Set Firebase for PWA on your site

If you are only using parts of the Mobsted platform^ such as App Saving Widgets and want to have your own Service Workers done right. Please read on the Following:

- DO NOT directly follow the Firebase tech docs.
- Use the following solution to avoid issues above:

• DO NOT directly follow the Firebase tech docs.

The problem you will have:

Firebase has the following official documentation:

https://firebase.google.com/docs/cloud-messaging/js/client#access_the_registration_token

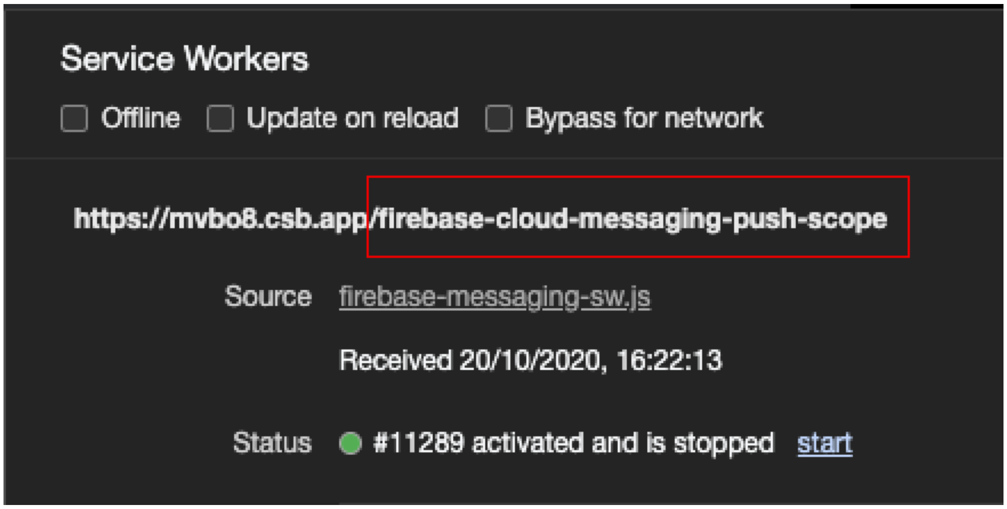
BUT, if a developer follows this docs, the web site will not function as PWA.

We created a live example to see this: https://codesandbox.io/s/pwa-fcm-mvbo8?file=/main.js

Here is the procedure from the doc:

1. Install firebase SDK
2. Create a file with firebase-messaging-sw.js name and place it in the root of your domain.
3. Firebase SDK itself will connect firebase-messaging-sw.js to the site with scope: firebase-cloud-messaging-push-scope

PWA can not function due to the scope (marked in red below) which is auto created by SDK:



Service Workers

□ Offline □ Update on reload □ Bypass for network

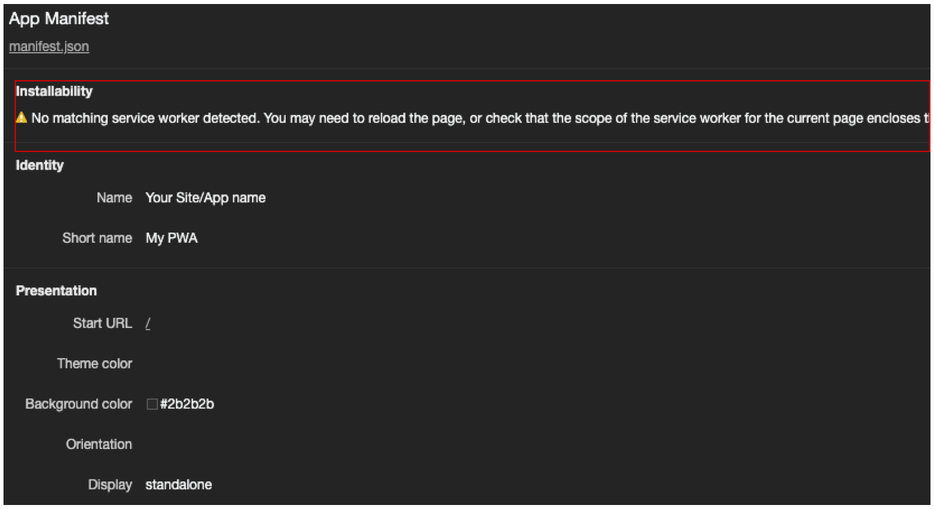
https://mvbo8.csb.app/firebase-cloud-messaging-push-scope

Source firebase-messaging-sw.js

Received 20/10/2020, 16:22:13

Status ● #11289 activated and is stopped start

Placing the SAME scope into manifest will obviously only work, if such page is created on a website, and only on THIS very page. On any other url on the site we get:



App Manifest

manifest.json

Installability

⚠ No matching service worker detected. You may need to reload the page, or check that the scope of the service worker for the current page encloses t

Identity

Name	Your Site/App name
Short name	My PWA

Presentation

Start URL	/
Theme color	
Background color	□#2b2b2b
Orientation	
Display	standalone

- Use the following solution to avoid issues above:

1. Register the SW in site’s manually (so NOT using SDK):

```
if ("serviceWorker" in navigator) { const registration = await navigator.serviceWorker.register( "firebase-messaging-sw.js" ); messaging.
useServiceWorker(registration); }
```

```
if ("serviceWorker" in navigator) {
    const registration = await navigator.serviceWorker.register(
      "firebase-messaging-sw.js"
    );
    messaging.useServiceWorker(registration);
}
```

2) send in this SW into Firebase using this Message method”:

https://firebase.google.com/docs/reference/js/firebase.messaging.Messaging#useserviceworker

This way the Scope of SW stays = to Root, if not set otherwise, or can be managed, as needed.

The issue does not exist, if you go for OneSignal.

Thanks all.