**ZelleSDK Integration for Xamarin**

1. Create a IZelleService.cs interface class inside Xamarin Solution.
2. This IZelleService.cs class will create a bridge between Xamarin Solution to Xamarin Android and IOS.



Code Snippet for IZelleService Interface class:

using System;

using System.Collections;

namespace FIApp

{

public interface IZelleService

{

void launchZell(String applicationName, String baseUrl, String institutionId, String ssoKey, String product, IDictionary data);

}

}

1. In Xamarin solution launch the Zelle by passing the required parameters through interface by using DependencyService.



Code Snippet:

using System;

using Xamarin.Forms;

namespace FIApp

{

public partial class MainPage : ContentPage

{

public MainPage()

{

InitializeComponent();

}

public void Button\_Clicked(object sender, EventArgs args)

{

var callZelle = DependencyService.Get<IZelleService>();

if (callZelle != null)

{

callZelle.launchZell(ApplicationName.Text, BaseUrl.Text, InstitutionId.Text, SSOKey.Text, ProductId.Text, null);

}

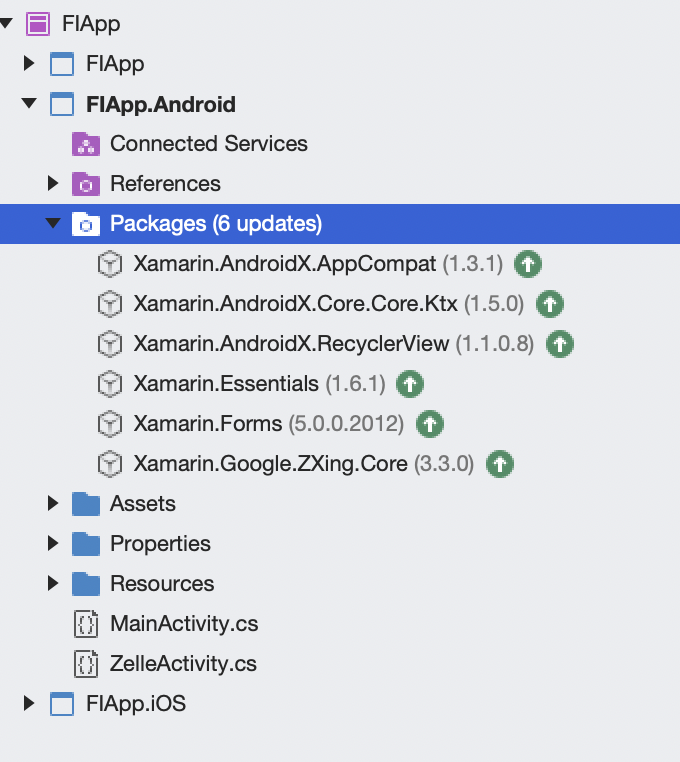
}

}

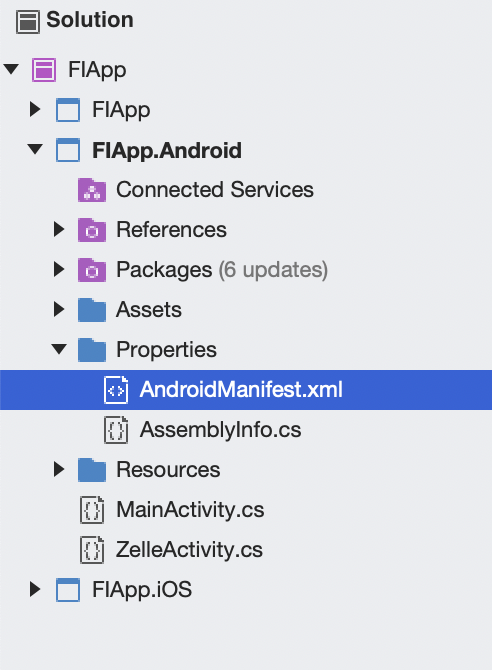
}

Xamarin Android:

1. Add the required library’s from NuGet for Xamarin android inside <AppName.android>.Packages.



1. Add the required permission and declare the respective class file path and tools:node="replace" inside application tag in Properties.AndroidManifest.xml file.



Code Snipper:

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:versionCode="1"

android:versionName="1.0"

package="com.fiserv.dps.mobile.fiapp">

<uses-sdk android:minSdkVersion="21" android:targetSdkVersion="30" />

<application android:label="FIApp.Android"

tools:node="replace"

android:theme="@style/MainTheme">

<activity android:name="com.fiserv.dps.mobile.sdk.activity.ScanQRActivity" />

<activity android:name="com.fiserv.dps.mobile.sdk.activity.ContactDetailActivity" />

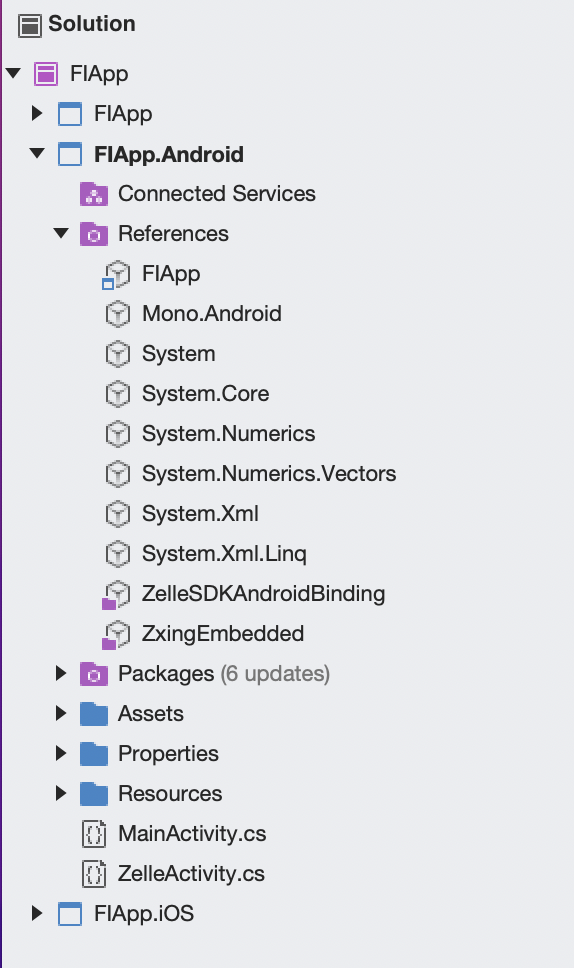
</application>

<uses-permission android:name="android.permission.INTERNET" />

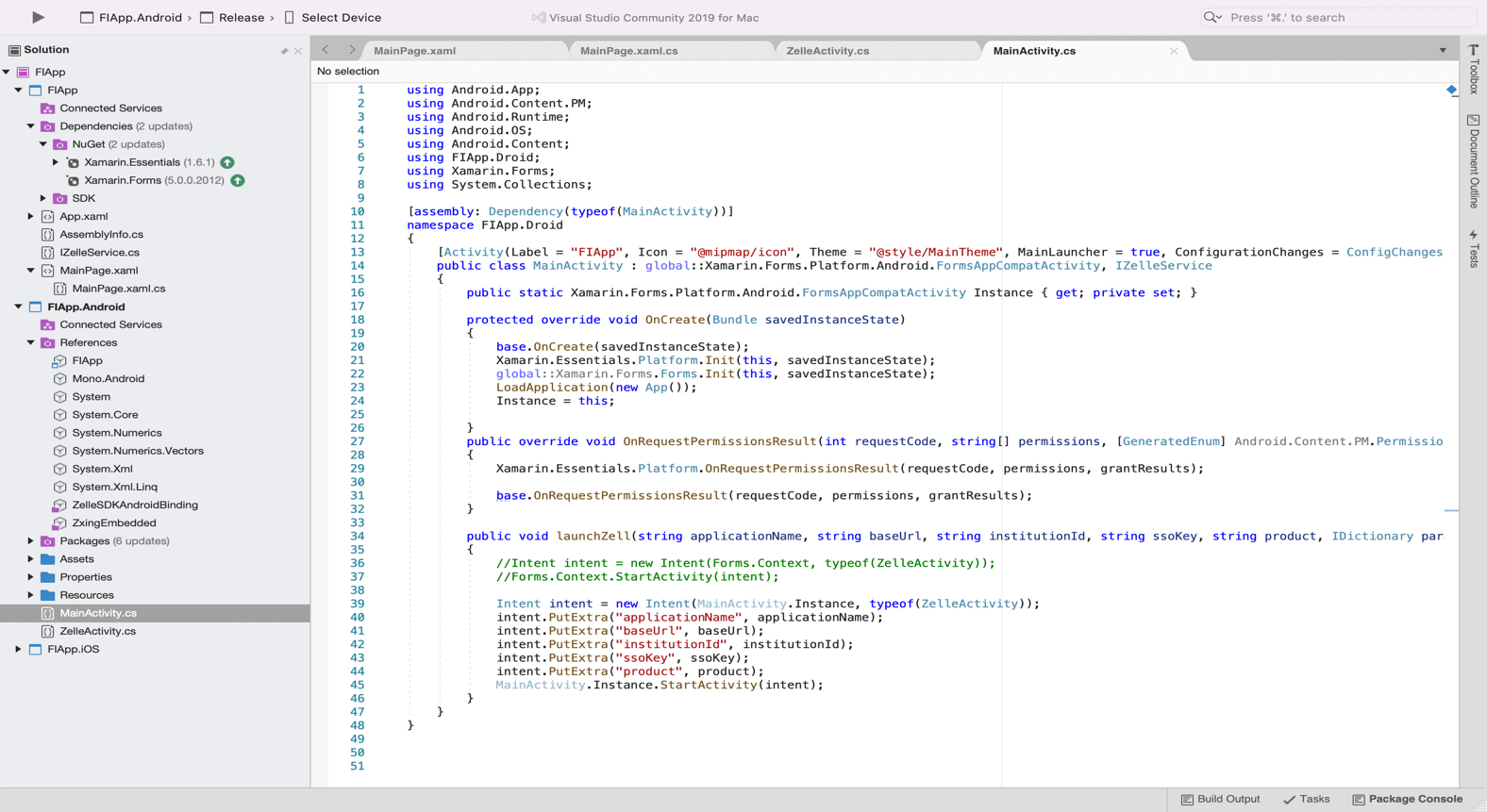
<uses-permission android:name="android.permission.CAMERA" />

</manifest>

1. Add the ZelleSDKAndroidBinding and ZxingEmbedded .dll file inside References folder.



1. Open the MainActivity.cs file and implement the interface class inside <ProjectName.android> folder.



1. Add the assembly for the interface communication connection from Xamarin solution to Xamarin android and ios.
2. This IZelleService.cs interface class will override launchZell function.

Code Snippet:

using Android.App;

using Android.Content.PM;

using Android.Runtime;

using Android.OS;

using Android.Content;

using FIApp.Droid;

using Xamarin.Forms;

using System.Collections;

[assembly: Dependency(typeof(MainActivity))]

namespace FIApp.Droid

{

[Activity(Label = "FIApp", Icon = "@mipmap/icon", Theme = "@style/MainTheme", MainLauncher = true, ConfigurationChanges = ConfigChanges.ScreenSize | ConfigChanges.Orientation | ConfigChanges.UiMode | ConfigChanges.ScreenLayout | ConfigChanges.SmallestScreenSize)]

public class MainActivity : global::Xamarin.Forms.Platform.Android.FormsAppCompatActivity, IZelleService

{

public static Xamarin.Forms.Platform.Android.FormsAppCompatActivity Instance { get; private set; }

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

Xamarin.Essentials.Platform.Init(this, savedInstanceState);

global::Xamarin.Forms.Forms.Init(this, savedInstanceState);

LoadApplication(new App());

Instance = this;

}

public override void OnRequestPermissionsResult(int requestCode, string[] permissions, [GeneratedEnum] Android.Content.PM.Permission[] grantResults)

{

Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions, grantResults);

base.OnRequestPermissionsResult(requestCode, permissions, grantResults);

}

public void launchZell(string applicationName, string baseUrl, string institutionId, string ssoKey, string product, IDictionary parameters)

{

//Intent intent = new Intent(Forms.Context, typeof(ZelleActivity));

//Forms.Context.StartActivity(intent);

Intent intent = new Intent(MainActivity.Instance, typeof(ZelleActivity));

intent.PutExtra("applicationName", applicationName);

intent.PutExtra("baseUrl", baseUrl);

intent.PutExtra("institutionId", institutionId);

intent.PutExtra("ssoKey", ssoKey);

intent.PutExtra("product", product);

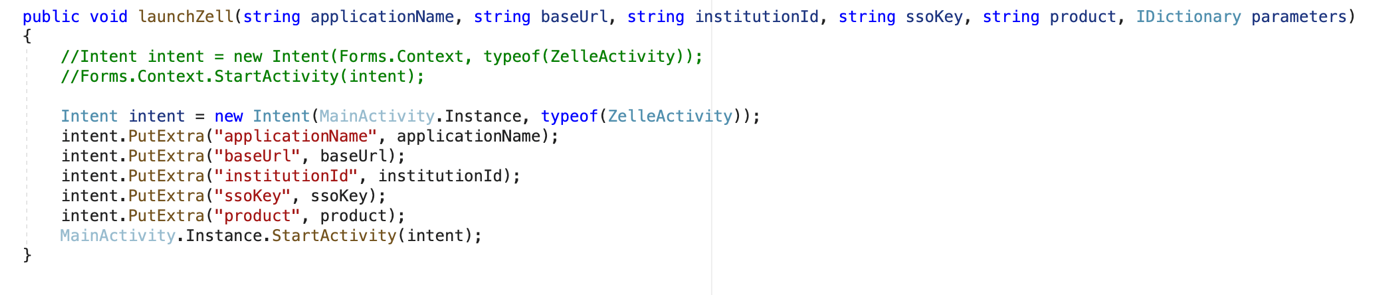
MainActivity.Instance.StartActivity(intent);

}

}

}

1. Get the values from launchZelle function and by using intent pass the values from one page to another.



Code Snippet:

public void launchZell(string applicationName, string baseUrl, string institutionId, string ssoKey, string product, IDictionary parameters)

{

//Intent intent = new Intent(Forms.Context, typeof(ZelleActivity));

//Forms.Context.StartActivity(intent);

Intent intent = new Intent(MainActivity.Instance, typeof(ZelleActivity));

intent.PutExtra("applicationName", applicationName);

intent.PutExtra("baseUrl", baseUrl);

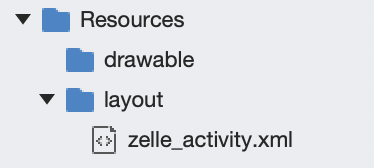
intent.PutExtra("institutionId", institutionId);

intent.PutExtra("ssoKey", ssoKey);

intent.PutExtra("product", product);

MainActivity.Instance.StartActivity(intent); }

1. Create XML file inside layout folder to design the framelayout to access Zelle.



Code Snippet for XML File:

<?xml version="1.0" encoding="UTF-8" ?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<FrameLayout

android:id="@+id/lay\_view"

android:layout\_width="match\_parent"

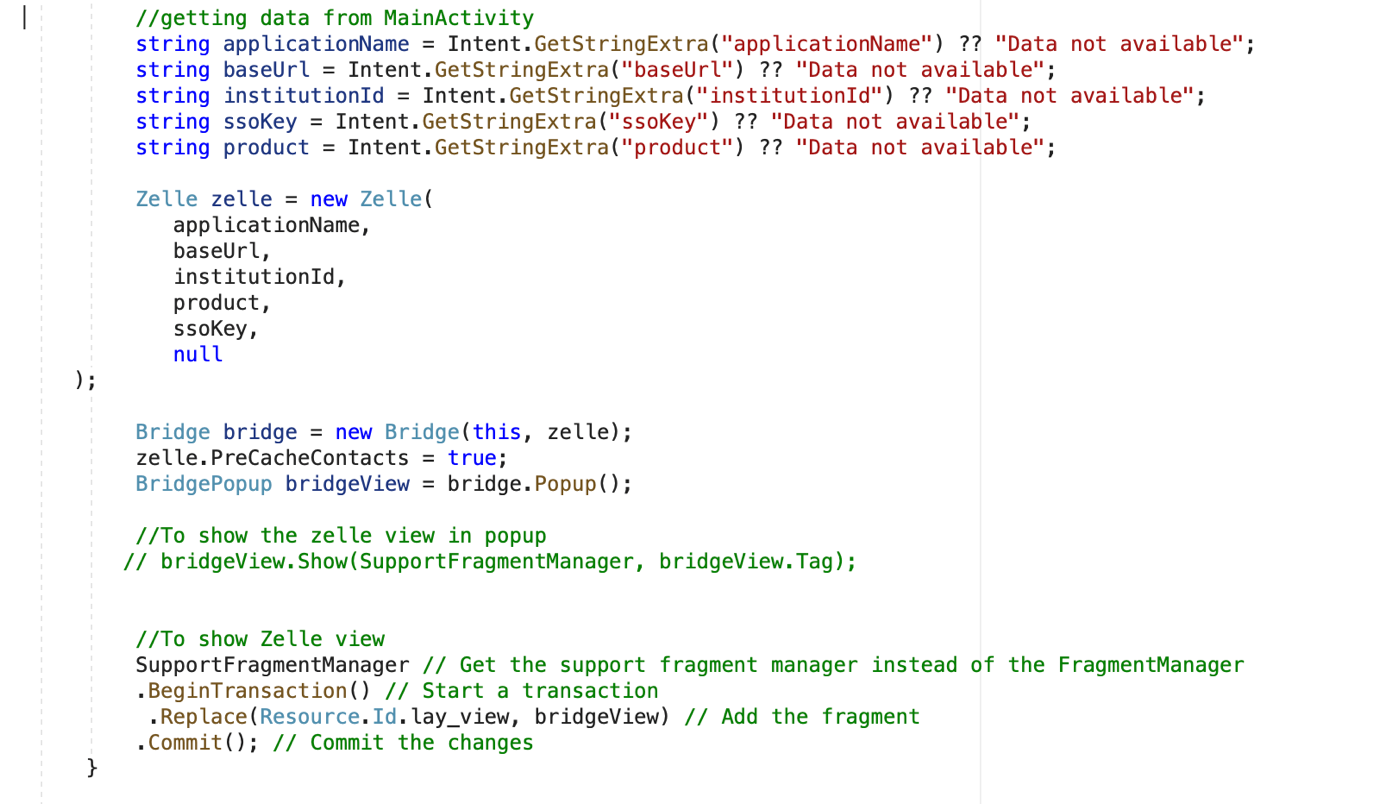
android:layout\_height="match\_parent" />

</LinearLayout>

1. Create a new class which extends AppCompatActivity and override OnCreate function.
2. Set the View for that class inside OnCreate function:

SetContentView(Resource.Layout.zelle\_activity);

1. Get the values from the previous activity using intent and initialize Zelle with respective params.



Code Snippet of Zelle Class:

using Android.App;

using Android.OS;

using AndroidX.AppCompat.App;

using Com.Fiserv.Dps.Mobile.Sdk.Bridge.Zelleview;

using Com.Fiserv.Dps.Mobile.Sdk.Bridge.Model;

using Com.Fiserv.Dps.Mobile.Sdk.Interfaces;

using Xamarin.Forms;

using Android.Widget;

using System.Collections;

namespace FIApp.Droid

{

[Activity(Label = "ZelleActivity", Theme = "@style/MainTheme", MainLauncher = true)]

public class ZelleActivity:AppCompatActivity, IGenericTag

{

public void SessionTag(string tag)

{

Toast.MakeText(Forms.Context, tag, ToastLength.Long).Show();

}

protected override void OnCreate(Bundle savedInstanceState)

{

base.OnCreate(savedInstanceState);

Xamarin.Essentials.Platform.Init(this, savedInstanceState);

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

SetContentView(Resource.Layout.zelle\_activity);

BridgeView.GenericTag = this;

//getting data from MainActivity

string applicationName = Intent.GetStringExtra("applicationName") ?? "Data not available";

string baseUrl = Intent.GetStringExtra("baseUrl") ?? "Data not available";

string institutionId = Intent.GetStringExtra("institutionId") ?? "Data not available";

string ssoKey = Intent.GetStringExtra("ssoKey") ?? "Data not available";

string product = Intent.GetStringExtra("product") ?? "Data not available";

Zelle zelle = new Zelle(

applicationName,

baseUrl,

institutionId,

product,

ssoKey,

null

);

Bridge bridge = new Bridge(this, zelle);

zelle.PreCacheContacts = true;

BridgePopup bridgeView = bridge.Popup();

//To show the zelle view in popup

// bridgeView.Show(SupportFragmentManager, bridgeView.Tag);

//To show Zelle view

SupportFragmentManager // Get the support fragment manager instead of the FragmentManager

.BeginTransaction() // Start a transaction

.Replace(Resource.Id.lay\_view, bridgeView) // Add the fragment

.Commit(); // Commit the changes

}

}

}