

# Push-Notification mit Firebase in Xamarin-Form

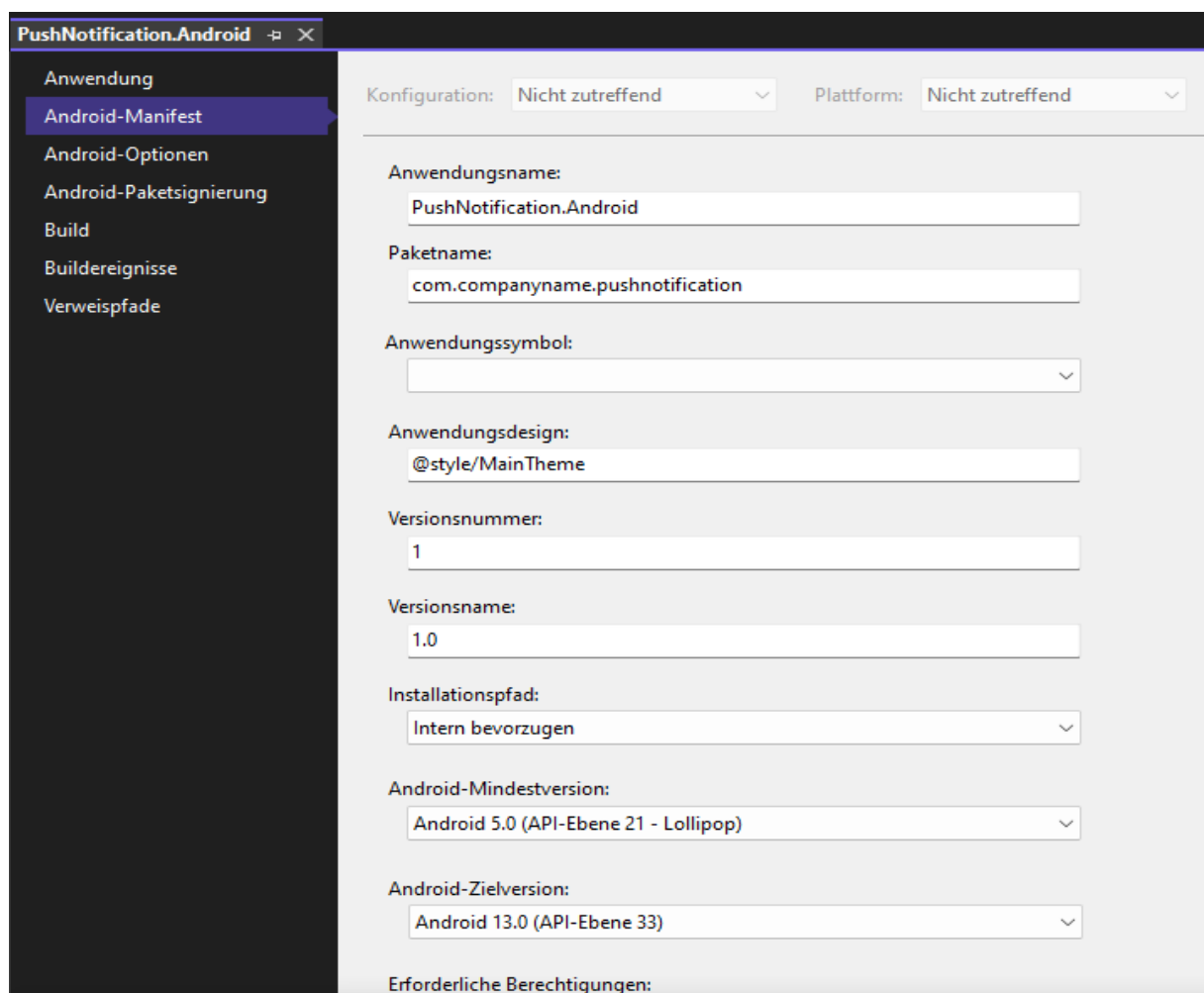
## Firestore

1. Ein Projekt erstellen unter <https://console.firebase.google.com>

2. App Hinzufügen mit **Paketname:**

Android Projekt - **Eigenschaften – Android-Manifest**

Sehe **Visual Studio** unten:



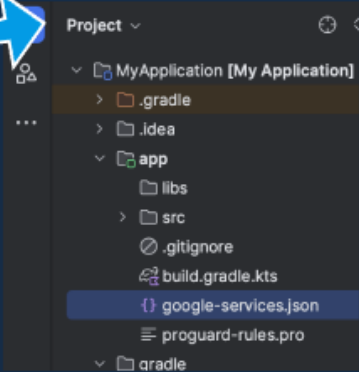
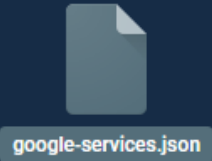
3. Lade **google-service.json** herunter. (für Android)

## 2 Konfigurationsdatei herunterladen und hinzufügen Anleitung für Android Studio unten | [Unity](#) | [C++](#)

[google-services.json herunterladen](#)

Wechseln Sie in Android Studio zur **Projektansicht**, um auf das Stammverzeichnis Ihres Projekts zuzugreifen.

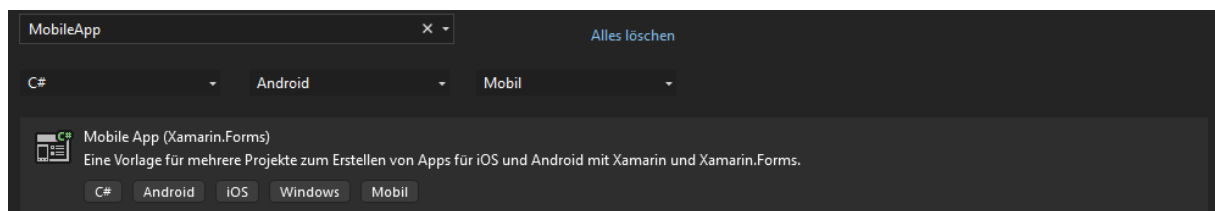
Verschieben Sie die heruntergeladene Datei `google-services.json` in das Modulstammverzeichnis (auf App-Ebene).



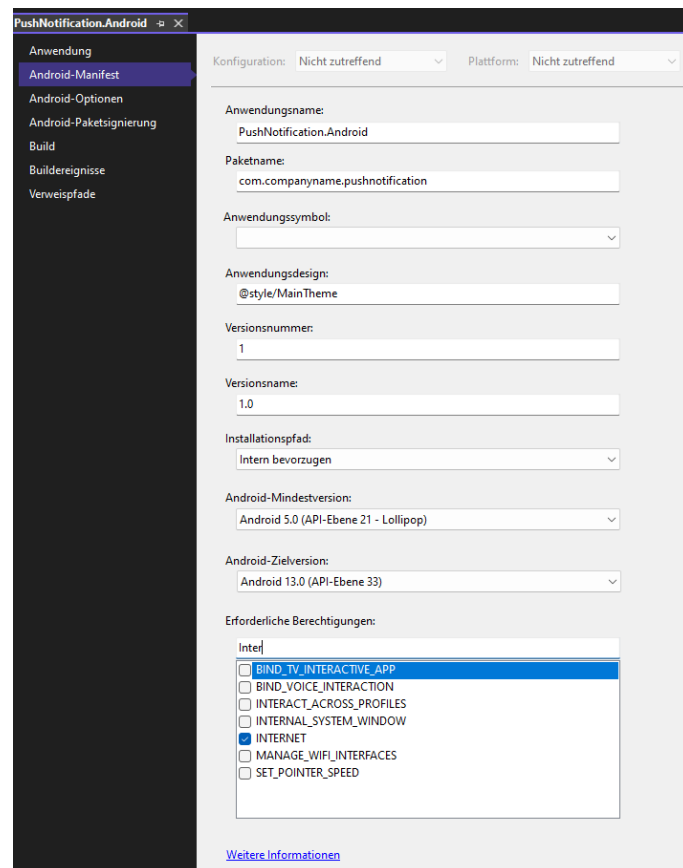
Weiter

## Visual Studio

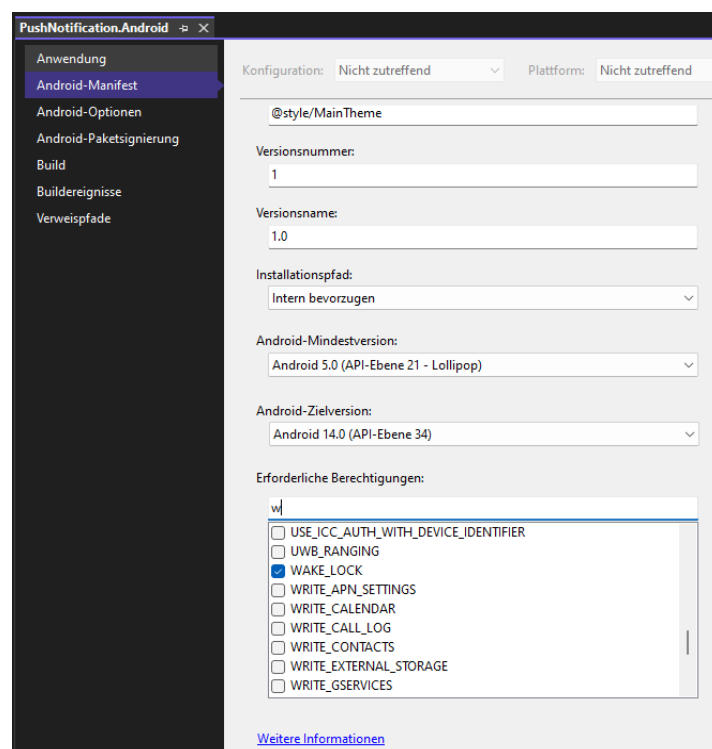
### 1. Projekt anlegen:



### 2. Android Projekt – **Eigenschaften - Erforderliche Berechtigungen:** **INTERNET** ankreuzen:

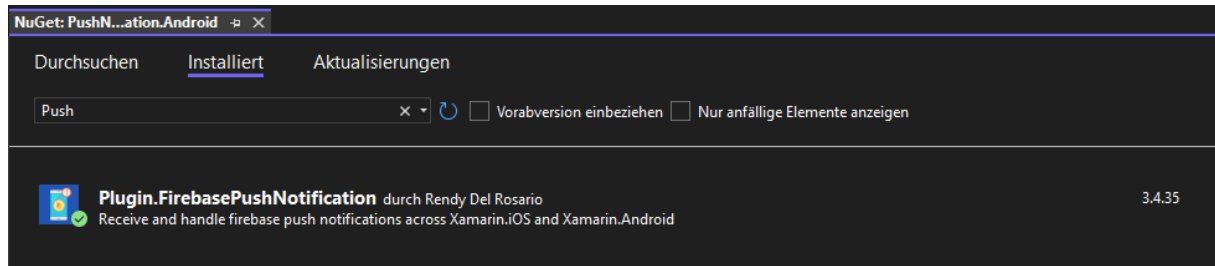


Android Projekt – **Eigenschaften – Android-Manifest - Erforderliche Berechtigungen: WAKE LOCK** ankreuzen:

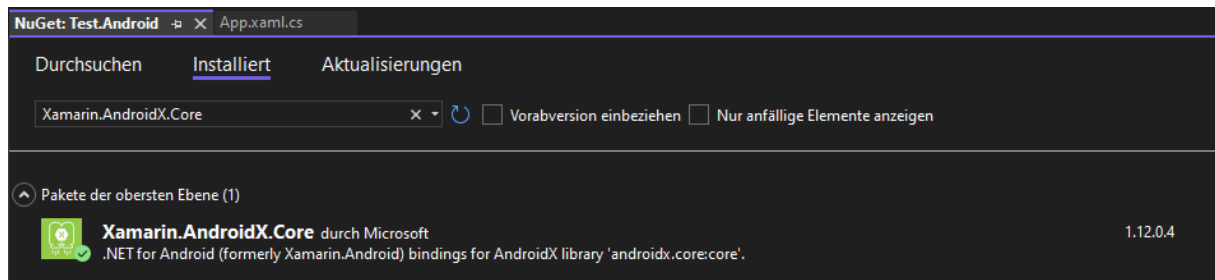
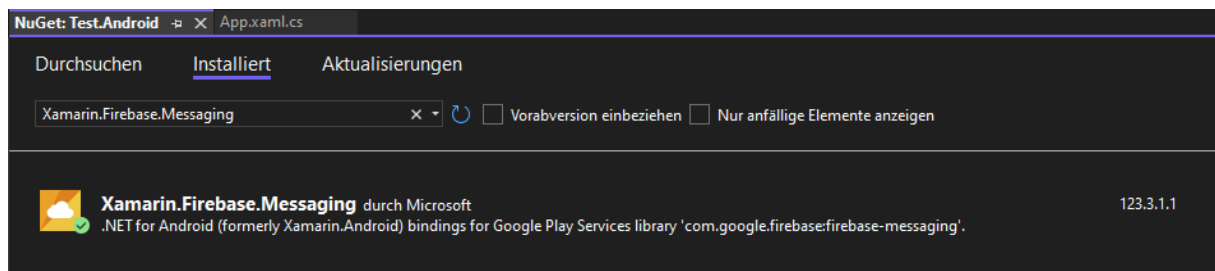
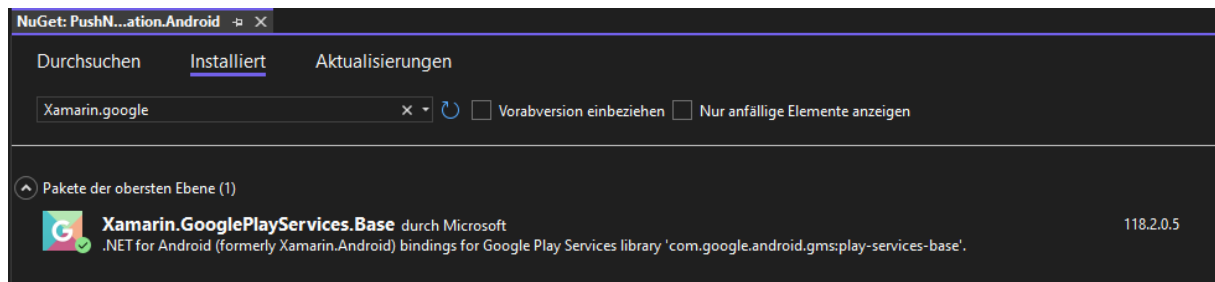


3. **NuGet** installieren:

## Hauptprojekt( C# ) – **NuGet-Pakete verwalten:** **Plugin.FirebasePushNotification** installieren:



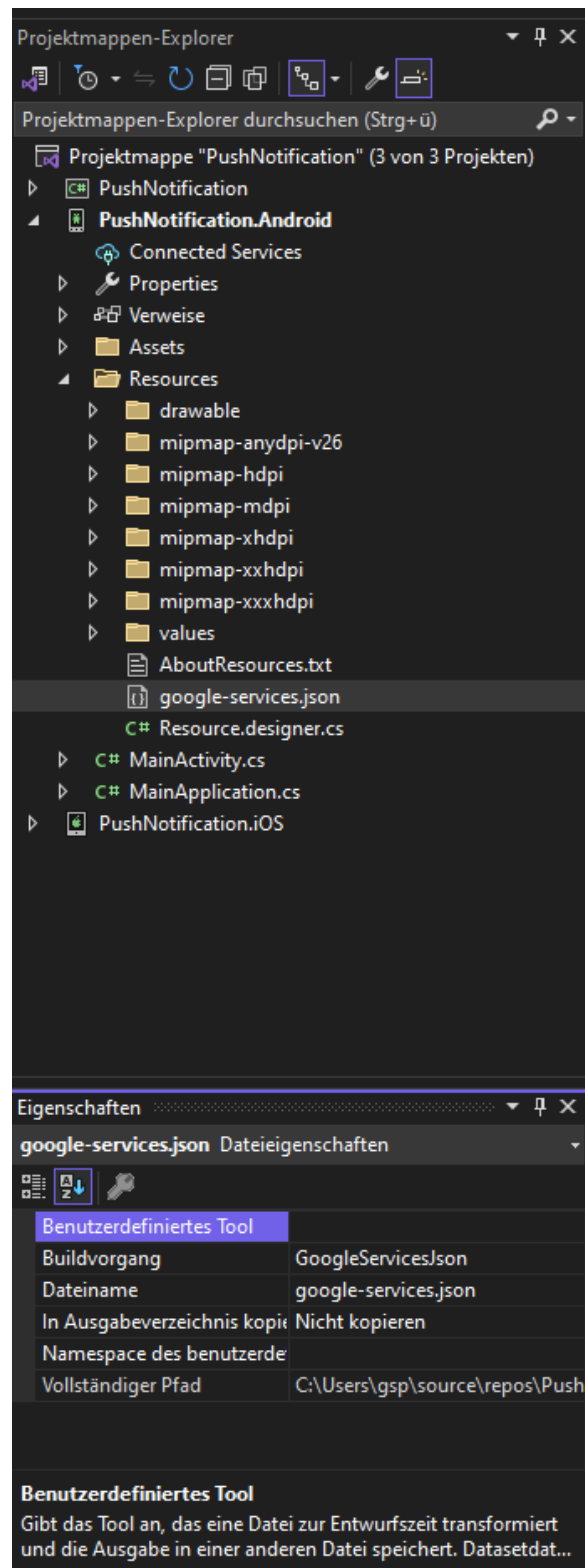
## Android Projekt - **NuGet-Pakete verwalten:**



Für **Xamarin.AndroidX.Core** ggf. weitere **NuGets** erforderlich.

## 4. Konfiguration

In Android Projekt die Datei **google-Service.json** im Ordner Resources kopiert und in Eigenschaften **Buildvorgang - Google ServicesJson**.



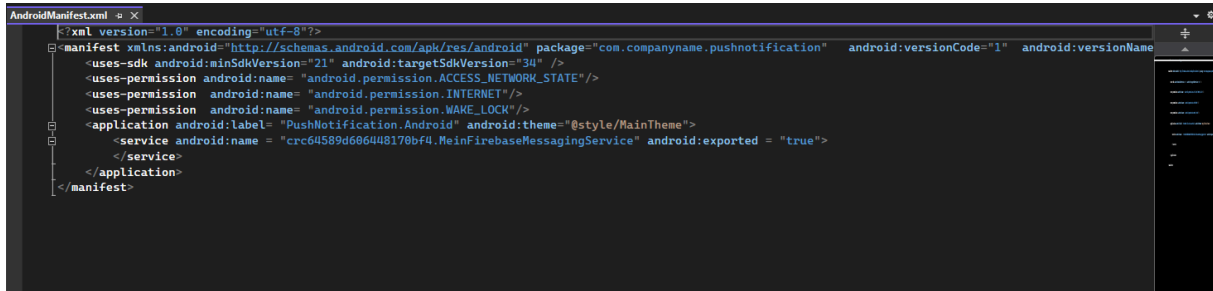
5. In Android Projekt – **Properties** – **AndroidManifest.XML** (mit Text Editor öffnen) die Application ändern:

```
<application android:label= "PushNotification.Android"
android:theme="@style/MainTheme"></application>
```

Service dazu einfügen:

```
<service android:name = "crc64589d606448170bf4.MeinFirebaseMessagingService"
android:exported = "true">
    </service>
```

Soll aussehen:



In Android Projekt neue Anwendung ( **MainApplication.cs** ) anlegen und den Code in die Hauptklasse reinkopieren:

```
public class MainApplication : Application
{
    public MainApplication(IntPtr handle, JniHandleOwnership transer) : base(handle, transer)
    {
    }

    public override void OnCreate()
    {
        base.OnCreate();

        //Set the default notification channel for your app when running Android Oreo
        if (Build.VERSION.SdkInt >= Android.OS.BuildVersionCodes.O)
        {
            //Change for your default notification channel id here

            FirebasePushNotificationManager.DefaultNotificationChannelId =
            "FirebasePushNotificationChannel";

            //Change for your default notification channel name here

            FirebasePushNotificationManager.DefaultNotificationChannelName = "General";
        }

        //If debug you should reset the token each time.
```

```

#if DEBUG

    FirebasePushNotificationManager.Initialize(this, true);
#else

    FirebasePushNotificationManager.Initialize(this, false);

#endif

//Handle notification when app is closed here

CrossFirebasePushNotification.Current.OnNotificationReceived += (s, p) =>
{
    };
}
}

```

Den folgenden Code in **MainActivity**-Klasse in die Hauptklasse reinkopieren:

```

[Activity(Label = "PushNotification", 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
{
    protected override void OnCreate(Bundle savedInstanceState)
    {
        base.OnCreate(savedInstanceState);

        Xamarin.Essentials.Platform.Init(this, savedInstanceState);
        global::Xamarin.Forms.Forms.Init(this, savedInstanceState);
        Plugin.FirebasePushNotification.FirebasePushNotificationManager.ProcessIntent(this, Intent);
        LoadApplication(new App());
    }

    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);
    }
}

```

Fügen in dem Hauptprojekt (C#) in ***App.xaml.cs*** den folgenden Code in der App()-Konstruktion hinzu:

```

public partial class App : Application
{
    public App()
    {
        InitializeComponent();

        DependencyService.Register<MockDataStore>();

        MainPage = new AppShell();

        CrossFirebasePushNotification.Current.OnTokenRefresh += (s, p) =>
        {
            System.Diagnostics.Debug.WriteLine($"TOKEN : {p.Token}");
        };

        CrossFirebasePushNotification.Current.OnNotificationReceived += (s, p) =>
        {
            System.Diagnostics.Debug.WriteLine("Received");

            foreach (var data in p.Data)
            {
                System.Diagnostics.Debug.WriteLine($"{data.Key} : {data.Value}");
            }
        };

        CrossFirebasePushNotification.Current.OnNotificationOpened += (s, p) =>
        {
            System.Diagnostics.Debug.WriteLine("Opened");

            foreach (var data in p.Data)
            {
                System.Diagnostics.Debug.WriteLine($"{data.Key} : {data.Value}");
            }
        }
    }
}

```



```

};

}

protected override void OnStart()

{

}

protected override void OnSleep()

{

}

protected override void OnResume()

{

}

}

```

Mit dem obigen Code erhalten wir ein Token von **p.Token !**

Nachricht senden:

**Firestore-Konsole -> Cloud-Messaging -> Neue Kampagne**

Titel und Text einfügen:

Token kopieren und einfügen:

