Integration Steps for Xamarin Ios Application

Prerequisites:
      1. Get production .p12 file from apple developer account
      2. Get App Id from Smartech Panel

Note: Identity (used in this documentation) must be primary key which defined on the smartech panel for identified user and blank for anonymous users

Example: 1) If primary key is email Id then identity must be Email ID for logged in user
      2) If primary key is mobile number then identity must be Mobile number for logged in user
      3) If primary key is custom ID then it must be custom ID for logged in user

Steps for Integrations:

1.Create a Xamarin.iOS application with Visual Studio.
-Go to File => New Project=>iOS Project

5. Download .dll file from below github link and add to your project
https://github.com/NetcoreSolutions/Smartech-xamarin-ios-sdk.git

3.Add Following key inside 'info.plist'

```
<key>NSAppTransportSecurity</key>
<dict>
        <key>NSAllowsArbitraryLoads</key>
        <true/>
</dict>
```

-Reference for Details
https://developer.apple.com/library/content/documentation/General/Reference/InfoPlistKeyReference/Articles/CocoaKeys.html#//apple_ref/doc/uid/TP40009251-SW33

4.Add following code in 'FinishedLaunching' method of 'AppDelegate.cs' file

```
NetCoreSharedManager.SharedInstance().SetUpApplicationId("<App_ID>");

if (UIDevice.CurrentDevice.CheckSystemVersion(8, 0))
{
   var pushSettings = UIUserNotificationSettings.GetSettingsForTypes(
```

```
        UIUserNotificationType.Alert | UIUserNotificationType.Badge |
UIUserNotificationType.Sound, new NSSet());

        UIApplication.SharedApplication.RegisterUserNotificationSettings(pushSettings);
        UIApplication.SharedApplication.RegisterForRemoteNotifications();
    }
    else
    {
        UIRemoteNotificationType notificationTypes = UIRemoteNotificationType.Alert |
UIRemoteNotificationType.Badge |
UIRemoteNotificationType.Sound;

UIApplication.SharedApplication.RegisterForRemoteNotificationTypes(notificationTypes);
    }
```

5.Override RegisteredForRemoteNotifications **method by following method:**

```
    public override void RegisteredForRemoteNotifications(UIApplication application,
NSData deviceToken)
    {
    string strEmail = "<niranjan.shirke@netcore.co.in>";
    NetCoreSharedManager.SharedInstance().SetUpIdentity(<identity>);
    NetCoreSharedManager.SharedInstance().SetDeviceToken(deviceToken);
    NetCoreInstallation.SharedInstance().NetCorePushRegisteration(<identity>);
    }
```

6. Follow below code for registering device to cloud for push notification:
```
    NetCoreSharedManager.SharedInstance().SetUpIdentity(<identity>);
    NetCoreInstallation.SharedInstance().NetCorePushRegisteration(<identity>);
```

Note: 1. Above code get fire on every launch
         2. It is not sign up/ register call of application

7.Follow below lines for respective events :

-Login
NetCoreInstallation.SharedInstance().NetCorePushLogin(<identity>);

Example: NetCoreInstallation.SharedInstance().NetCorePushLogin("xyz@example.com");

-Logout:
NetCoreInstallation.SharedInstance().NetCorePushLogout();
```

**-Custom Activity:**
NetCoreAppTracking.SharedInstance().SendAppTrackingEvent(<Event ID>, <Custom payload>);

**Example:**
```
var keys = new object[] { "s^name", "i^id", "f^price" } ;
var values = new object[] { "Tester", 23, 9849 } ;
var payload = NSDictionary.FromObjectsAndKeys(values, keys);

var payload_details = new NSMutableArray<NSDictionary>();
payload_details.Add(new NSDictionary(payload));
```

NetCoreAppTracking.SharedInstance().SendAppTrackingEventWithCustomPayload(2,payload_details);

**-Profile Push:**
```
NetCoreInstallation.SharedInstance().NetCoreProfilePush(<identity>, <Profile details>);
```

**Example:**
```
var keys = new object[] { "NAME", "AGE", "MOBILE" } ;
var values = new object[] { "Tester", "23", "9898948849" } ;

var profile_details = NSDictionary.FromObjectsAndKeys(values, keys);

NetCoreInstallation.SharedInstance().NetCoreProfilePush("xyz@example.com",
```
profile_details);