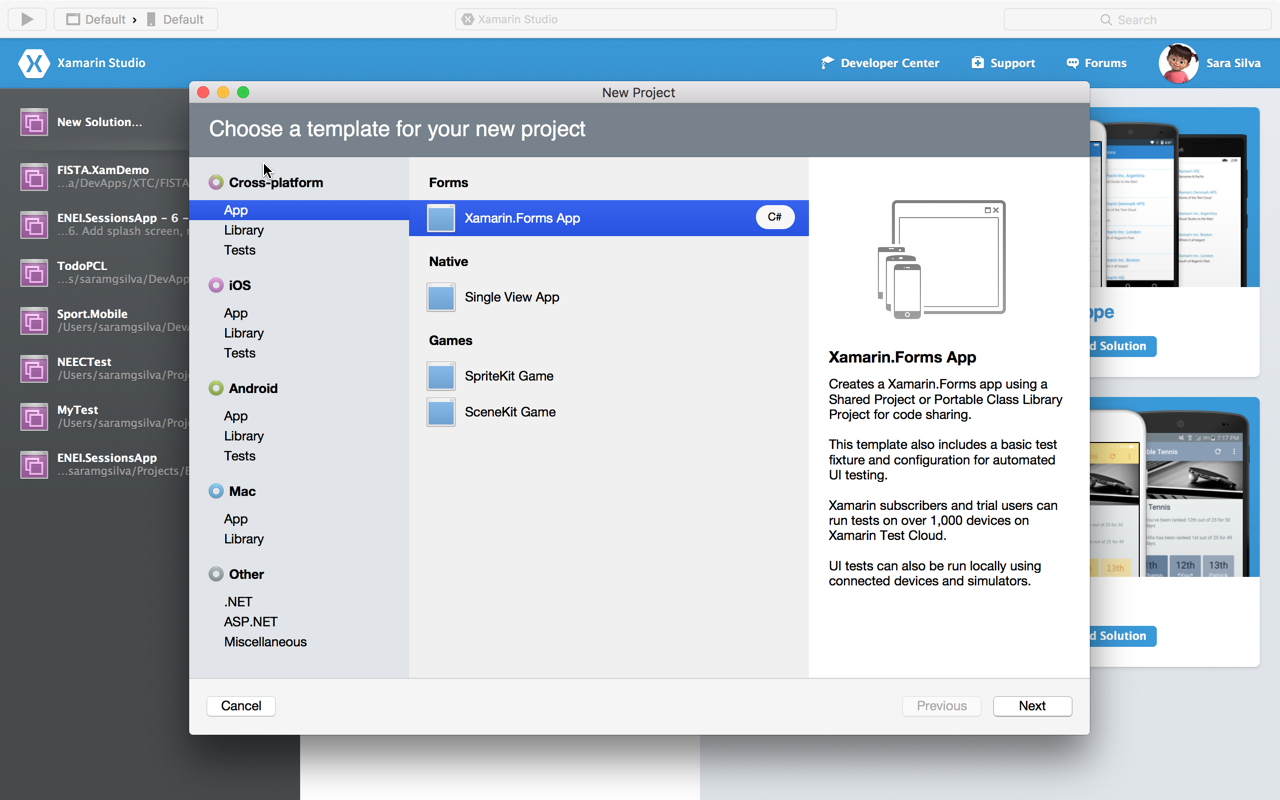
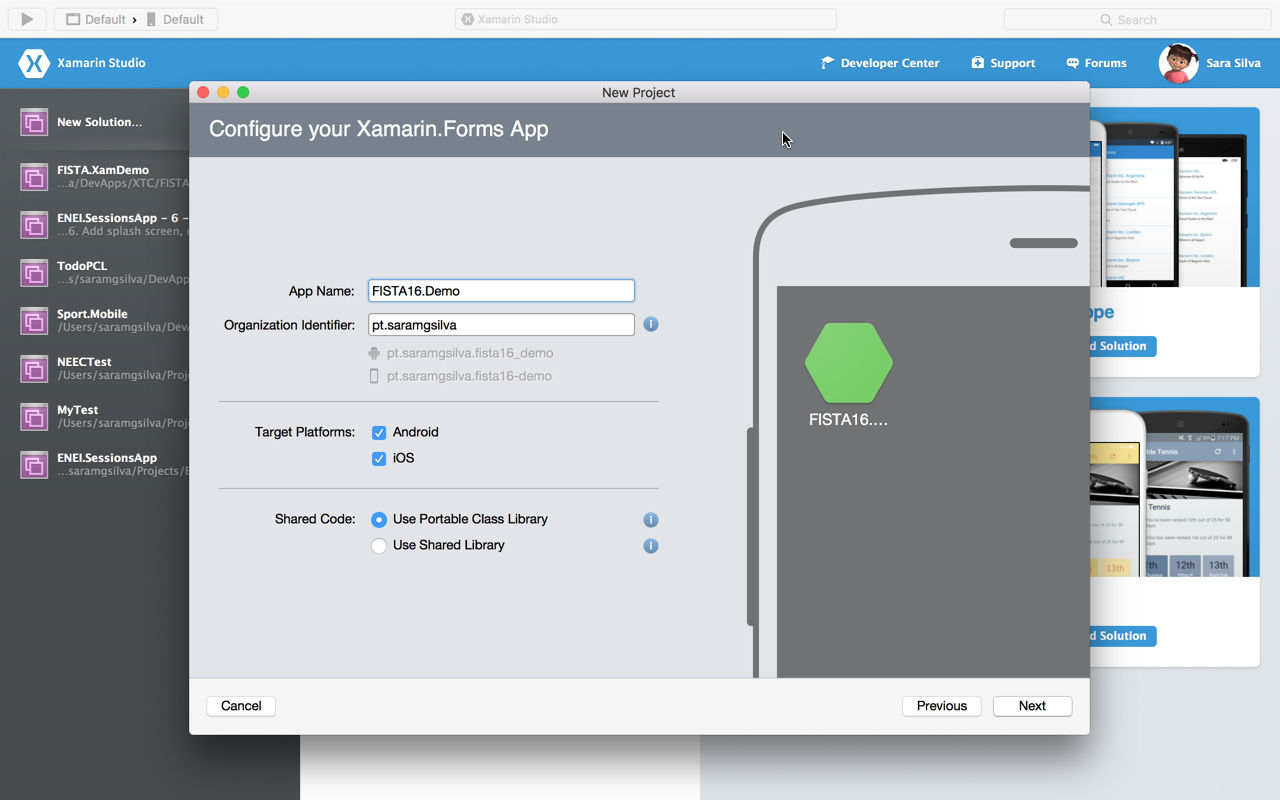
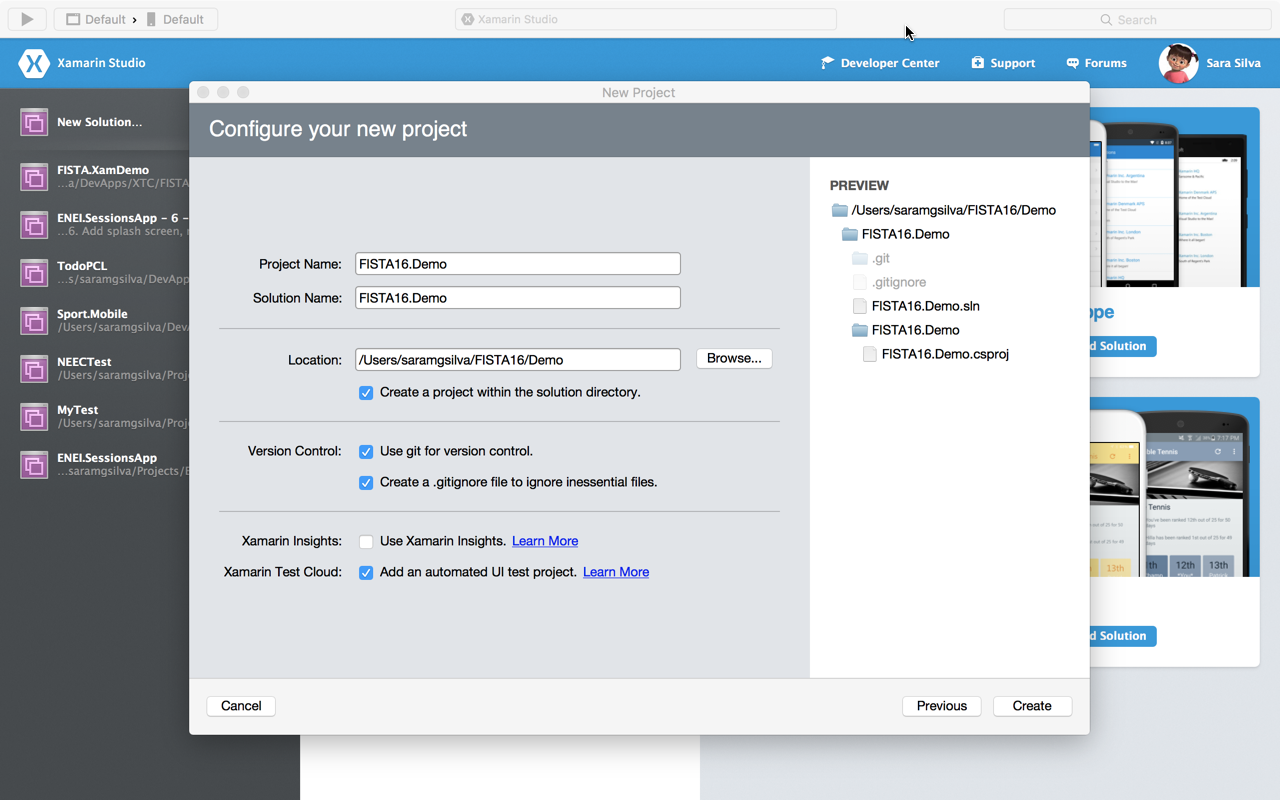
FISTA16/Xamarin Workshop – Lab: Creating a mobile application

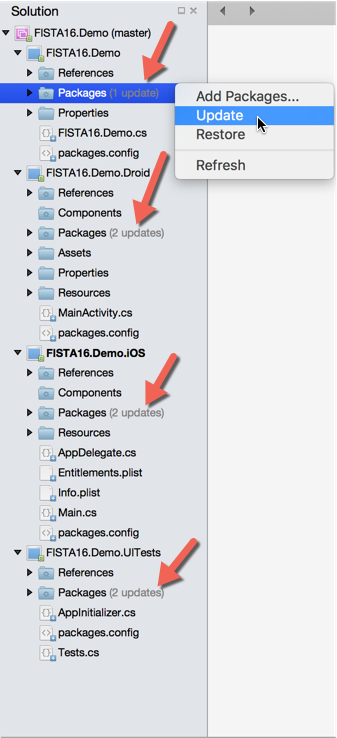
# 1. Creating the Xamarin Project







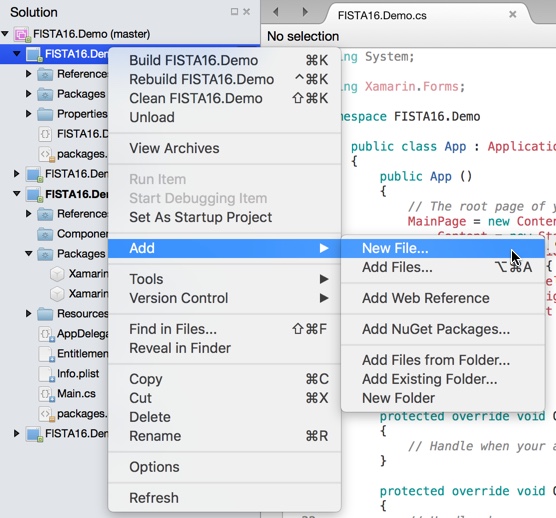
# 2. Update Nuget packages

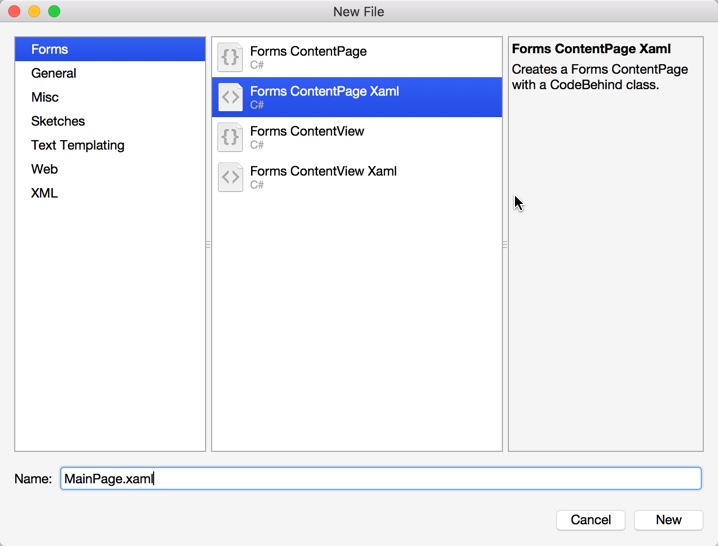


**Note**: Do not update the NUnit nuget package because Xamarin.UITest is not compatible with v3.

# 3. Creating the MainPage using XAML

# 3.1. Adding a new XAML page





Rename the file FISTA16.Demo.cs to App.cs, and after it remove the code that defines the MainPage (defined int the App constructor) and add the following code, which define the MainPage created as the MainPage of the application:

public App ()  
        {  
            *// The root page of your application*  
            MainPage = new NavigationPage (new MainPage ())   
            {  
                BarBackgroundColor = Color.White,  
                BarTextColor = Color.Black,  
                BackgroundColor = Color.White,  
            };  
        }

# 3.2. Defining the UI

## XAML

<?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="FISTA16.Demo.MainPage"  
         StyleId="UserProfilePage"  
        Title="User Profile Form"  
        BackgroundColor="White">  
 <ContentPage.Content>  
        <Grid Padding="20,10,20,0">  
            <Grid.RowDefinitions>  
                <RowDefinition Height="Auto" />  
                <RowDefinition Height="Auto" />  
                <RowDefinition Height="Auto" />  
                <RowDefinition Height="Auto" />  
                <RowDefinition Height="Auto" />  
                <RowDefinition Height="Auto" />  
                <RowDefinition Height="20" />  
                <RowDefinition Height="\*" />  
            </Grid.RowDefinitions>  
  
            <Label Grid.Row="0" Text="Name:" TextColor="Black">  
                <Label.FontSize>  
                    <OnPlatform Android="22"  
                                WinPhone="18"  
                                iOS="20"  
                                x:TypeArguments="x:Double" />  
                </Label.FontSize>  
            </Label>  
  
            <Entry x:Name="NameEntry"  
                   Grid.Row="1" TextColor="Black"  
                   StyleId="NameEntry"  
                   TextChanged="MobileNumberEntry\_OnTextChanged"  
                   VerticalOptions="CenterAndExpand" />  
  
            <Label Grid.Row="3" Text="Mobile Number:" TextColor="Black">  
                <Label.FontSize>  
                    <OnPlatform Android="22"  
                                WinPhone="18"  
                                iOS="20"  
                                x:TypeArguments="x:Double" />  
                </Label.FontSize>  
            </Label>  
  
            <Entry x:Name="MobileNumberEntry"  
                   Grid.Row="4" TextColor="Black"  
                   StyleId="MobileNumberEntry"  
                   TextChanged="MobileNumberEntry\_OnTextChanged"  
                   VerticalOptions="CenterAndExpand" />  
  
            <Button Grid.Row="5"  
                    StyleId="CheckUser"  
                    Clicked="Button\_OnClicked"  
                    Text="Check User" />  
  
            <Label x:Name="FeedbackLabel"  
                   Grid.Row="7" TextColor="Black"  
                   HorizontalOptions="StartAndExpand"  
                   HorizontalTextAlignment="Center"  
                   StyleId="FeedbackLabel"  
                   VerticalOptions="CenterAndExpand">  
                <Label.FontSize>  
                    <OnPlatform Android="22"  
                                WinPhone="18"  
                                iOS="20"  
                                x:TypeArguments="x:Double" />  
                </Label.FontSize>  
            </Label>  
        </Grid>  
    </ContentPage.Content>  
</ContentPage>

## C#

using System;  
using System.Collections.Generic;  
  
using Xamarin.Forms;  
  
namespace FISTA16.Demo  
{  
    public partial class MainPage : ContentPage  
    {  
        public MainPage ()  
        {  
            InitializeComponent ();  
        }  
  
        private void Button\_OnClicked(object sender, EventArgs e)  
        {  
            string message;  
            var isValid = IsUserValid(NameEntry.Text, MobileNumberEntry.Text, out message);  
  
            if (isValid)  
            {  
                FeedbackLabel.BackgroundColor = Color.Green;  
                FeedbackLabel.TextColor = Color.White;  
                FeedbackLabel.Text = "User valid!";  
            }  
            else  
            {  
                FeedbackLabel.BackgroundColor= Color.Red;  
                FeedbackLabel.TextColor= Color.White;  
                FeedbackLabel.Text = message;  
            }  
        }  
  
        public bool IsUserValid(string name, string mobileNumber, out string message)  
        {  
            var isValid = true;  
            message = string.Empty;  
            int value = 0;  
  
            if (string.IsNullOrWhiteSpace(name))  
            {  
                message = "The user must have a name. ";  
                isValid = false;  
            }  
  
            if (string.IsNullOrWhiteSpace(mobileNumber))  
            {  
                message += "The mobile number must be defined.";  
                isValid = false;  
            }  
            else if (!int.TryParse(mobileNumber, out value))  
            {  
                message += "The mobile number only accept numbers.";  
                isValid = false;  
            }  
            else if (mobileNumber.Length < 9)  
            {  
                message += "The mobile number is too short.";  
                isValid = false;  
            }  
            else if (mobileNumber.Length > 9)  
            {  
                message += "The mobile number is too long.";  
                isValid = false;  
            }  
  
            return isValid;  
        }  
  
        private void MobileNumberEntry\_OnTextChanged(object sender, TextChangedEventArgs e)  
        {  
            if (string.IsNullOrEmpty(MobileNumberEntry.Text) || string.IsNullOrEmpty(NameEntry.Text))  
            {  
                FeedbackLabel.BackgroundColor = Color.White;  
                FeedbackLabel.TextColor = Color.Black;  
                FeedbackLabel.Text = string.Empty;  
            }  
        }  
    }  
  
}

# 4. Defining the UI Acceptance Tests

## 4.1 Subscribing the *Forms.ViewInitialized* event

- iOS

Forms.ViewInitialized += (object sender, ViewInitializedEventArgs e) =>   
            {  
                if (null != e.View.StyleId) {  
                    e.NativeView.AccessibilityIdentifier = e.View.StyleId;  
                }  
            };

- Android

Xamarin.Forms.Forms.ViewInitialized += (object sender, Xamarin.Forms.ViewInitializedEventArgs e) =>   
            {  
                if (!string.IsNullOrWhiteSpace(e.View.StyleId)) {  
                    e.NativeView.ContentDescription = e.View.StyleId;  
                }  
            };

## 4.2 Configurating the app to test

* iOS

if (platform == Platform.iOS)

 {  
            return ConfigureApp  
                .iOS  
                .EnableLocalScreenshots()  
                *//***TODO***: Update this path to point to your iOS app and uncomment the*  
                *// code if the app is not included in the solution.*  
                *//.AppBundle ("../../../iOS/bin/iPhoneSimulator/Debug/XamarinForms.iOS.app")*  
                    .StartApp ();

}

* Android

if (platform == Platform.Android) {  
                return ConfigureApp  
                    .Android  
                    .EnableLocalScreenshots()  
                    *//***TODO***: Update this path to point to your Android app and uncomment the*  
                    *// code if the app is not included in the solution.*  
                    *//.ApkFile ("../../../Droid/bin/Debug/xamarinforms.apk")*  
                    .StartApp (); }

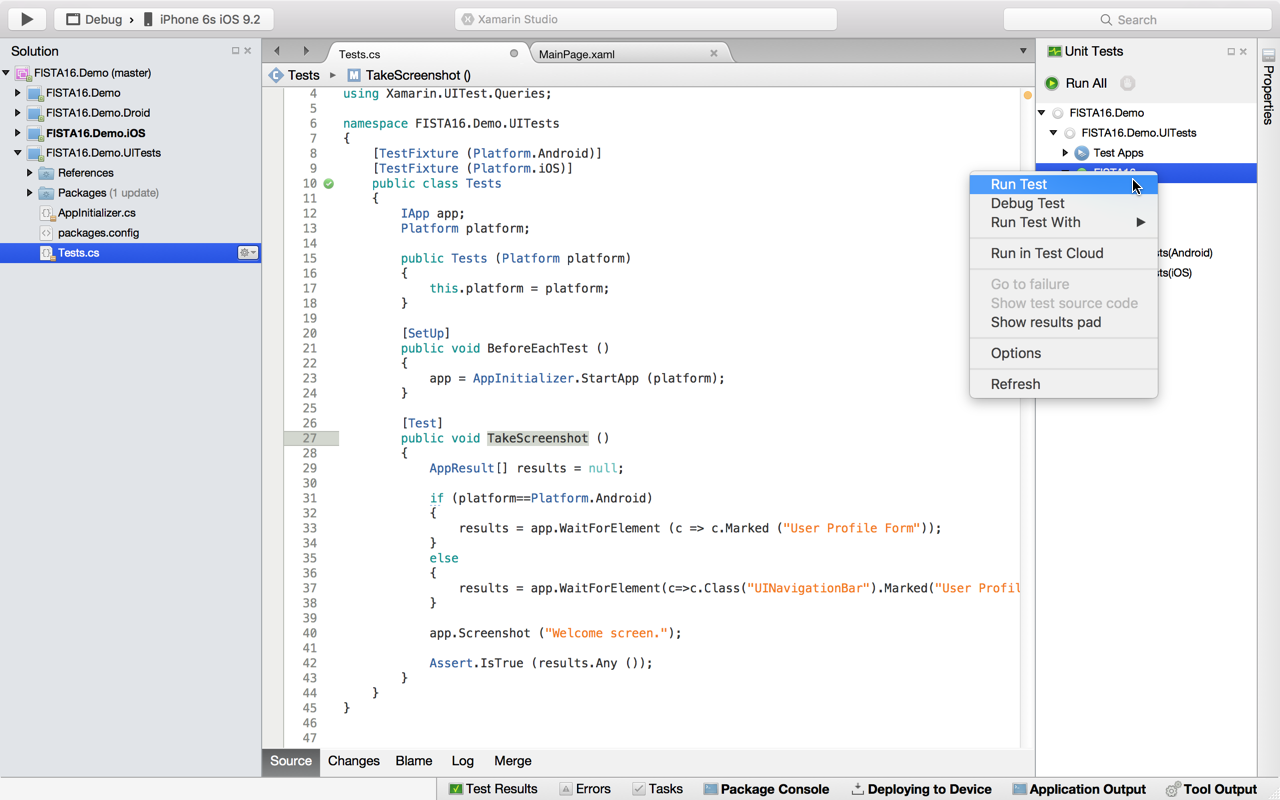
It is a requirement to add the Internet capability to the Android project

## 4.3. Taking a screenshot

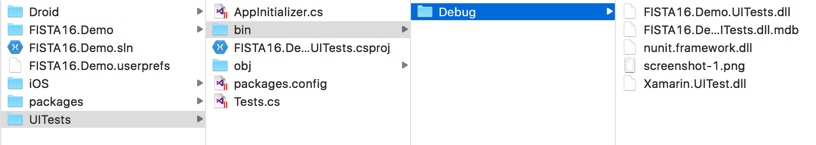
Open the Tests.cs file and change the code from the method WelcomeTextIsDisplayed  to

[Test]  
        public void TakeScreenshot()  
        {  
            AppResult[] results = null;  
  
            if (platform==Platform.Android)   
            {  
                results = app.WaitForElement (c => c.Marked ("User Profile Form"));  
            }   
            else   
            {  
                results = app.WaitForElement(c=>c.Class("UINavigationBar").Marked("User Profile Form"));  
            }  
  
            app.Screenshot ("Welcome screen.");  
  
            Assert.IsTrue (results.Any ());  
        }

Run the test for both targets



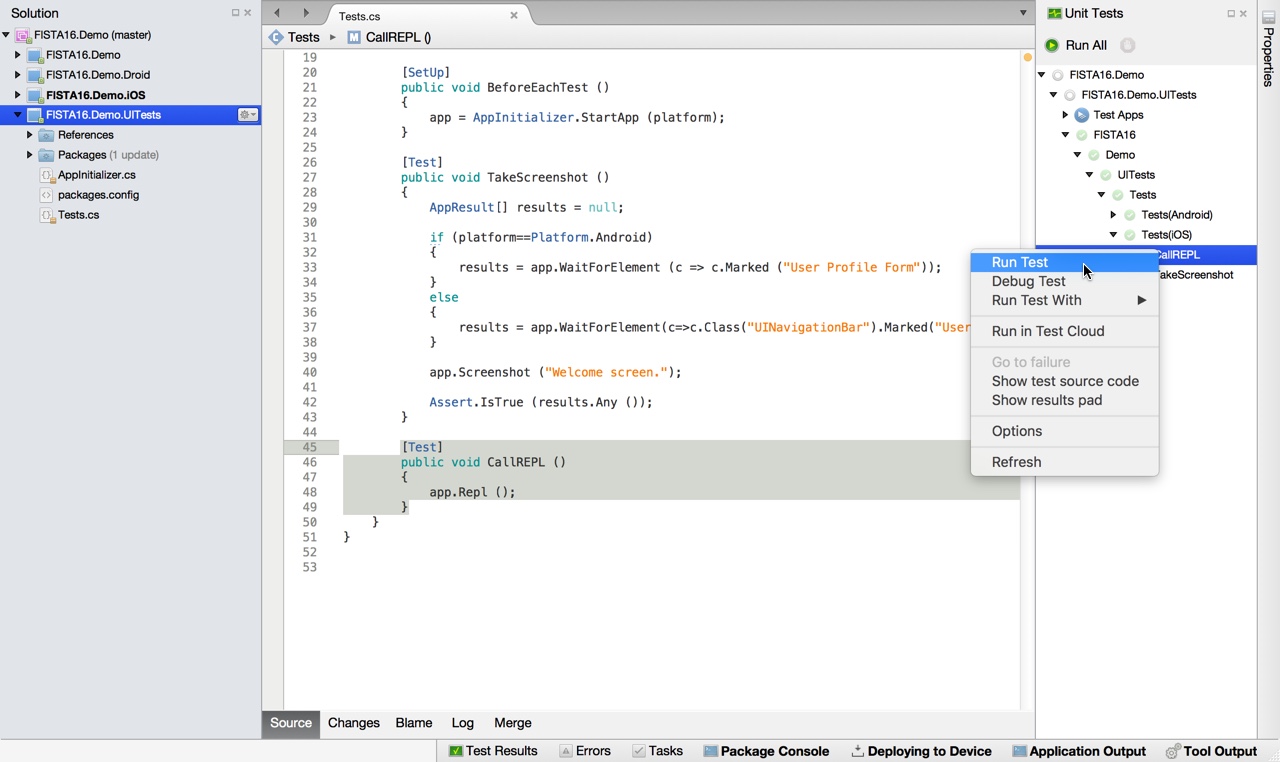
The screenshot can be find inside the bin folder as following

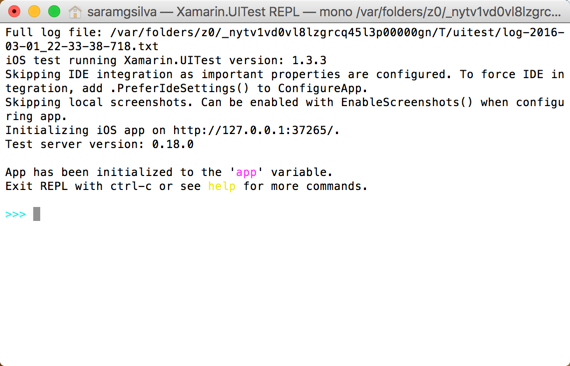


## 4.4. Showing the REPL window

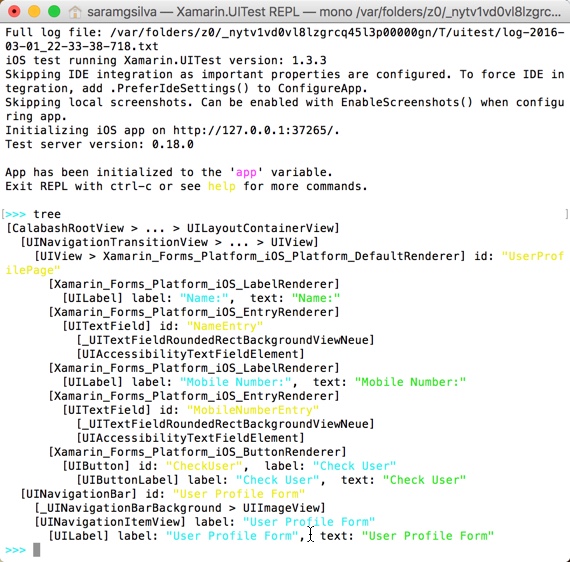
Selecting only one target, run a test to call the REPL Window and interact with the application

[Test]  
        public void CallREPL ()  
        {  
            app.Repl ();  
        }





write tree to see the tree defined by the controls



## 4.5. Defining tests

* iOS

[TestFixture (Platform.iOS)]

public partial class Tests\_iOS

{

IApp app;

Platform platform;

public Tests\_iOS (Platform platform)

{

this.platform = platform;

}

[SetUp]

public void BeforeEachTest ()

{

app = AppInitializer.StartApp (platform);

}

[Test]

public void InvokeREPL()

{

// Invoke the REPL so that we can explore the user interface

app.Repl();

}

[Test]

public void ScreenshotSample ()

{

app.Screenshot ("First screen.");

}

[Test]

public void VerifyUserProfile\_UserValid()

{

app.WaitForElement(c=>c.Class("UINavigationBar").Marked("User Profile Form"));

app.EnterText(c => c.Class("UITextField").Marked("NameEntry"),"Sara Silva");

app.EnterText(c => c.Class("UITextField").Marked("MobileNumberEntry"),"123456789");

app.Tap(c=>c.Marked("CheckUser").Class("UIButton"));

app.WaitForElement (c => c.Marked ("User valid!").Class ("UILabel").Marked("FeedbackLabel"));

}

[Test]

public void VerifyUserProfile\_MobileNumberIsTooLong()

{

app.WaitForElement(c=>c.Class("UINavigationBar").Marked("User Profile Form"));

app.EnterText(c => c.Class("UITextField").Marked("NameEntry"),"Sara Silva");

app.EnterText(c => c.Class("UITextField").Marked("MobileNumberEntry"),"1234567890");

app.Tap(c=>c.Marked("CheckUser").Class("UIButton"));

app.WaitForElement (c => c.Marked ("The mobile number is too long.").Class ("UILabel").Marked("FeedbackLabel"));

}

[Test]

public void VerifyUserProfile\_MobileNumberIsTooShort()

{

app.WaitForElement(c=>c.Class("UINavigationBar").Marked("User Profile Form"));

app.EnterText(c => c.Class("UITextField").Marked("NameEntry"),"Sara Silva");

app.EnterText(c => c.Class("UITextField").Marked("MobileNumberEntry"),"12345678");

app.Tap(c=>c.Marked("CheckUser").Class("UIButton"));

app.WaitForElement (c => c.Marked ("The mobile number is too short.").Class ("UILabel").Marked("FeedbackLabel"));

}

[Test]

public void VerifyUserProfile\_MobileNumberOnlyAcceptNumbers()

{

app.WaitForElement(c=>c.Class("UINavigationBar").Marked("User Profile Form"));

app.EnterText(c => c.Class("UITextField").Marked("NameEntry"),"Sara Silva");

app.EnterText(c => c.Class("UITextField").Marked("MobileNumberEntry"),"ABC");

app.Tap(c=>c.Marked("CheckUser").Class("UIButton"));

app.WaitForElement (c => c.Marked ("The mobile number only accept numbers.").Class ("UILabel").Marked("FeedbackLabel"));

}

[Test]

public void VerifyUserProfile\_MobileNumberMustBeDefined()

{

app.WaitForElement(c=>c.Class("UINavigationBar").Marked("User Profile Form"));

app.EnterText(c => c.Class("UITextField").Marked("NameEntry"),"Sara Silva");

app.Tap(c=>c.Marked("CheckUser").Class("UIButton"));

app.WaitForElement (c => c.Marked ("The mobile number must be defined.").Class ("UILabel").Marked("FeedbackLabel"));

}

[Test]

public void VerifyUserProfile\_UserMustHaveName()

{

app.WaitForElement(c=>c.Class("UINavigationBar").Marked("User Profile Form"));

app.EnterText(c => c.Class("UITextField").Marked("MobileNumberEntry"),"123456789");

app.Tap(c=>c.Marked("CheckUser").Class("UIButton"));

app.WaitForElement (c => c.Marked ("The user must have a name. ").Class ("UILabel").Marked("FeedbackLabel"));

}

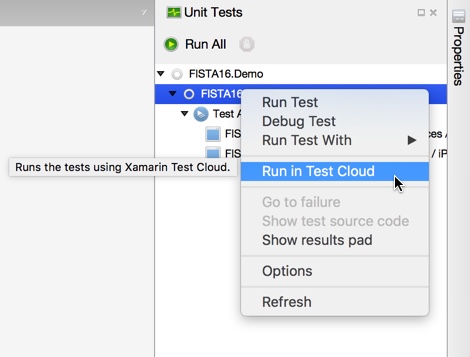
}

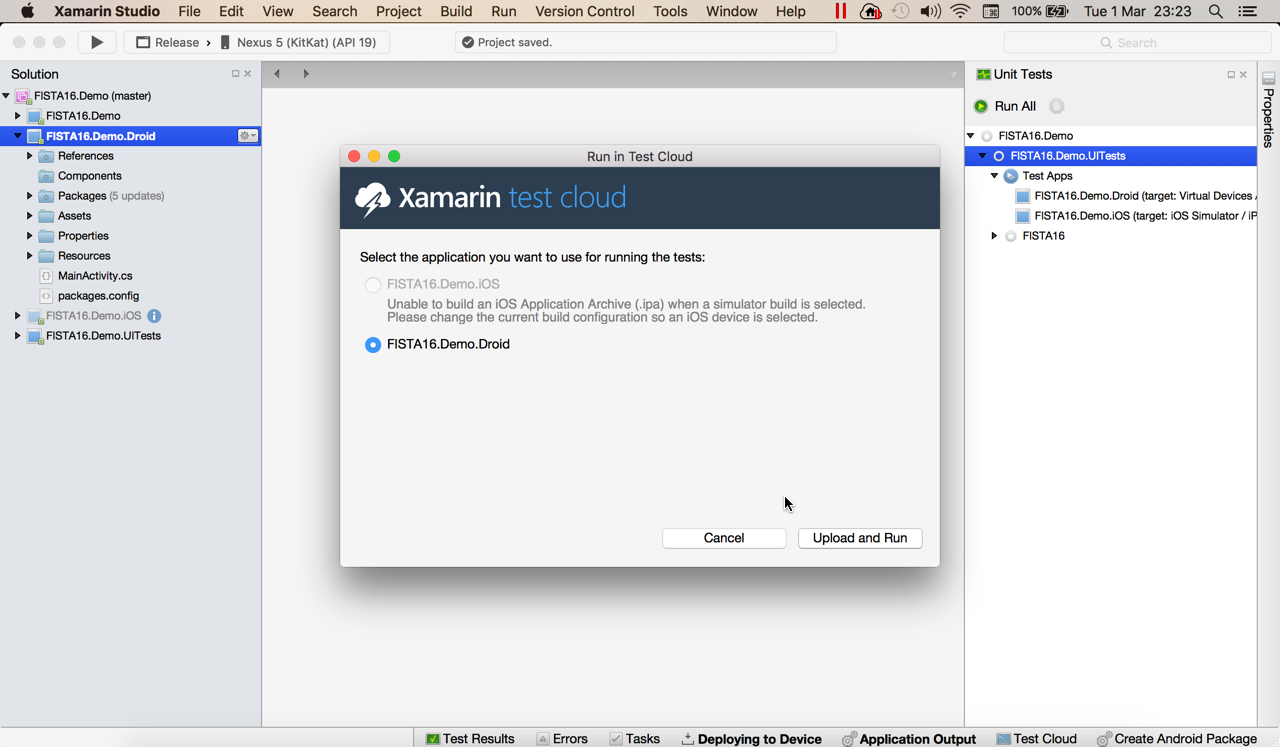
* Android

[TestFixture (Platform.Android)]  
    public partial class Tests\_Android  
    {  
        IApp app;  
        Platform platform;  
  
        public Tests\_Android (Platform platform)  
        {  
            this.platform = platform;  
        }  
  
        [SetUp]  
        public void BeforeEachTest ()  
        {  
            app = AppInitializer.StartApp (platform);  
        }  
  
        [Test]  
        public void VerifyUserProfile\_UserValid()  
        {  
            app.WaitForElement(c => c.Marked("action\_bar\_title").Text("User Profile Form"));  
  
            app.EnterText(c => c.Marked("NameEntry"),"Sara Silva");  
  
            app.EnterText(c => c.Marked("MobileNumberEntry"),"123456789");  
  
            app.Tap(c=>c.Marked("CheckUser"));  
  
            app.WaitForElement(c => c.Marked("FeedbackLabel").Text("User valid!"));  
        }  
  
        [Test]  
        public void VerifyUserProfile\_MobileNumberIsTooLong()  
        {  
  
            app.WaitForElement(c => c.Marked("action\_bar\_title").Text("User Profile Form"));  
  
            app.EnterText(c => c.Marked("NameEntry"),"Sara Silva");  
  
            app.EnterText(c => c.Marked("MobileNumberEntry"),"1234567890");  
  
            app.Tap(c=>c.Marked("CheckUser"));  
  
            *//var query= app.Query(c => c.Marked("FeedbackLabel"));*  
  
            app.WaitForElement(c => c.Marked("FeedbackLabel").Text("The mobile number is too long."));  
        }  
  
        [Test]  
        public void VerifyUserProfile\_MobileNumberIsTooShort()  
        {  
  
            app.WaitForElement(c => c.Marked("action\_bar\_title").Text("User Profile Form"));  
  
            app.EnterText(c => c.Marked("NameEntry"),"Sara Silva");  
  
            app.EnterText(c => c.Marked("MobileNumberEntry"),"12345678");  
  
            app.Tap(c=>c.Marked("CheckUser"));  
  
            app.WaitForElement(c => c.Marked("FeedbackLabel").Text("The mobile number is too short."));  
        }  
  
        [Test]  
        public void VerifyUserProfile\_MobileNumberOnlyAcceptNumbers()  
        {  
  
            app.WaitForElement(c => c.Marked("action\_bar\_title").Text("User Profile Form"));  
  
            app.EnterText(c => c.Marked("NameEntry"),"Sara Silva");  
  
            app.EnterText(c => c.Marked("MobileNumberEntry"),"ABC");  
  
            app.Tap(c=>c.Marked("CheckUser"));  
  
            app.WaitForElement(c => c.Marked("FeedbackLabel").Text("The mobile number only accept numbers."));  
        }  
  
        [Test]  
        public void VerifyUserProfile\_MobileNumberMustBeDefined()  
        {  
  
            app.WaitForElement(c => c.Marked("action\_bar\_title").Text("User Profile Form"));  
  
            app.EnterText(c => c.Marked("NameEntry"),"Sara Silva");  
  
            app.Tap(c=>c.Marked("CheckUser"));  
  
            app.WaitForElement(c => c.Marked("FeedbackLabel").Text("The mobile number must be defined."));  
        }  
  
        [Test]  
        public void VerifyUserProfile\_UserMustHaveName()  
        {  
  
            app.WaitForElement(c => c.Marked("action\_bar\_title").Text("User Profile Form"));  
  
            app.EnterText(c => c.Marked("MobileNumberEntry"),"123456789");  
  
            app.Tap(c=>c.Marked("CheckUser"));  
  
            app.WaitForElement(c => c.Marked("FeedbackLabel").Text("The user must have a name. "));  
        }

# 5. Running tests in Xamarin Test Cloud

Select the Android project as startup project, select Release mode.





The browser will be called

https://testcloud.xamarin.com/upload/8562e067-42de-46de-89b7-46e2c135b63e

