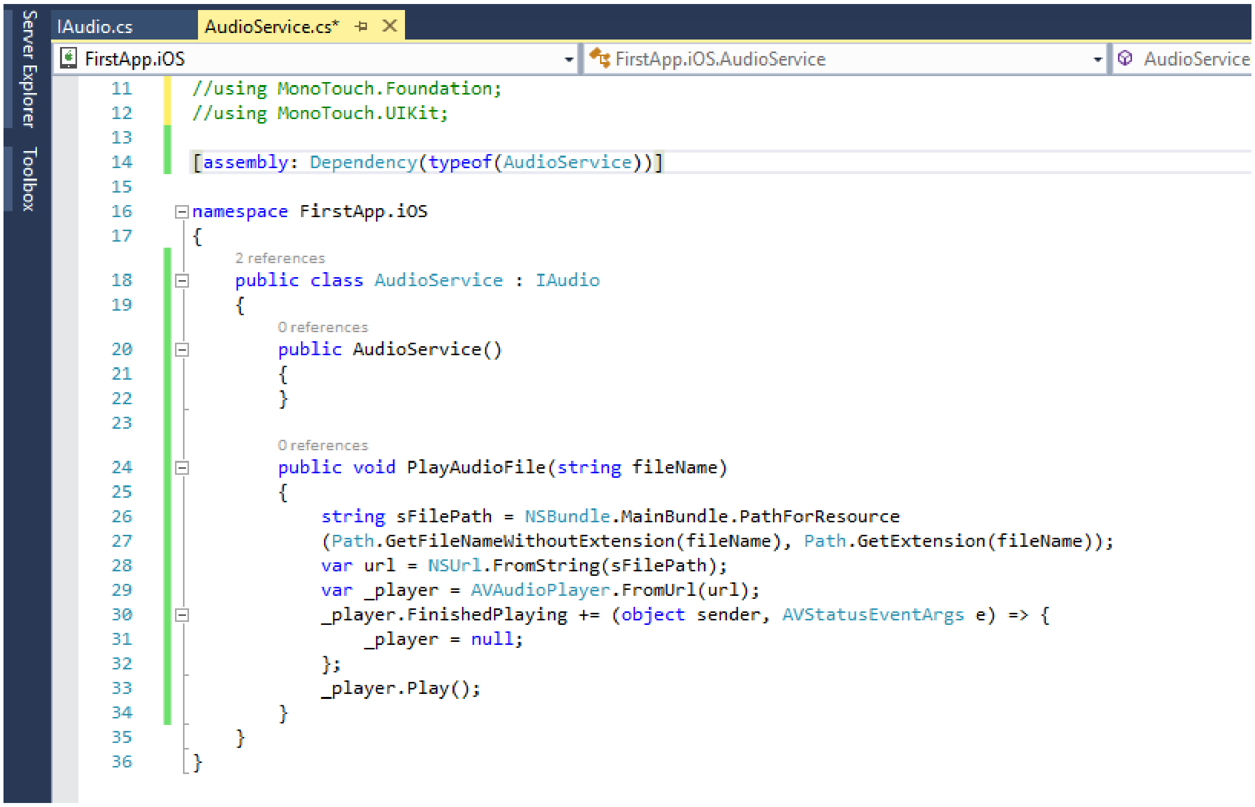

Add New Item - FirstApp.iOS
?
×
Installed
Cross-Platform
Apple
Online
Sort by:
Default
Search Installed Templates (Ctrl+E)
Click here to go online and find templates.
Add
Cancel

Asset Catalog Apple
Class Apple
Collection View Cell Apple
Dialog View Controller Apple
Empty Storyboard Apple
Empty User Interface Apple
Entitlements.plist Apple
Info.plist Apple
Metal File Apple
Preprocessed Razor Template Apple

Type: Apple
An empty class definition

Name: AudioService.cs

And Add the code to “**AudioServices.cs**”



```
//using MonoTouch.Foundation;
//using MonoTouch.UIKit;

[assembly: Dependency(typeof(AudioService))]

namespace FirstApp.iOS
{
    public class AudioService : IAudio
    {
        public AudioService()
        {
        }

        public void PlayAudioFile(string fileName)
        {
            string sFilePath = NSBundle.MainBundle.PathForResource
            (Path.GetFileNameWithoutExtension(fileName), Path.GetExtension(fileName));
            var url = NSUrl.FromString(sFilePath);
            var _player = AVAudioPlayer.FromUrl(url);
            _player.FinishedPlaying += (object sender, AVStatusEventArgs e) => {
                _player = null;
            };
            _player.Play();
        }
    }
}
```

UWP/Windows/Windows Phone 8.1

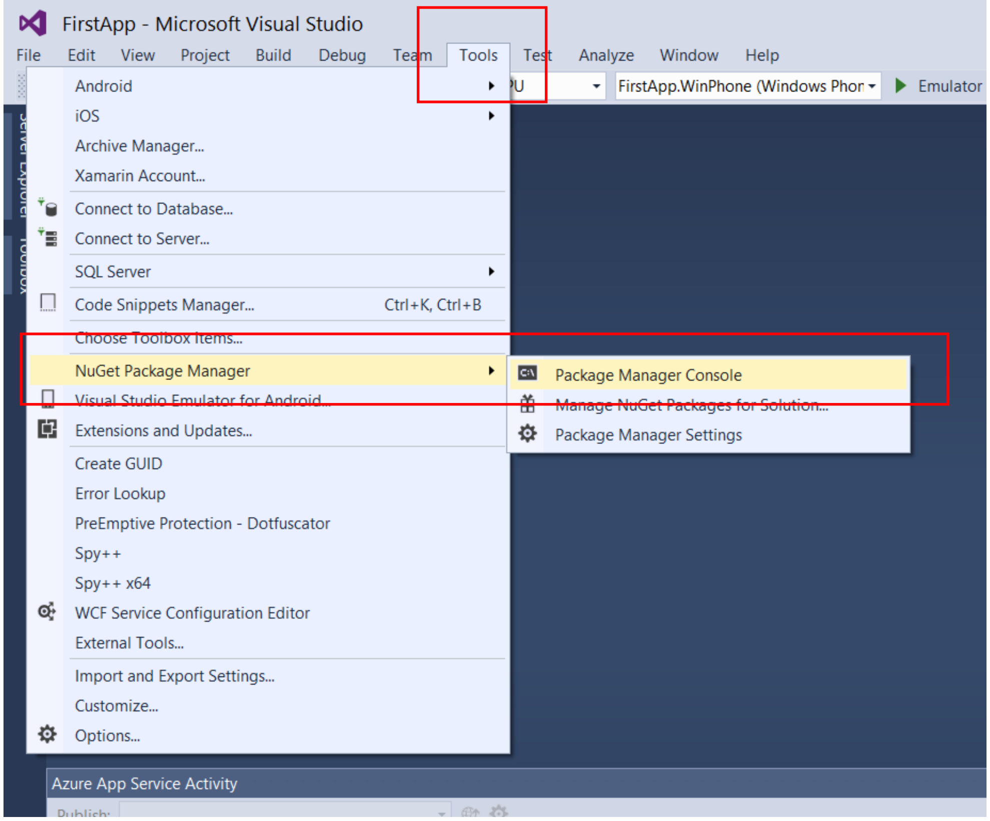
==========================

Temporary unavailable

Chapter 3: Plugin

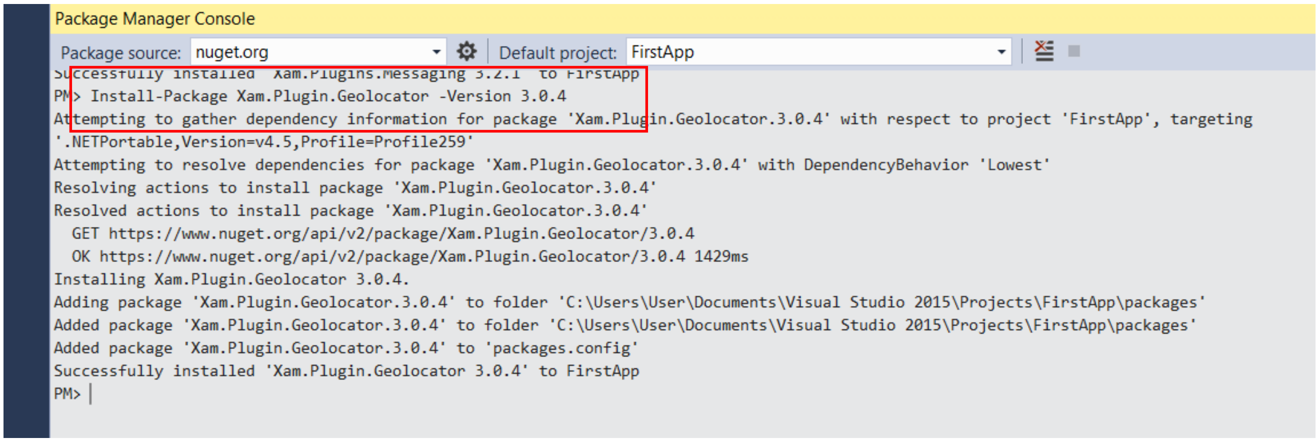
Geolocator Plugin

1. Go to Tools > Nuget Package Manager > Package Manager Console.



2. Key In the Command & Enter for installation of Library.

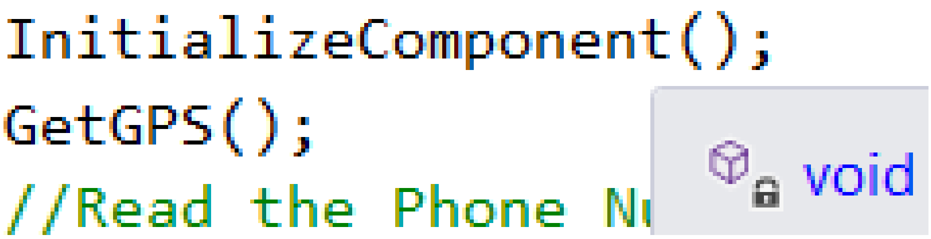
```
Install-Package Xam.Plugin.Geolocator -Version 3.0.4
```



```
Successfully installed 'Xam.Plugins.Messaging 3.2.1' to FirstApp
PM> Install-Package Xam.Plugin.Geolocator -Version 3.0.4
Attempting to gather dependency information for package 'Xam.Plugin.Geolocator.3.0.4' with respect to project 'FirstApp', targeting
'.NETPortable,Version=v4.5,Profile=Profile259'
Attempting to resolve dependencies for package 'Xam.Plugin.Geolocator.3.0.4' with DependencyBehavior 'Lowest'
Resolving actions to install package 'Xam.Plugin.Geolocator.3.0.4'
Resolved actions to install package 'Xam.Plugin.Geolocator.3.0.4'
  GET https://www.nuget.org/api/v2/package/Xam.Plugin.Geolocator/3.0.4
  OK https://www.nuget.org/api/v2/package/Xam.Plugin.Geolocator/3.0.4 1429ms
Installing Xam.Plugin.Geolocator 3.0.4.
Adding package 'Xam.Plugin.Geolocator.3.0.4' to folder 'C:\Users\User\Documents\Visual Studio 2015\Projects\FirstApp\packages'
Added package 'Xam.Plugin.Geolocator.3.0.4' to folder 'C:\Users\User\Documents\Visual Studio 2015\Projects\FirstApp\packages'
Added package 'Xam.Plugin.Geolocator.3.0.4' to 'packages.config'
Successfully installed 'Xam.Plugin.Geolocator 3.0.4' to FirstApp
PM>
```

3. Go to **HomePage.xaml.cs** add the Function for retrieve the location details

GetGPS(); at Below **InitializeComponent();**

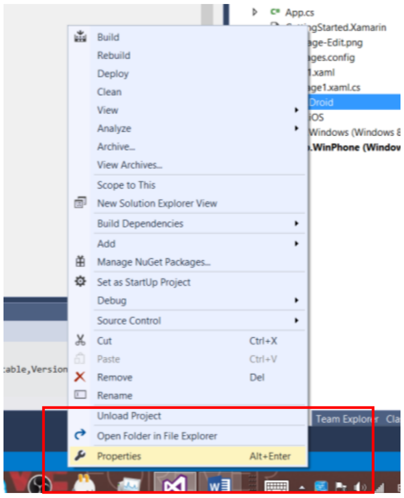


```
InitializeComponent();
GetGPS();
//Read the Phone Nu
```

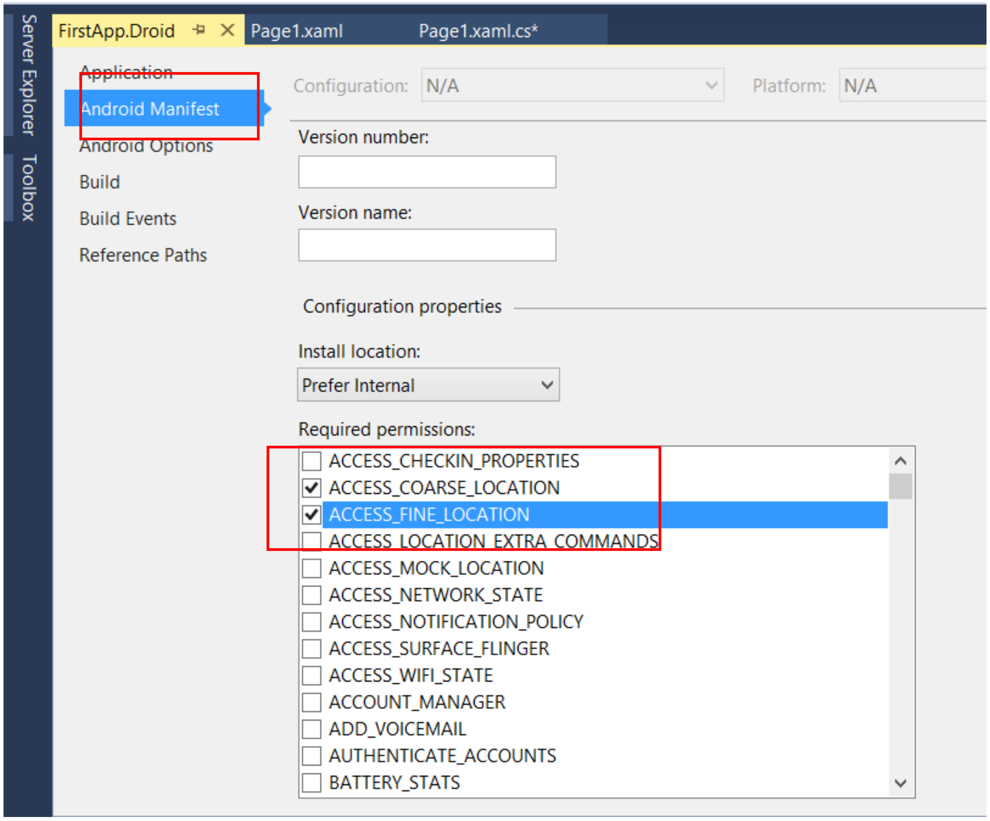
And Enter the Code Below

```
private async void GetGPS()
        {
            try
            {
                var locator = CrossGeolocator.Current;
                locator.DesiredAccuracy = 1000;
                lbllocation.Text = "Getting gps";

                var position = await locator.GetPositionAsync(timeoutMilliseconds: 10000);

                if (position == null)
                {
                    lbllocation.Text = "null gps :(";
                    return;
                }
                lbllocation.Text = string.Format("Time: {0} \nLat: {1} \nLong: {2}"+
                   " \nAltitude: {3} \nAltitude Accuracy: {4} \nAccuracy: {5} \nHeading: {6} \nSpeed: {7}",
                    position.Timestamp, position.Latitude, position.Longitude,
                    position.Altitude, position.AltitudeAccuracy, position.Accuracy, position.Heading, position.Speed);

            }
            catch //(Exception ex)
            {
                // Xamarin.Insights.Report(ex);
                // await DisplayAlert("Uh oh", "Something went wrong, but don't worry we captured it in Xamarin Insights! Thanks.", "OK");
            }
        }
        protected override void OnAppearing()
        {
            base.OnAppearing();
            try
            {
                CrossGeolocator.Current.PositionChanged +=
CrossGeolocator_Current_PositionChanged;
                CrossGeolocator.Current.PositionError += CrossGeolocator_Current_PositionError;
            }
            catch
            {
            }
        }

        void CrossGeolocator_Current_PositionError(object sender,
Plugin.Geolocator.Abstractions.PositionErrorEventArgs e)
        {

            lbllocation.Text = "Location error: " + e.Error.ToString();
        }

        void CrossGeolocator_Current_PositionChanged(object sender,
Plugin.Geolocator.Abstractions.PositionEventArgs e)
        {
            var position = e.Position;
            lbllocation.Text = string.Format("Time: {0} \nLat: {1} \nLong: {2} \nAltitude: "
                +"{3} \nAltitude Accuracy: {4} \nAccuracy: {5} \nHeading: {6} \nSpeed: {7}",
                position.Timestamp, position.Latitude, position.Longitude,
                position.Altitude, position.AltitudeAccuracy, position.Accuracy, position.Heading,
position.Speed);


        }

        protected override void OnDisappearing()
        {
            base.OnDisappearing();
            try
            {
                CrossGeolocator.Current.PositionChanged -=
CrossGeolocator_Current_PositionChanged;
                CrossGeolocator.Current.PositionError -= CrossGeolocator_Current_PositionError;
            }
            catch
            {
            }
        }
```

4. Go to **FirstApp.Droid** , Right Click > Select Properties



5. Go to **Android Manifest** Enable Location Permission



Configuration: N/A

Platform: N/A

Version number:

Version name:

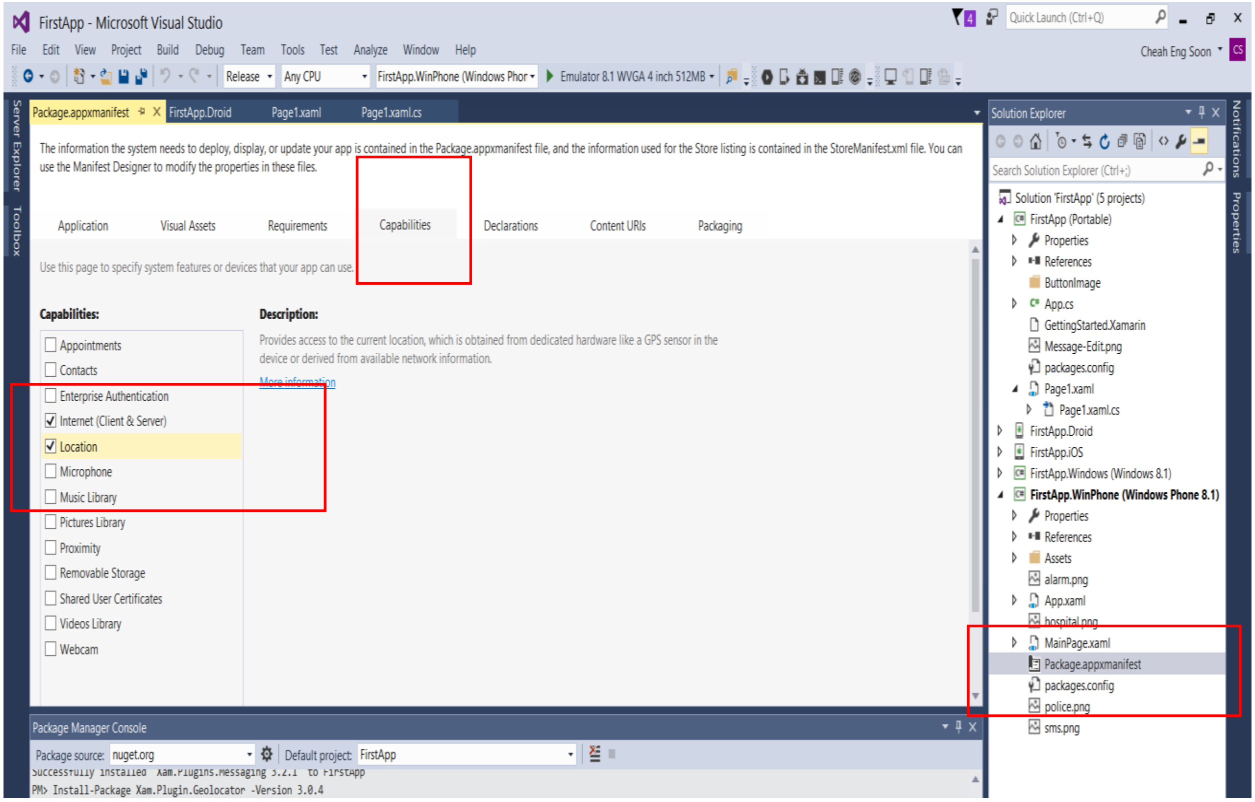
Install location: Prefer Internal

Required permissions:

- □ ACCESS_CHECKIN_PROPERTIES
- ☑ ACCESS_COARSE_LOCATION
- ☑ ACCESS_FINE_LOCATION
- □ ACCESS_LOCATION_EXTRA_COMMANDS
- □ ACCESS_MOCK_LOCATION
- □ ACCESS_NETWORK_STATE
- □ ACCESS_NOTIFICATION_POLICY
- □ ACCESS_SURFACE_FLINGER
- □ ACCESS_WIFI_STATE
- □ ACCOUNT_MANAGER
- □ ADD_VOICEMAIL
- □ AUTHENTICATE_ACCOUNTS
- □ BATTERY_STATS

For **Windows Phone/Windows/UWP**

Go to “**Package.appxmanifest**” to Enable Location Services



The information the system needs to deploy, display, or update your app is contained in the Package.appxmanifest file, and the information used for the Store listing is contained in the StoreManifest.xml file. You can use the Manifest Designer to modify the properties in these files.

Use this page to specify system features or devices that your app can use.

Capabilities:

- □ Appointments
- □ Contacts
- □ Enterprise Authentication
- ☑ Internet (Client & Server)
- ☑ Location
- □ Microphone
- □ Music Library
- □ Pictures Library
- □ Proximity
- □ Removable Storage
- □ Shared User Certificates
- □ Videos Library
- □ Webcam

Description:

Provides access to the current location, which is obtained from dedicated hardware like a GPS sensor in the device or derived from available network information.

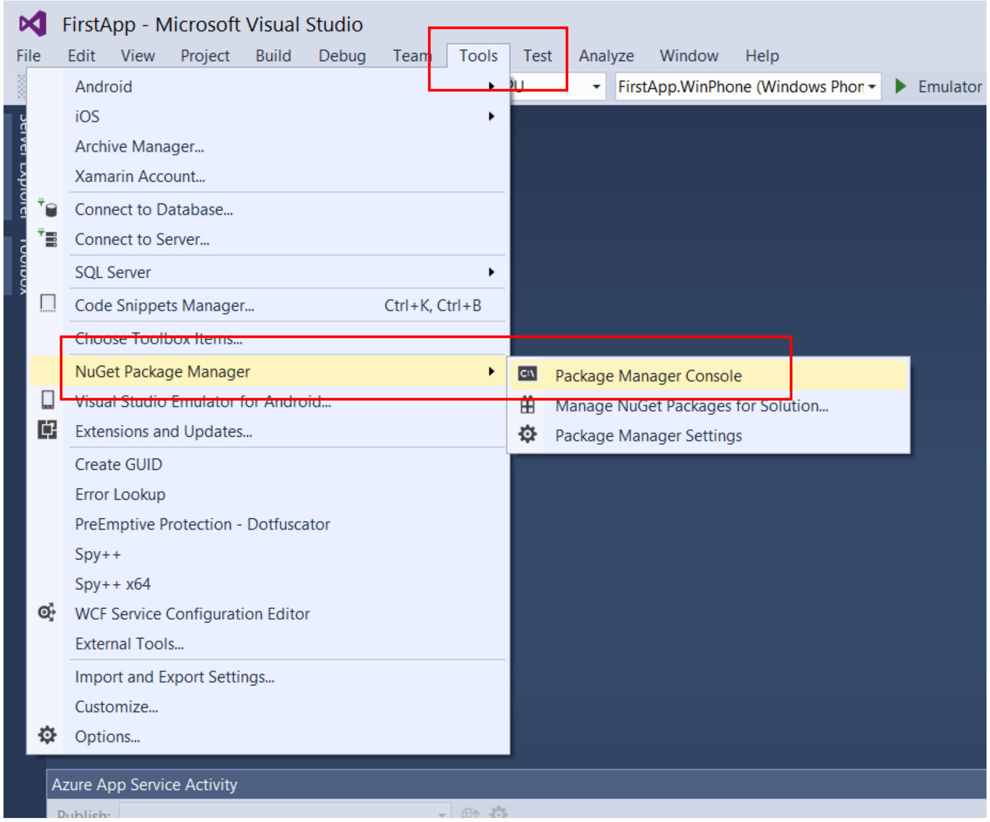
Package source: nuget.org

Default project: FirstApp

```
Successfully installed  Xam.Plugins.Messaging 3.2.1  to FirstApp
PM> Install-Package Xam.Plugin.Geolocator -Version 3.0.4
```

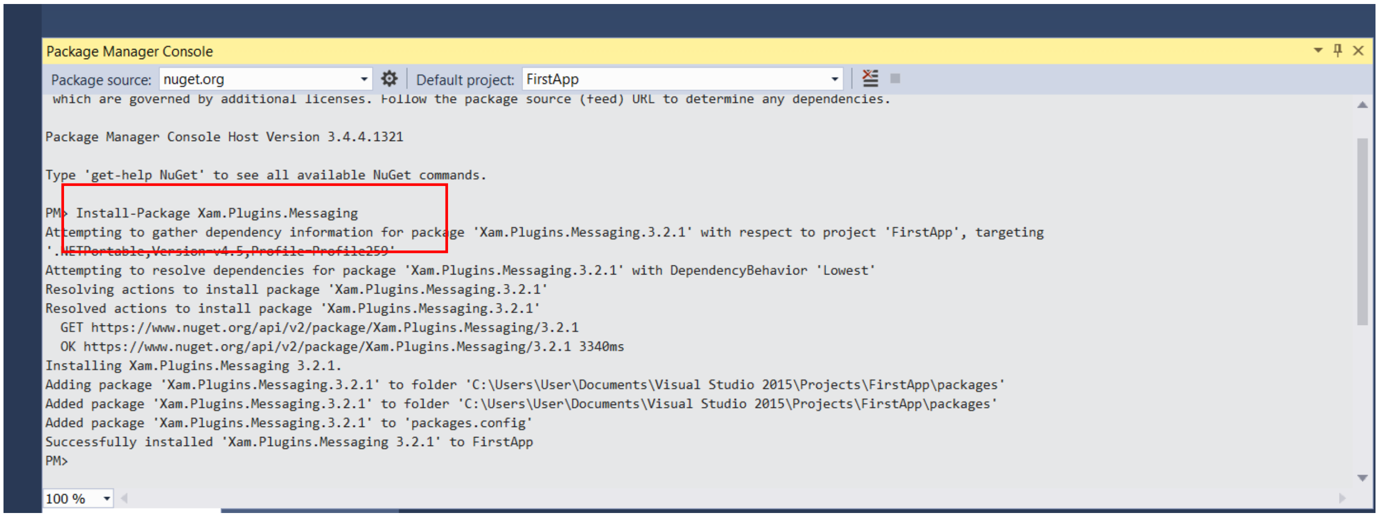
SMS & Call Plugin

1. Go to Tools > Nuget Package Manager > Package Manager Console.



2. Enter the Command & Enter for installation of Library.

Install-Package Xam.Plugins.Messaging

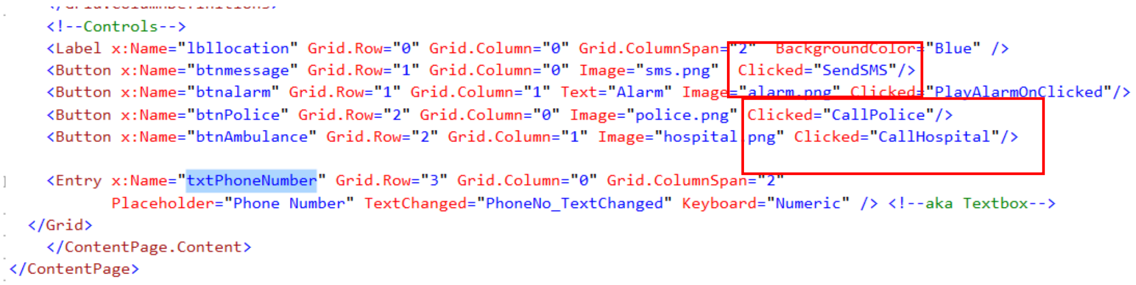


Package source: nuget.org

Default project: FirstApp

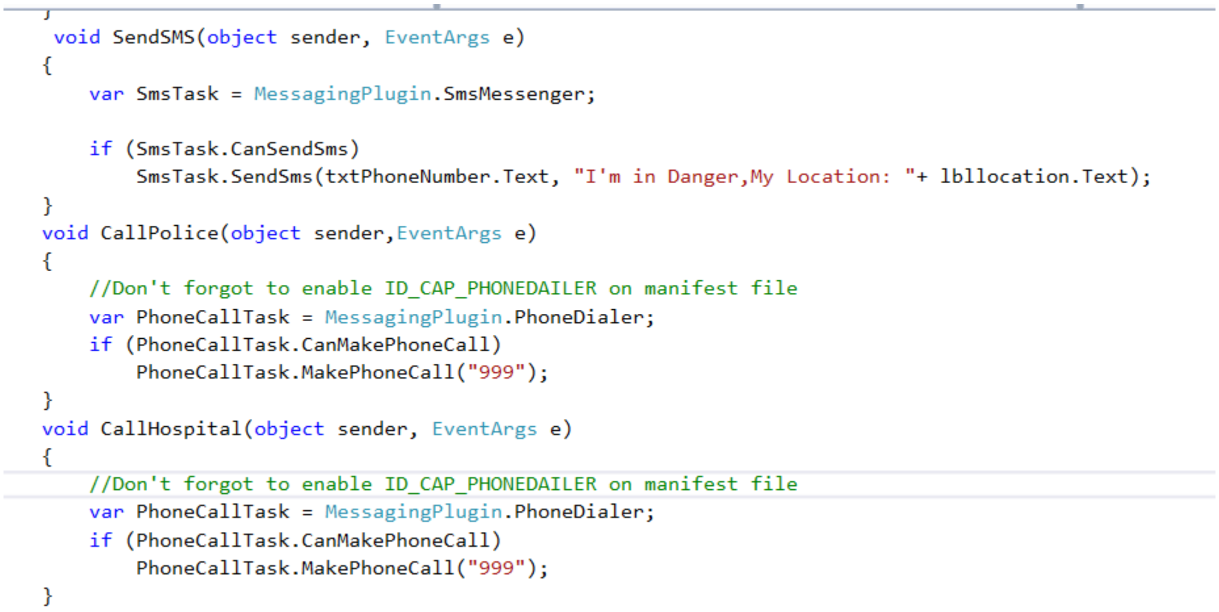
```
which are governed by additional licenses. Follow the package source (feed) URL to determine any dependencies.

Package Manager Console Host Version 3.4.4.1321

Type 'get-help NuGet' to see all available NuGet commands.

PM> Install-Package Xam.Plugins.Messaging
Attempting to gather dependency information for package 'Xam.Plugins.Messaging.3.2.1' with respect to project 'FirstApp', targeting
'.NETPortable,Version=v4.5,Profile=Profile259'
Attempting to resolve dependencies for package 'Xam.Plugins.Messaging.3.2.1' with DependencyBehavior 'Lowest'
Resolving actions to install package 'Xam.Plugins.Messaging.3.2.1'
Resolved actions to install package 'Xam.Plugins.Messaging.3.2.1'
  GET https://www.nuget.org/api/v2/package/Xam.Plugins.Messaging/3.2.1
  OK https://www.nuget.org/api/v2/package/Xam.Plugins.Messaging/3.2.1 3340ms
Installing Xam.Plugins.Messaging 3.2.1.
Adding package 'Xam.Plugins.Messaging.3.2.1' to folder 'C:\Users\User\Documents\Visual Studio 2015\Projects\FirstApp\packages'
Added package 'Xam.Plugins.Messaging.3.2.1' to folder 'C:\Users\User\Documents\Visual Studio 2015\Projects\FirstApp\packages'
Added package 'Xam.Plugins.Messaging.3.2.1' to 'packages.config'
Successfully installed 'Xam.Plugins.Messaging 3.2.1' to FirstApp
PM>
```

3. Go to **HomePage.xaml** create an event handler for Call & SMS Function



```
    <!--Controls-->
    <Label x:Name="lbllocation" Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2"  BackgroundColor="Blue" />
    <Button x:Name="btnmessage" Grid.Row="1" Grid.Column="0" Image="sms.png"  Clicked="SendSMS"/>
    <Button x:Name="btnalarm" Grid.Row="1" Grid.Column="1" Text="Alarm" Image="alarm.png" Clicked="PlayAlarmOnClicked"/>
    <Button x:Name="btnPolice" Grid.Row="2" Grid.Column="0" Image="police.png" Clicked="CallPolice"/>
    <Button x:Name="btnAmbulance" Grid.Row="2" Grid.Column="1" Image="hospital.png" Clicked="CallHospital"/>

    <Entry x:Name="txtPhoneNumber" Grid.Row="3" Grid.Column="0" Grid.ColumnSpan="2"
           Placeholder="Phone Number" TextChanged="PhoneNo_TextChanged" Keyboard="Numeric" /> <!--aka Textbox-->
  </Grid>
    </ContentPage.Content>
</ContentPage>
```

4. Go to **HomePage.xaml.cs** add the Backend Code.



```
     void SendSMS(object sender, EventArgs e)
    {
        var SmsTask = MessagingPlugin.SmsMessenger;

        if (SmsTask.CanSendSms)
            SmsTask.SendSms(txtPhoneNumber.Text, "I'm in Danger,My Location: "+ lbllocation.Text);
    }
    void CallPolice(object sender,EventArgs e)
    {
        //Don't forgot to enable ID_CAP_PHONEDAILER on manifest file
        var PhoneCallTask = MessagingPlugin.PhoneDialer;
        if (PhoneCallTask.CanMakePhoneCall)
            PhoneCallTask.MakePhoneCall("999");
    }
    void CallHospital(object sender, EventArgs e)
    {
        //Don't forgot to enable ID_CAP_PHONEDAILER on manifest file
        var PhoneCallTask = MessagingPlugin.PhoneDialer;
        if (PhoneCallTask.CanMakePhoneCall)
            PhoneCallTask.MakePhoneCall("999");
    }
```

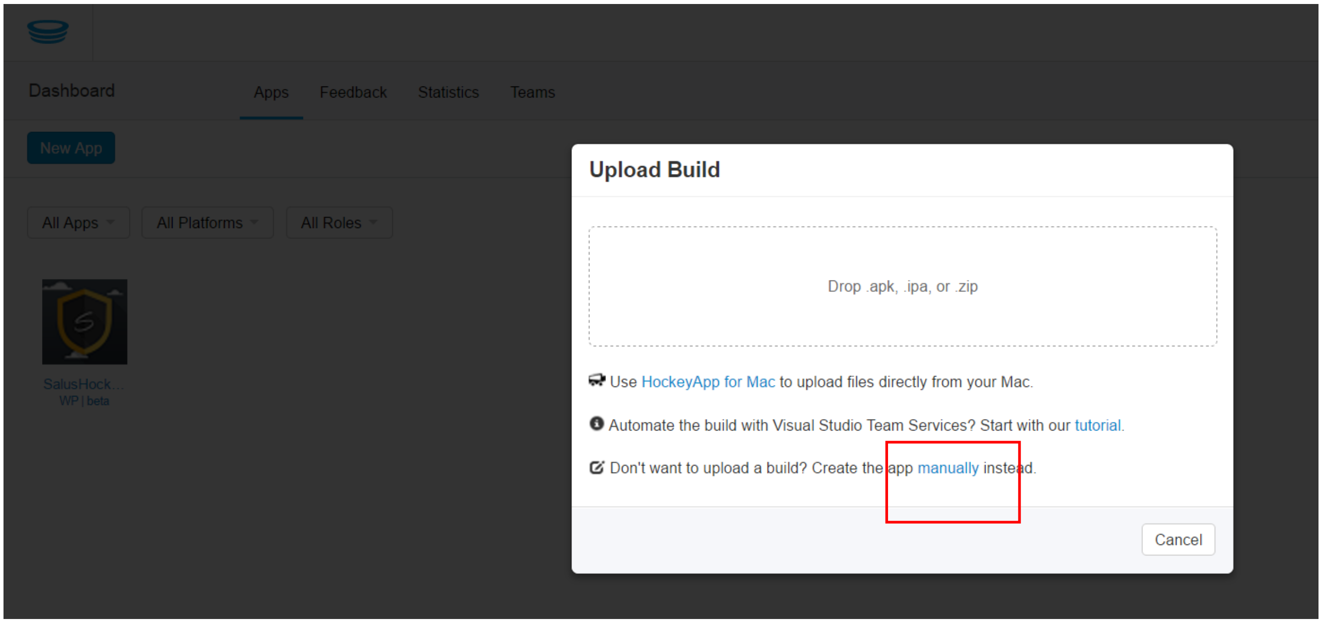
5. Enable the **CALL_PHONE** and **SEND_SMS** in Android Manifest . For WindowsPhone/UWP enable **ID_CAP_PHONEDAILER**

CHAPTER 4: Integrate with HockeyApp

1. Go to https://hockeyapp.net/

2. Sign Up with your Microsoft Account

3. Create a New App and Select “manually”



Upload Build

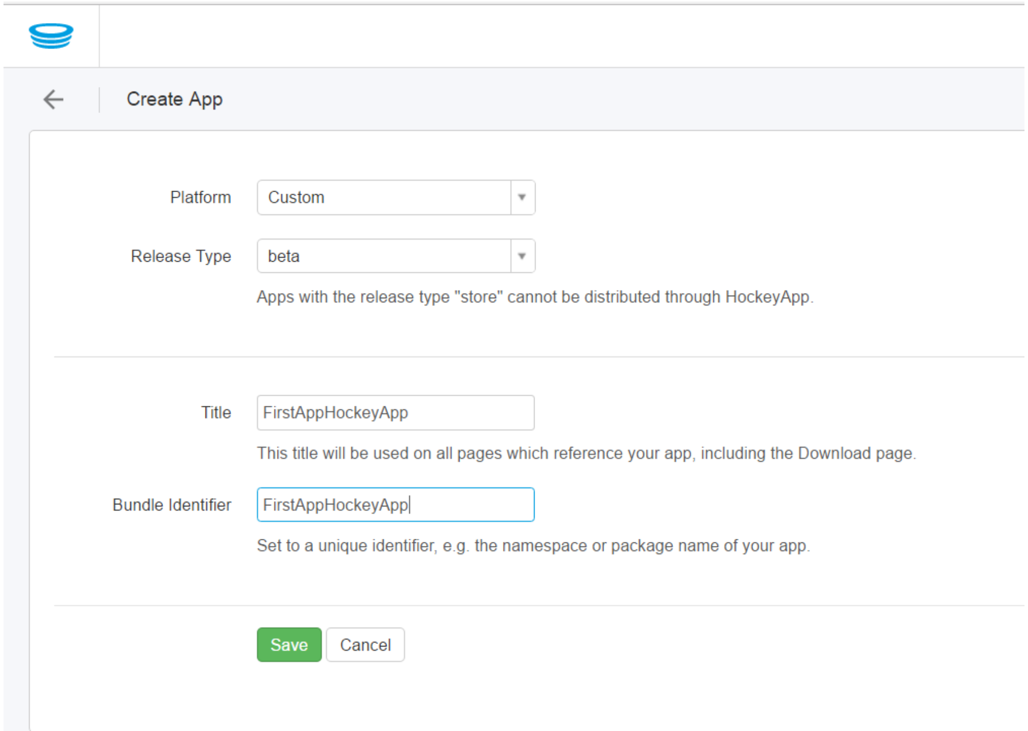
Drop .apk, .ipa, or .zip

Use HockeyApp for Mac to upload files directly from your Mac.

Automate the build with Visual Studio Team Services? Start with our tutorial.

Don't want to upload a build? Create the app manually instead.

4. Fill in the Details & Save.


Create App

Platform: Custom

Release Type: beta

Apps with the release type "store" cannot be distributed through HockeyApp.

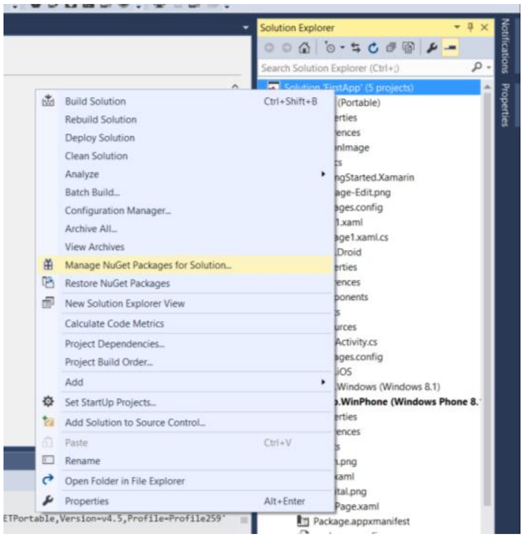
Title: FirstAppHockeyApp

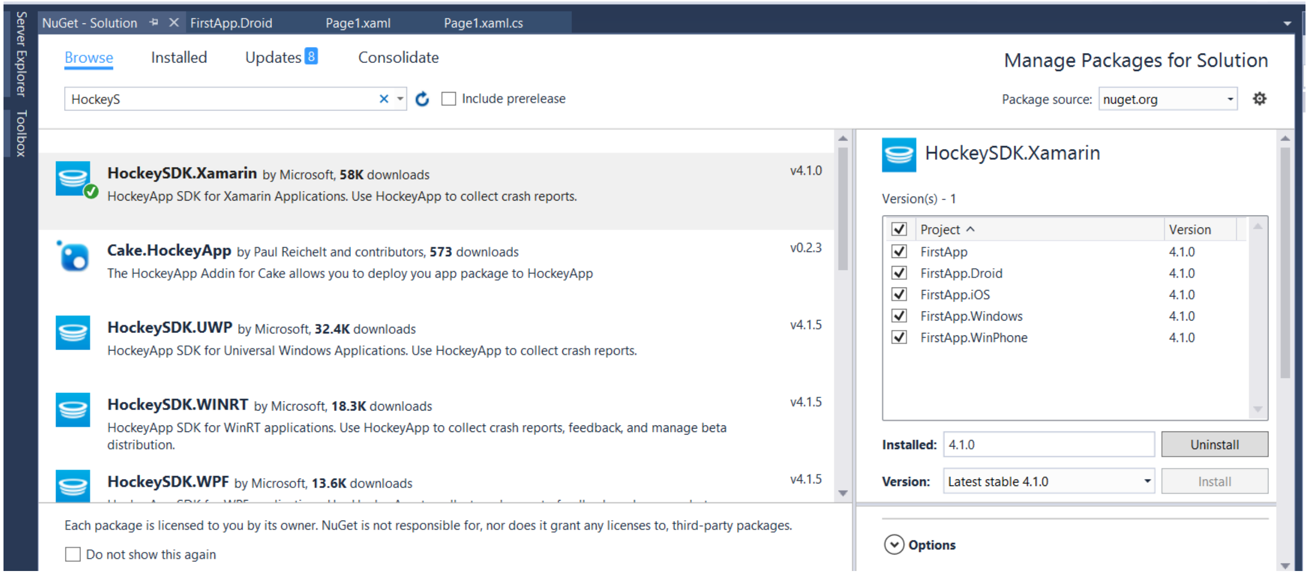
This title will be used on all pages which reference your app, including the Download page.

Bundle Identifier: FirstAppHockeyApp

Set to a unique identifier, e.g. the namespace or package name of your app.

5. Go to Project > Manage NuGet Packages …. And Search HockeySDK.Xamarin





HockeyS

HockeySDK.Xamarin by Microsoft, **58K** downloads v4.1.0
HockeyApp SDK for Xamarin Applications. Use HockeyApp to collect crash reports.

Cake.HockeyApp by Paul Reichelt and contributors, **573** downloads v0.2.3
The HockeyApp Addin for Cake allows you to deploy you app package to HockeyApp

HockeySDK.UWP by Microsoft, **32.4K** downloads v4.1.5
HockeyApp SDK for Universal Windows Applications. Use HockeyApp to collect crash reports.

HockeySDK.WINRT by Microsoft, **18.3K** downloads v4.1.5
HockeyApp SDK for WinRT applications. Use HockeyApp to collect crash reports, feedback, and manage beta distribution.

HockeySDK.WPF by Microsoft, **13.6K** downloads v4.1.5

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

□ Do not show this again

HockeySDK.Xamarin

Version(s) - 1

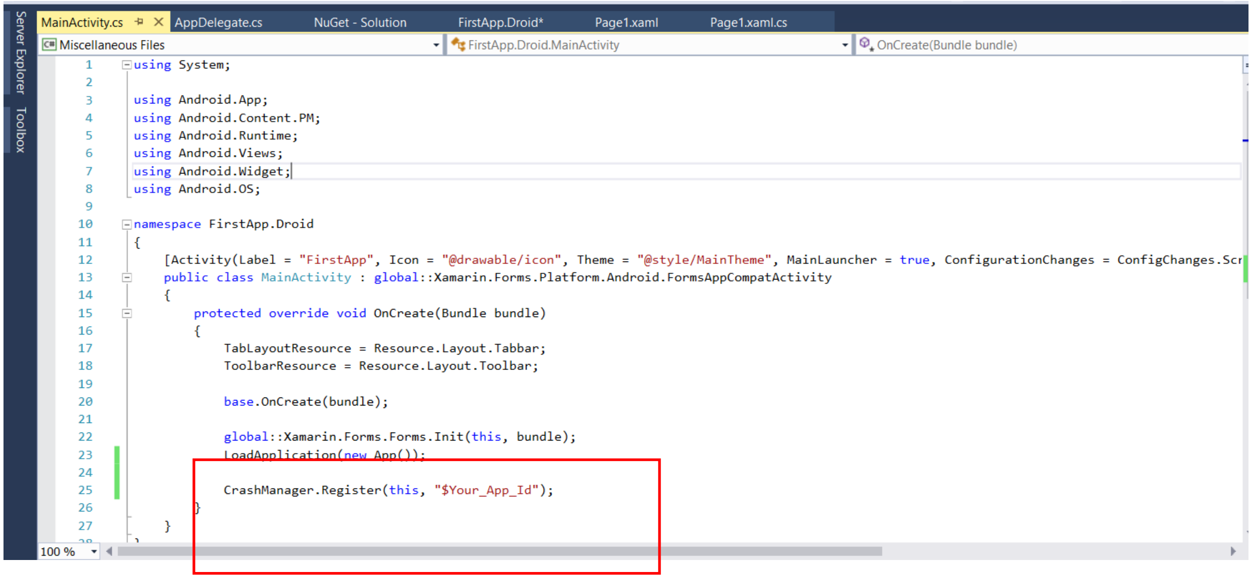
☑ Project	Version
☑ FirstApp	4.1.0
☑ FirstApp.Droid	4.1.0
☑ FirstApp.iOS	4.1.0
☑ FirstApp.Windows	4.1.0
☑ FirstApp.WinPhone	4.1.0

Installed: 4.1.0

Version: Latest stable 4.1.0

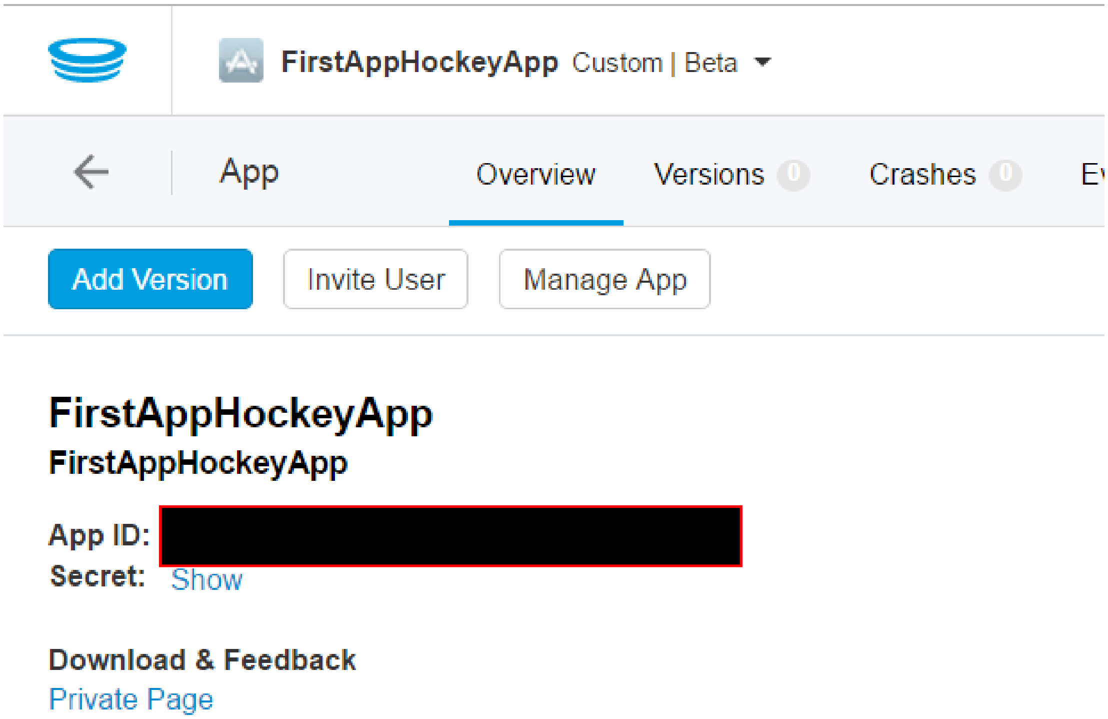
For Android

Open **MainActivity.cs**



```
using System;

using Android.App;
using Android.Content.PM;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using Android.OS;

namespace FirstApp.Droid
{
    [Activity(Label = "FirstApp", Icon = "@drawable/icon", Theme = "@style/MainTheme", MainLauncher = true, ConfigurationChanges = ConfigChanges.Scr
    public class MainActivity : global::Xamarin.Forms.Platform.Android.FormsAppCompatActivity
    {
        protected override void OnCreate(Bundle bundle)
        {
            TabLayoutResource = Resource.Layout.Tabbar;
            ToolbarResource = Resource.Layout.Toolbar;

            base.OnCreate(bundle);

            global::Xamarin.Forms.Forms.Init(this, bundle);
            LoadApplication(new App());

            CrashManager.Register(this, "$Your_App_Id");
        }
    }
```

Replace **“$Your_App_Id”** That you can get in Hockey App.



FirstAppHockeyApp

FirstAppHockeyApp

App ID: [redacted]

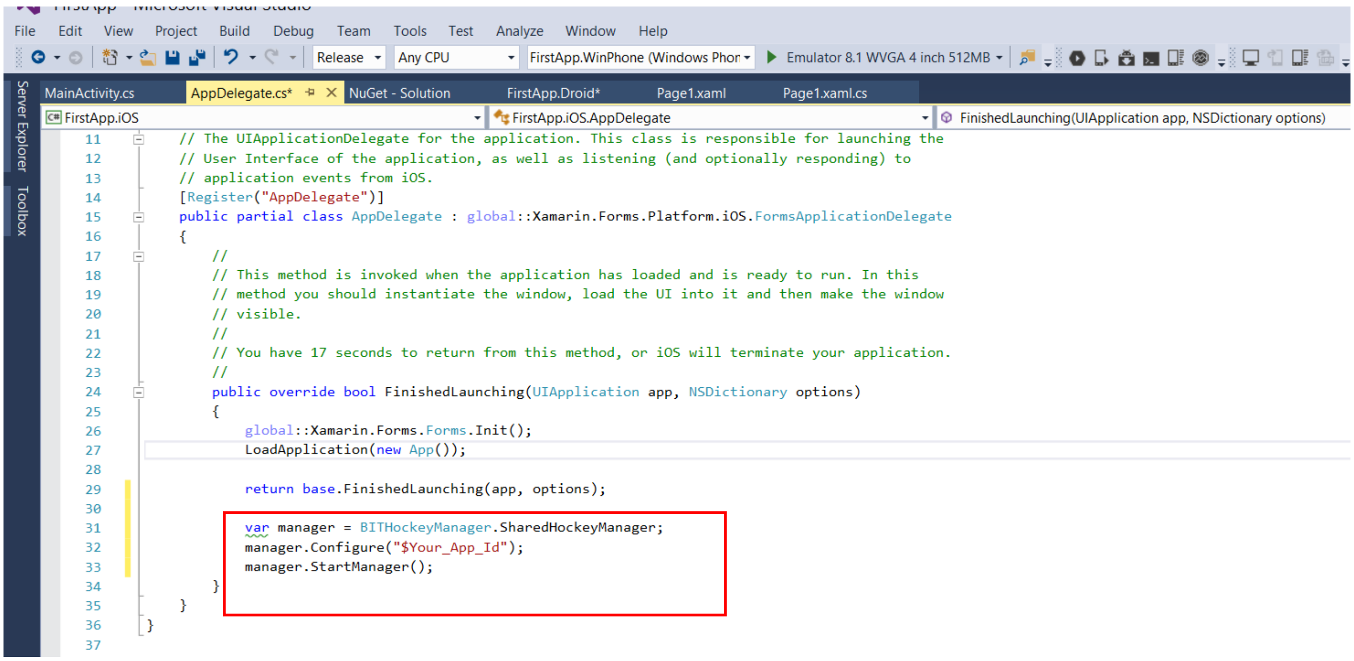
Secret: Show

Download & Feedback

Private Page

For iOS

1. Open **AppDelegate.cs** and add this new line of code.


FirstApp - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Release
Any CPU
FirstApp.WinPhone (Windows Phor
Emulator 8.1 WVGA 4 inch 512MB
Server Explorer
Toolbox
MainActivity.cs
AppDelegate.cs*
NuGet - Solution
FirstApp.Droid*
Page1.xaml
Page1.xaml.cs
FirstApp.iOS
FirstApp.iOS.AppDelegate
FinishedLaunching(UIApplication app, NSDictionary options)

```
    // The UIApplicationDelegate for the application. This class is responsible for launching the
    // User Interface of the application, as well as listening (and optionally responding) to
    // application events from iOS.
    [Register("AppDelegate")]
    public partial class AppDelegate : global::Xamarin.Forms.Platform.iOS.FormsApplicationDelegate
    {
        //
        // This method is invoked when the application has loaded and is ready to run. In this
        // method you should instantiate the window, load the UI into it and then make the window
        // visible.
        //
        // You have 17 seconds to return from this method, or iOS will terminate your application.
        //
        public override bool FinishedLaunching(UIApplication app, NSDictionary options)
        {
            global::Xamarin.Forms.Forms.Init();
            LoadApplication(new App());

            return base.FinishedLaunching(app, options);

            var manager = BITHockeyManager.SharedHockeyManager;
            manager.Configure("$Your_App_Id");
            manager.StartManager();
        }
    }
}
```

For UWP

https://www.hockeyapp.net/blog/2016/10/18/the-latest-on-hockeysdk-for-uwp.html

References:

Microsoft Virtual Academy

https://mva.microsoft.com/en-US/training-courses/xamarin-for-absolute-beginners-16182

Xamarin Malaysia Developers

https://www.facebook.com/groups/xamarinmydev/

HockeyApp

https://hockeyapp.net/

Source Code