# **Sample repository**

**Fcm.Xamarin.Android** - Firebase notifications for Xamarin Android

**Xamarin.Android** - Azure Notifications Hub for Xamarin Android

**Xamarin.Android.Summary.docx** - my report about this documentation

URL to report document for current documentation <https://github.com/PsyhoLord/AzureNotificationHub/blob/master/Xamarin.Android.Summary.docx>

Repo with my working sample: <https://github.com/PsyhoLord/AzureNotificationHub>

Commit named: "Fixed Azure Hub notifications" is what is missed in current documentation, why I couldn't send any notification to Android. Also, You can use this code as "[NotificationHubs app](https://github.com/Azure/azure-notificationhubs-samples/tree/master/dotnet/Xamarin/GetStartedXamarinAndroid) sample" which is not providing C# working solution.

# **Missed things**

## **AndroidManifest.xml**

Before line with: <application android.... starts

Need to insert permissions:

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

  <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />

  <uses-permission android:name="android.permission.WAKE\_LOCK" />

  <uses-permission android:name="android.permission.GET\_ACCOUNTS"/>

## **MainActivity.cs**

### **Check for Play Services**

Need to add a check for google play services, will prevent a lot of developer issues. Add method:

public bool IsPlayServicesAvailable()

{

    int resultCode = GoogleApiAvailability.Instance.IsGooglePlayServicesAvailable(this);

    if (resultCode != ConnectionResult.Success)

    {

        if (GoogleApiAvailability.Instance.IsUserResolvableError(resultCode))

            Log.Debug(TAG, GoogleApiAvailability.Instance.GetErrorString(resultCode));

        else

        {

            Log.Debug(TAG, "This device is not supported");

            Finish();

        }

        return false;

    }

    Log.Debug(TAG, "Google Play Services is available.");

    return true;

}

In method **OnCreate**, after lines with code

if (Intent.Extras != null)

{

    foreach (var key in Intent.Extras.KeySet())

    {

        if (key != null)

        {

            var value = Intent.Extras.GetString(key);

            Log.Debug(TAG, "Key: {0} Value: {1}", key, value);

        }

    }

}

add

IsPlayServicesAvailable();

### **Set channel ID**

Add to MainActivity: variable for channel id

internal static readonly string CHANNEL\_ID = "my\_notification\_channel";

## **MyFirebaseMessagingService.cs**

### **Update Notification.Builder**

Into **MyFirebaseMessagingService.cs**

Add usings:

using Android.OS;

using Android.Support.V4.App;

using Build = Android.OS.Build;

Method: SendNotification is using deprecated method

new Notification.Builder(this)

need to be updated with new one:

new NotificationCompat.Builder(this)

### **Fix for not showing notifications**

Add to construction of notification builder:

.SetShowWhen(false) // if remove notification comes to device, but not shown

## **Fix for Android SDK 26+**

### **MyFirebaseMessagingService.cs**

For new Android (8 and upper version) is required to set Channel.

So in file **MyFirebaseMessagingService.cs** in method **SendNotification**

after create notification builder need to add this lines:

if (Build.VERSION.SdkInt >= BuildVersionCodes.O)

{

    notificationBuilder.SetChannelId(MainActivity.CHANNEL\_ID);

}

### **MainActivity.cs**

Add method:

private void CreateNotificationChannel()

{

    if (Build.VERSION.SdkInt < BuildVersionCodes.O)

    {

        // Notification channels are new in API 26 (and not a part of the

        // support library). There is no need to create a notification

        // channel on older versions of Android.

        return;

    }

    var channelName = CHANNEL\_ID;

    var channelDescription = string.Empty;

    var channel = new NotificationChannel(CHANNEL\_ID, channelName, NotificationImportance.Default)

    {

        Description = channelDescription

    };

    var notificationManager = (NotificationManager)GetSystemService(NotificationService);

    notificationManager.CreateNotificationChannel(channel);

}

Call method in OnCreate after

IsPlayServicesAvailable();

CreateNotificationChannel();