Prerequisites & Integration Steps

- **Prerequisites:**

    1. **google-services.json** file from Firebase Console
    2. **Server Key** from Firebase Console
    3. **App Id** from Smartech panel

    Note: Minimum SDK version should be at least 16 (Jelly Bean) or above.

- **Integration Steps:**

    - **Adding google-services.json file**
        1. Find **<project_name>.csproj** file and add given snippet in the file

```
<ItemGroup>
   <GoogleServicesJson Include="google-services.json">
   <CopyToOutputDirectory>Always</CopyToOutputDirectory>
   </GoogleServicesJson>
</ItemGroup>
```

        2. Add **google-services.json** file inside the **project directory**.

    - **Adding required packages**
        1. Navigate to **'Packages'** folder and click on **'Add nuget packages'**.
        2. Select packages mentioned below from **'Nuget Package Manager'**.
            a. Xamarin.Firebase.Messaging
            b. GoogleGson
            c. Xamarin.Firebase.JobDispatcher
            d. Xamarin.GooglePlayServices.Base
            e. Xamarin.GooglePlayServices.Location

    - **Adding XamarinSmartechAndroid.dll file**
        1. Find **'XamarinSmartechAndroid.dll'** file from the github repository.
        2. Find **'References'** folder, right click on the same to click on **'Edit References'** from the menu.
        3. Navigate to **'.NetAssembly'** and find **'XamarinSmartechAndroid.dll'** to add the same.

    - **To register device for push notifications**

      To register the device for receiving push notifications from Smartech panel**,** add given snippet in **onCreate above base.onCreate in the Launching Activity** of the app.

```
NetcoreSDK.Register(this.Application, <app_id>, <unique_user_identity>);
```

- **To capture user login**

  To capture login activity of the user, add given snippet inside the activity/fragment when the user gets logged in successfully.

```
NetcoreSDK.Login(context, <unique_user_identity> );
```

- **To capture user logout**

  To capture logout activity of the user, add given snippet inside the activity/fragment when the user gets logged out successfully.

```
NetcoreSDK.Logout(context, <unique_user_identity> );
```

- **To capture custom activity**

  To capture custom activity performed by the user, add given snippet as per the requirement.

```
NetcoreSDK.Track(context, <unique_user_identity>,
        <event_id>, <payload>);
```

  e.g.

```
JSONObject jsonObject = new JSONObject();
JSONArray jsonArray = new JSONArray();
JSONObject payload = new JSONObject();
try {
        payload.put("name", "Nexus 5");
        payload.put("prid", 2);
        payload.put("price", 15000);
        payload.put("prqt", 1);
        jsonArray.put(payload);
        jsonObject.put("payload", jsonArray);
        NetcoreSDK.Track(context, <unique_user_identity>,
                <event_id>, jsonObject.ToString());
}
catch (JSONException e) { e.printStackTrace(); }
```

- **To capture user attributes**

  To capture and map user attributes, add given snippet as per the requirement.

```
NetcoreSDK.Profile(context, <unique_user_identity>, <profile_object>);
```

  e.g.

```
JSONObject profile = new JSONObject();
try {
        profile.Put("NAME", "John Doe");
        profile.Put("AGE", 35);
        profile.Put("MOBILE", 9876543210);
        NetcoreSDK.Profile(context,
                <unique_user_identity>, profile);
}
catch (JSONException e) { e.printStackTrace(); }
```

  Note: Use attribute name in capital letters as shown above.

- **To use custom push notification icon**

  SDK uses launcher icon for push notifications by default and in order to change it, use a custom icon by adding given snippet in the **Launching Activity of the app**.

```
NetcoreSDK.SetPushIcon(context, <path_to_drawable_icon>);
```

  e.g.

```
NetcoreSDK.SetPushIcon(context, Resource.Drawable.icon);
```

  Note: The notification icon being used should strictly be in **.png format** as per Google's guidelines. Preferable size for the push notification icons are mentioned below.

drawable-mdpi	: 24 x 24
drawable-hdpi	: 36 x 36
drawable-xhdpi	: 48 x 48
drawable-xxhdpi	: 72 x 72
drawable-xxxhdpi	: 96 x 96

- **To fetch delivered push notifications**

  To fetch delivered push notifications, add given snippet as per the requirement.

  ```
  JSONArray notifications = NetcoreSDK.GetNotifications(context);
  ```

  Note: The method **returns a 'JSONArray' of last 10 delivered push notifications** for the user.

- **To implement push notifications in existing FCM class**

  To use push notifications from Smartech panel along with existing set up of the FCM class, add given snippet **inside the FCM receiver class**.

  ```
  boolean pushFromSmartech =
      NetcoreSDK.HandleNotification(context, remoteMessage);
  ```

  Note: The method returns a **boolean** value.

  - Returns **true**, if the push notification is received from the Smartech panel. It will also render the push notification without any extra efforts further.

  - Return **false**, if the push notification is not received from the Smartech panel. In this case, handle the push notification at your end as per the requirement.

- **To opt out user from being tracked (GDPR Policy)**

  If the end user wants to opt out of being tracked, add given snippet as per the requirement.

  ```
  NetcoreSDK.OptOut(context, <boolean_flag>);
  ```

  Note: The method accepts a **boolean** value.

  - If an end user wants to opt out, the flag should be passed as **true**. Once the user opts out, SDK will not be able to track that particular user further and no communications will be received by that user.

  - If an end user wants to opt in, the flag should be passed as **false**. Once the user opts in, SDK will be able to track that particular user further and next communications will be received by that user.

- **To implement deep link in the application**
  To implement deep link in the application, add given snippet **inside corresponding activity of the app**.

```
[IntentFilter(new[] { Android.Content.Intent.ActionView },
DataScheme = "<scheme_name>",
DataHost = "<host_name>",
DataPathPrefix = "/",
Categories = new[] { Android.Content.Intent.CategoryDefault,
Android.Content.Intent.CategoryBrowsable })]
```

  e.g.

```
[IntentFilter(new[] { Android.Content.Intent.ActionView },
DataScheme = "http",
DataHost = "action.netcore.com",
DataPathPrefix = "/",
Categories = new[] { Android.Content.Intent.CategoryDefault,
Android.Content.Intent.CategoryBrowsable })]
```