Sending Push Notifications

Push notifications are an important part of any mobile app. They allow you to send notifications to your users even when they are not using your app. This guide will show you how to send push notifications to different mobile app frameworks from your Supabase edge functions.

Expo Push Notifications | Firebase Cloud Messaging

Expo makes implementing push notifications easy. All the hassle with device information and communicating with Firebase Cloud Messaging (FCM) or Apple Push Notification Service (APNs) is done behind the scenes. This allows you to treat Android and iOS notifications in the same way and save time both on the frontend and backend.

Find the example code on GitHub.

Supabase setup

Create a new Supabase project.

Link your project: `supabase link --project-ref your-supabase-project-ref`

Start Supabase locally: `supabase start`

Push up the schema: `supabase db push` (schema is defined in supabase/migrations)

Expo setup

To utilize Expo's push notification service, you must configure your app by installing a set of libraries, implementing functions to handle notifications, and setting up credentials for Android and iOS. Follow the official Expo Push Notifications Setup Guide to get the

credentials for Android and iOS. This project uses Expo's EAS build service to simplify this part.

1. Install the dependencies: `npm i`
2. Create a new Expo project
3. Link this app to your project: `npm install --global eas-cli && eas init --id your-expo-project-id`
4. Create a build for your physical device
5. Start the development server for your project: `npx expo start --dev-client`
6. Scan the QR code shown in the terminal with your physical device.
7. Sign up/in to create a user in Supabase Auth.

Enhanced security for push notifications

1. Navigate to your Expo Access Token Settings.
2. Create a new token for usage in Supabase Edge Functions.
3. Toggle on "Enhanced Security for Push Notifications".
4. Create the local `.env` file: `cp .env.local.example .env.local`
5. In the newly created `.env.local` file, set your `EXPO_ACCESS_TOKEN` value.

Deploy the Supabase Edge Function

The database webhook handler to send push notifications is located in supabase/functions/push/index.ts. Deploy the function to your linked project and set the `EXPO_ACCESS_TOKEN` secret.

1. `supabase functions deploy push`
2. `supabase secrets set --env-file .env.local`

```
import { createClient } from 'npm:@supabase/supabase-js@2'

console.log('Hello from Functions!')

```

```
interface Notification {
  id: string
  user_id: string
  body: string
}

interface WebhookPayload {
  type: 'INSERT' | 'UPDATE' | 'DELETE'
  table: string
  record: Notification
  schema: 'public'
  old_record: null | Notification
}

const supabase = createClient(
  Deno.env.get('SUPABASE_URL')!,
  Deno.env.get('SUPABASE_SERVICE_ROLE_KEY')!
)

Deno.serve(async (req) => {
  const payload: WebhookPayload = await req.json()
  const { data } = await supabase
    .from('profiles')
    .select('expo_push_token')
    .eq('id', payload.record.user_id)
    .single()

  const res = await fetch('https://exp.host/--/api/v2/push/send', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
      Authorization: `Bearer ${Deno.env.get('EXPO_ACCESS_TOKEN')}`,
    },
```

Edge Functions

```
    body: JSON.stringify({
      to: data?.expo_push_token,
      sound: 'default',
      body: payload.record.body,
    }),
  }).then((res) => res.json())

  return new Response(JSON.stringify(res), {
    headers: { 'Content-Type': 'application/json' },
  })
})
```

Create the database webhook

Navigate to the Database Webhooks settings in your Supabase Dashboard.

1. Enable and create a new hook.
2. Conditions to fire webhook: Select the `notifications` table and tick the `Insert` event.
3. Webhook configuration: Supabase Edge Functions.
4. Edge Function: Select the `push` edge function and leave the method as `POST` and timeout as `1000`.
5. HTTP Headers: Click "Add new header" > "Add auth header with service key" and leave Content-type: `application/json`.
6. Click "Create webhook".

Send push notification

1. Navigate to the table editor in your Supabase Dashboard.
2. In your `notifications` table, insert a new row.
3. Watch the magic happen 🪄

Edit this page on GitHub