Push-Messages with Ionic and OneSignal

Philipp Höhne for Zurich Ionic Meetup, 2018-09-06

Contact: philipp@hoehne.ch
Blog: http://blog.hoehne.ch

What you will learn

- What is One Signal
- Project Setup
- Build a client with Ionic
- Sever-side pushes
- Pro’s & Con’s

What is OneSignal?

- Free service to send push messages to Android, iOS & Windows-Mobile devices and Web Browsers
- Plugin for Ionic available and integration is fairly easy
- Web Dash-Board to run and monitor campaigns
- Powerful Web API for server side integration
- Email: Sendgrid, Mailgun, Mandril [BETA]

Pre-requisites

- OneSignal Account
- Google Account for Firebase Cloud Messaging (for Android)
- Apple Developer Account (for iOS support)
- Ionic 1, 2 or 3
- Cocoa Pods (on Mac)
- Xcode (on Mac, for iOS)
- iOS Device (for testing)

Setup projects / apps

1. Define you widget- (Ionic) / bundle- (iOS) ID
   `e.g. com.myserver.mypushdemo`
2. Setup iOS and/or Android Firebase project
   Hint: for iOS enable “App Groups” and “Push Notifications”
3. Create iOS certificates (manually or with the OneSignal tool)
4. Setup OneSignal project

Setup Ionic App

- Make sure Cocoa pods are installed and up-2-date (MacOS)
  ```
  sudo gem install cocoapods –pre / pod repo update
  ```
- Create new Ionic app
  ```
  ionic start OneSignalDemo blank
  ```
- Add platform(s):
  ```
  ionic cordova platform add ios [/ android / …]
  ```
- Add Cordova push plugin
  ```
  ionic cordova plugin add onesignal-cordova-plugin
  ```
- Add OneSignal module
  ```
  npm install --save @ionic-native/onesignal
  ```
  (!)
- Set widget-id in config.xml

Client side (Ionic) code
`app.module.ts`

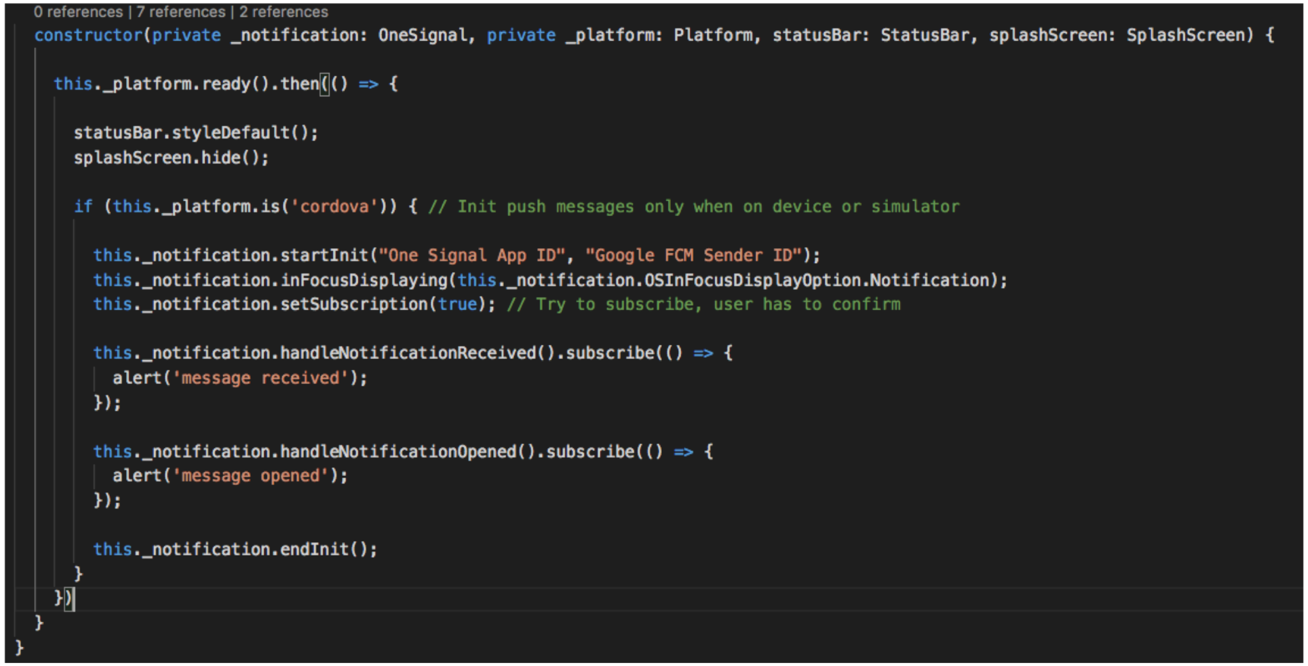
- Import OneSignal

```
import { OneSignal } from '@ionic-native/onesignal’;
```

- Add to providers

```
providers: [
           StatusBar,
           SplashScreen,
           OneSignal,
           {provide: ErrorHandler, useClass:IonicErrorHandler}
]
```

Client side (Ionic) code
app.component.ts

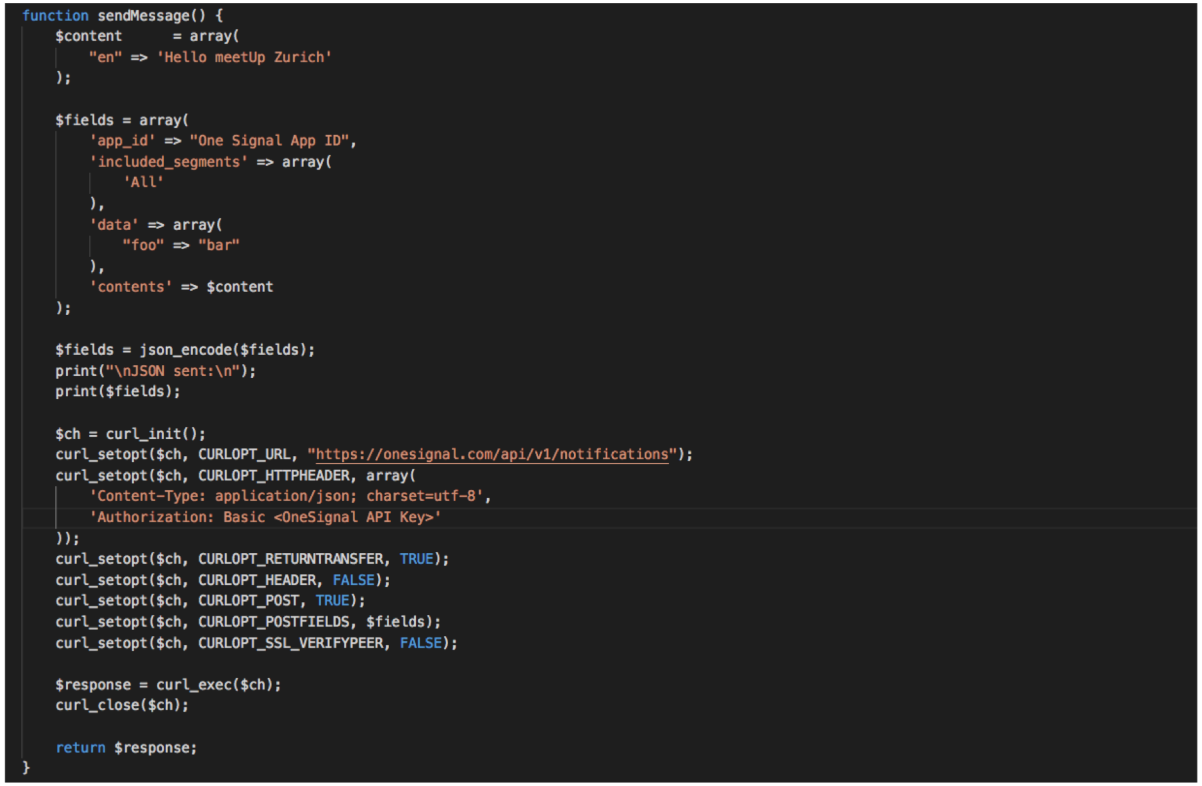


```
  0 references | 7 references | 2 references
  constructor(private _notification: OneSignal, private _platform: Platform, statusBar: StatusBar, splashScreen: SplashScreen) {

    this._platform.ready().then(() => {

      statusBar.styleDefault();
      splashScreen.hide();

      if (this._platform.is('cordova')) { // Init push messages only when on device or simulator

        this._notification.startInit("One Signal App ID", "Google FCM Sender ID");
        this._notification.inFocusDisplaying(this._notification.OSInFocusDisplayOption.Notification);
        this._notification.setSubscription(true); // Try to subscribe, user has to confirm

        this._notification.handleNotificationReceived().subscribe(() => {
          alert('message received');
        });

        this._notification.handleNotificationOpened().subscribe(() => {
          alert('message opened');
        });

        this._notification.endInit();
      }
    })
  }
}
```

Build & Test Client (iOS)

- Build Ionic app
  ```
  ionic cordova build ios
  ```
- Open project in Xcode
  ```
  ../platforms/ios/OneSignalDemo.xcodeproj
  ```
- Set General -> Signing -> Team
- Enable Capabilities -> Push Notification
- Attach iOS Device (iPhone or iPad)
- Build & Run
- Start app on device, then check on OneSignal Console whether your device got registered.
- Good? Send your first message!

Sending messages server side (Example: PHP)



```php
function sendMessage() {
    $content      = array(
        "en" => 'Hello meetUp Zurich'
    );

    $fields = array(
        'app_id' => "One Signal App ID",
        'included_segments' => array(
            'All'
        ),
        'data' => array(
            "foo" => "bar"
        ),
        'contents' => $content
    );

    $fields = json_encode($fields);
    print("\nJSON sent:\n");
    print($fields);

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, "https://onesignal.com/api/v1/notifications");
    curl_setopt($ch, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/json; charset=utf-8',
        'Authorization: Basic <OneSignal API Key>'
    ));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
    curl_setopt($ch, CURLOPT_HEADER, FALSE);
    curl_setopt($ch, CURLOPT_POST, TRUE);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $fields);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);

    $response = curl_exec($ch);
    curl_close($ch);

    return $response;
}
```

Why to use OneSignal?

- It’s fee (no quotas, no limits)
- Supports Android, iOS, Windows Mobile, Web-Push and eMail
- Online Dashboard to create and watch campaigns
- Powerful Web API
- Target group segregation (by device, activity, location etc.)
- A/B Testing
- Plugin for Ionic available
- Little client side code
- No backend required to get started, messages can be sent from web console

Why not to use OneSignal?

- “There is no free lunch.” => OneSignal states that they earn money from selling data to 3rd parties but there are no further details about this. It might be tricky to explain this to your clients in your app’s privacy policy. Paid offerings “when privacy is a concern”.
- Ionic 4 not officially supported yet
- Documentation looks good at the 1st glance but has flaws in the details
- Free or cheap alternatives from big players
  - Google Firebase Cloud Messaging (directly)
  - Amazon SNS (1 mio msg/month free, 1 USD/mio msg after)

Links

- OneSignal: http:/onesignal.com
- Firebase Cloud Messaging: https://firebase.google.com/docs/cloud-messaging
- FCM + Ionic w/o OneSignal: https://angularfirebase.com/lessons/ionic-native-with-firebase-fcm-push-notifications-ios-android/
- Amazon SNS + Ionic: https://angularfirebase.com/lessons/ionic-native-with-firebase-fcm-push-notifications-ios-android/