



Build, deploy and scale Angular apps
quickly on DigitalOcean

[Get started for free](#)

 DigitalOcean

 NAPPD

[Follow @EnappdStore](#)



How to implement Firebase Push notifications in Ionic 5



Firebase Push Notifications in Ionic 5

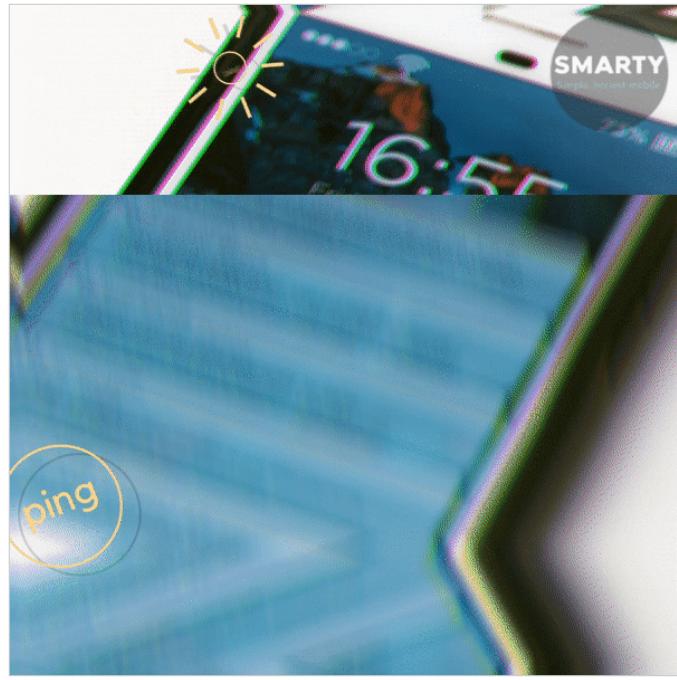
In this post, we'll learn how to setup Push notification in Ionic 5 app using Firebase. We'll send notification using Firebase console and receive them in the app. We will also learn how to handle push notifications in your Ionic 5 apps.



No no... not the push we are talking about 😊

What are Push Notifications

Ok, I'll try to define it in my own language. How do you know when you get a new message on Whatsapp, Facebook or Twitter, if the app is closed ? Yes, those small messages that pop down from the top of the screen, showing new messages, updates or news are called Push notifications (or simply Push)



Yes, those are push notifications

One might wonder why to integrate Push notifications in an app. Let me tell you, push notifications is the **single biggest feature** to keep your users engaged with your app. People are obsessed with notifications. People keep checking their phones regularly so they don't miss any notification. If your app sends regular (and interesting) push, your users will be happy, and you'll see much less app uninstalls than you'll see without push.

What is Ionic 5?

You probably already know about Ionic, but I'm putting it here just for the sake of beginners. **Ionic** is a complete open-source SDK for hybrid mobile app development created by Max Lynch, Ben Sperry and Adam Bradley of Drifty Co. in 2013. Ionic provides tools and services for developing hybrid mobile apps using Web technologies like CSS, HTML5, and Sass. Apps can be built with these Web technologies and then distributed through native app stores to be installed on devices by leveraging Cordova.

So, in other words—if you create Native apps in Android, you code in Java. If you create Native apps in iOS, you code in Obj-C or Swift. Both of these are powerful but complex languages. **With Cordova (and Ionic) you can write a single piece of code for your app that can run on both iOS and Android** (and windows!), that too with the simplicity of HTML, CSS, and JS.

Structure

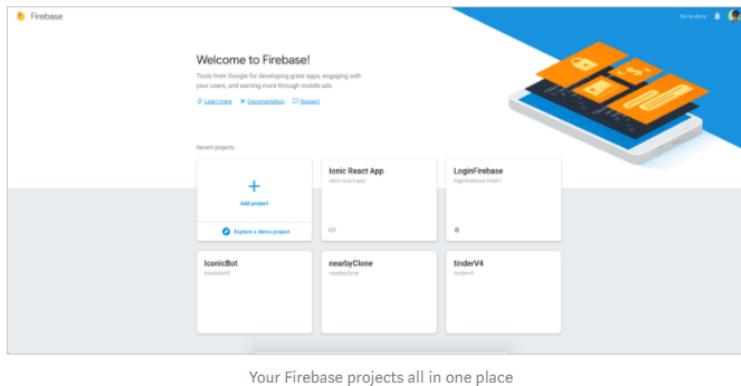
I will go ahead in step-by-step fashion so you can follow easily.

1. Create a Firebase project and find Push options
2. Create a basic Ionic 5 app
3. Connect your Ionic 5 app with Firebase and install Push plugin

4. Build the app on Android
5. Send notifications from Firebase console
6. Receive different type of notifications in app
7. Handle Push notifications in your app
8. Custom Push notification using node server

1. Create a Firebase project and find Push options

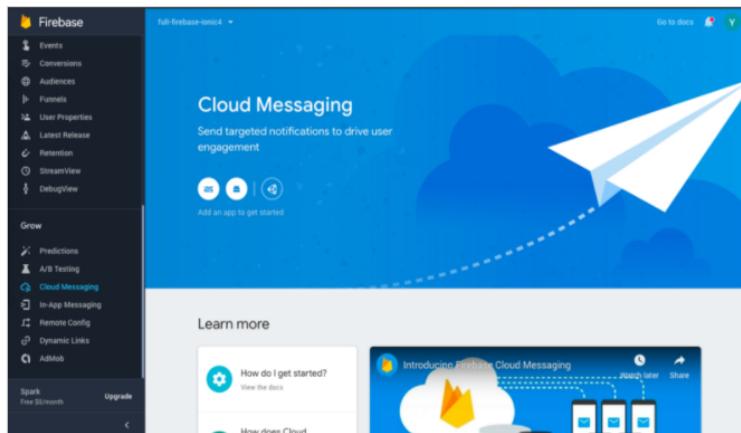
Go to [Firebase](#) and create your first project (or use an existing one). Your console should look like this



Your Firebase projects all in one place

Note—It's really easy to create a Firebase project, but if you still face any issue, follow step 1–4 of [this blog](#)

Click on your project and you'll enter the project's dashboard. Look for **Cloud Messaging** tab. This is where the magic will happen !

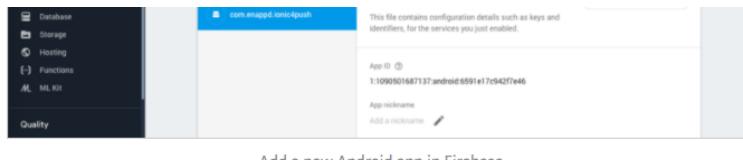


Cloud messaging section of Firebase console—You can send push notifications from here

Push Notifications Settings

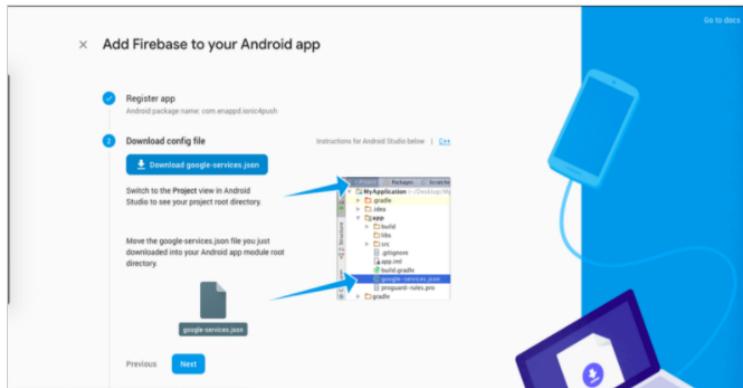
Firebase console also contains push notifications settings for web, android and iOS. Here you can find your **sender_id**, upload iOS push certificates etc. For setting up options, you'll first have to create an Android app in Firebase console.





Add a new Android app in Firebase

During the process, it will ask you to enter app's **package name** and provide **google-services.json**. Make sure you keep the package name same as what you are setting in `config.xml` of your app. (Ionic 5 default package name is `io.ionic.starter`, don't use the default package name)



Download google-services.json

2. Create a basic Ionic 5 app

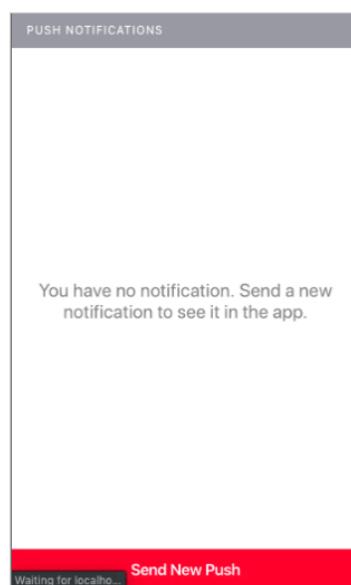
Creating a basic Ionic 5 app is very easy. Assuming you have all basic requirements installed in your system, run

```
$ ionic start MyApp blank
```

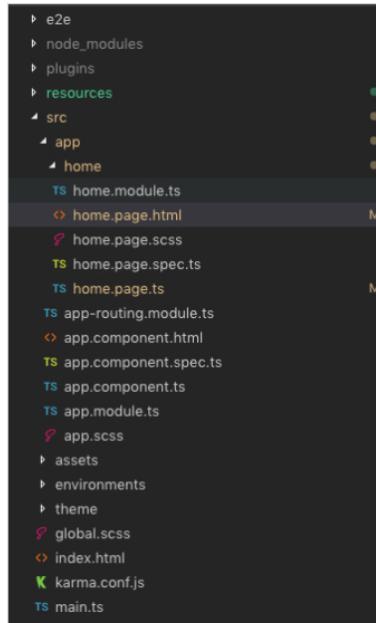
This creates your app with title `MyApp` and blank template.

For more details on how to create a basic Ionic 5 app, refer to my blog [How to create an Ionic 5 app](#)

With slight modifications, your homepage will look like this.



The file structure looks something like this, just for an idea



File structure for Ionic 5 Push sample app

3. Install Push plugin in your app

We will implement Push using Ionic Native FCM plugin. To install this plugin, run

```
$ ionic cordova plugin add cordova-plugin-fcm-with-dependency-updated
$ npm install @ionic-native/fcm
```

This will install the plugin. Import the plugin in `app.module.ts`

```

6  import { SplashScreen } from '@ionic-native/splash-screen/ngx';
7  import { StatusBar } from '@ionic-native/status-bar/ngx';
8
9  import { AppComponent } from './app.component';
10 import { AppRoutingModule } from './app-routing.module';
11 | import { FCM } from '@ionic-native/fcm/ngx';
12
13 @NgModule({
14   declarations: [AppComponent],
15   entryComponents: [],
16   imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule],
17   providers: [
18     StatusBar,
19     SplashScreen,
20   |   FCM,
21   |   { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
22   ],
23   bootstrap: [AppComponent]
24 })
25 export class AppModule { }
26

```

Import push plugin in `app.moudle.ts`

Also, import the plugin in your `home.page.ts` where we'll implement certain actions. `home.page.ts` code will look like following

```

1  import { Component } from '@angular/core';
2  import { FCM } from '@ionic-native/fcm/ngx';
3  import { Platform } from '@ionic/angular';
4
5  @Component({
6    selector: 'app-home',

```

```

7     templateUrl: 'home.page.html',
8     styleUrls: ['home.page.scss'],
9   })
10    export class HomePage {
11      pushes: any = [];
12      constructor(private fcm: FCM, public plt: Platform) {
13        this.plt.ready()
14          .then(() => {
15            this.fcm.onNotification().subscribe(data => {
16              if (data.wasTapped) {
17                console.log("Received in background");
18              } else {
19                console.log("Received in foreground");
20              }
21            });
22
23            this.fcm.onTokenRefresh().subscribe(token => {
24              // Register your new token in your back-end if you want
25              // backend.registerToken(token);
26            });
27          })
28      }
29      subscribeToTopic() {
30        this.fcm.subscribeToTopic('enappd');
31      }
32      getToken() {
33        this.fcm.getToken().then(token => {
34          // Register your new token in your back-end if you want
35          // backend.registerToken(token);
36        });
37      }
38      unsubscribeFromTopic() {
39        this.fcm.unsubscribeFromTopic('enappd');
40      }
41    }

```

home.page.ts hosted with ❤ by GitHub

[view raw](#)

Ionic 5 Firebase push—Setup push notifications

Note that we are calling the push plugin methods after platform is ready. This is a good practice when you are initializing the push plugin right after the app starts.

Let's walkthrough the code

- **subscribeToTopic**—This method essentially subscribes the user to a specific notification channel. In latest mobile OS, there can be multiple notifications channels. E.g. You can turn off promotional notifications in Uber, but you still want to get the ride related notifications. So you keep the ride related notification subscribed
- **unsubscribeFromTopic**—Pretty much self explanatory. You can unsubscribe from individual notification channels
- **getToken**—Essentially gets the FCM token for the app from Firebase, you can then register this token in your DB for further use. Tokens are used to identify user, and then target notifications to specific users
- **onTokenRefresh**—Essentially does similar to getToken() but a step further. It gets the latest token in case of a token is refreshed by the app. Tokens can be refreshed in following cases

1. App deletes Instance ID
2. App is restored on a new device
3. User uninstalls/reinstall the app
4. User clears app data

`onTokenRefresh` has been deprecated in favor of overriding `onNewToken` in `FirebaseMessagingService`. This method will be invoked on token changes even if `onNewToken` is also used.

IMPORTANT—Place the `google-services.json` file you downloaded from Firebase in root of your project. This is extremely important, as this is how the app and Firebase recognize each other. Without this, the app will not build on device.

More plugin details can be found in its [Github repository](#)

4. Build the app on Android

If you have carried out the above steps correctly, Android build should be a breeze.

Run following command to create Android platform

```
$ ionic cordova platform add android
```

Once platform is added, run the app on device (Make sure you have a device attached to the system. Push notifications might not work on simulator)

```
$ ionic cordova run android
```

Once your app is up and running on the device, we'll send notifications from Firebase console.

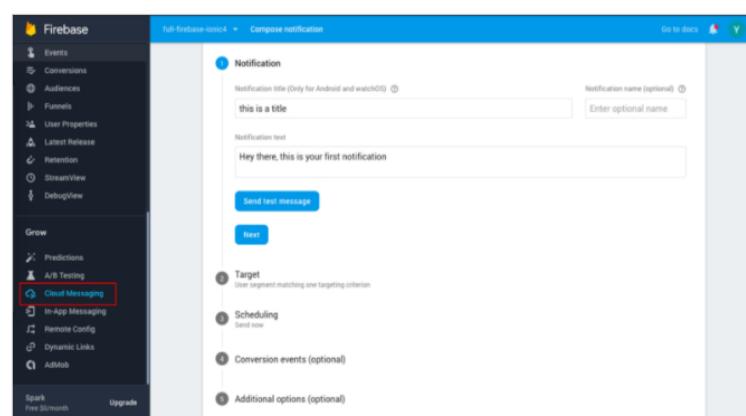
5. Send notifications from Firebase console

Now that the app is ready to receive Push notification, let's see how to send them.

([A detailed guide for Sending Notifications from Firebase Console](#))

Head to your Firebase console → Cloud messaging section → Get started

You'll see a form with which you can send a push notification

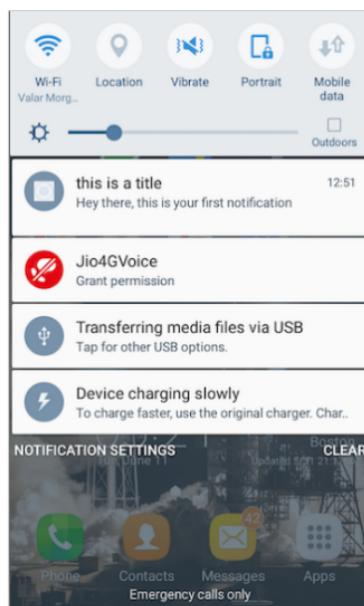


Firebase cloud messaging dashboard

I won't go into details of all the options available in the dashboard. Let's setup the basics

- Enter a Notification title
- Enter a Notification text

Leave all the options as is, for now. Now hit **Review → Publish** the notification. Make sure your app is closed at this point in the device. Your device will receive a notification within few seconds



Push received in the app

Notice there is no icon for the push because we haven't defined any icon yet. The notification is showing the default icon of ionic apps.

Congratulations ! You just sent your first Push in Ionic 5 🎉 🎉

6. Receive different types of Push in Ionic apps

With FCM, you can send two types of messages to clients:

- **Notification messages**, sometimes thought of as “display messages.” These are handled by the FCM SDK automatically. This is what we just sent.
- **Data messages**, which are handled by the client app. You can send custom data with these messages, but these require custom back-end (or Firebase cloud functions) to be sent

Notification messages contain a predefined set of user-visible keys. Data messages, by contrast, contain only your user-defined custom key-value pairs. Notification messages can contain an optional data payload. Maximum payload for both message types is 4KB, except when sending messages from the Firebase console, which enforces a 1024 character limit.

Sending and Receiving a Notification message

This is what we just did in above section—the first notification we sent. Let's see how it looks in the browser console.

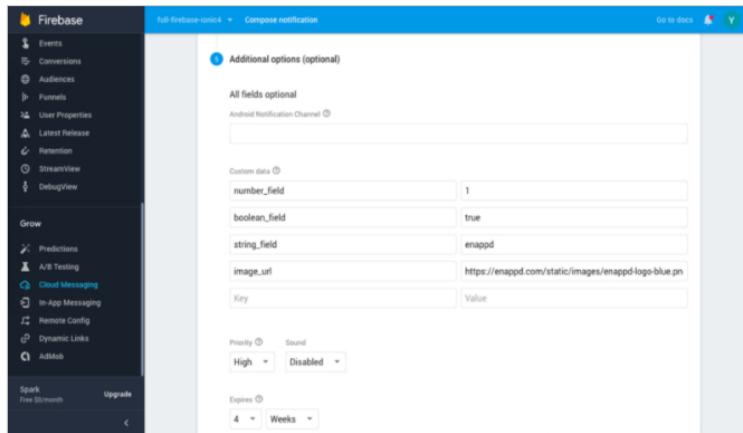
```
⚠ The key "viewport-fit" is not recognized and ignored.  
FCMPlugin.js: is created  
FCMPlugin Ready OK  
Angular is running in the development mode. Call enableProdMode() to enable the production mode.  
Ionic Native: deviceready event fired after 2988 ms  
▼ Object ⓘ  
  body: "Hey there, this is your first notification"  
  title: "this is a title"  
  wasTapped: false  
  ► __proto__: Object  
Received in foreground
```

Data sent along with Notification message

Note, you don't have any custom data fields here that you can handle

Sending and Receiving a Data message

To send a data message, head over to **Additional Options** section in Firebase cloud messaging form



Add additional field to be sent as Data message

When the notification is received, the data appears as follows

```
▼ Object ⓘ  
  body: "Hey there, this is your first notification"  
  boolean_field: "true"  
  image_url: "https://enappd.com/static/images/enappd-logo-blue.png"  
  number_field: "1"  
  string_field: "enappd"  
  title: "this is a title"  
  wasTapped: false  
  ► __proto__: Object  
Received in foreground
```

Notice the new data fields in the message

These new data fields can be used by the app for any purpose. Currently all these fields come as **String** field only, so be careful

7. Handle Push notifications in your app

Push notification can be received in three states in your Ionic app

- When app is in foreground (open and visible)
- When app is in background (open but minimized)
- When app is closed (killed)

As you know from daily experience, notifications are required to work in all 3 cases. So we handle them a little differently for each case

7a. When app is in foreground

The plugin we are using, allows foreground notifications to be received silently by default. That means, you do get the data sent by the push, but there is no notification shown in the app.

If you want to show a notification in such case, you can show an alert or customized modal message to user, like what Tinder shows when you get a match.



It's a match and it's a notification !!

7b. When app is in background

When the app is in background, your phone shows a standard notification popping from the top (for most devices !). When you tap the notification, you won't receive any data 😞

Turns out Firebase does not support all the options of Data notifications currently. To receive a callback in the app for this type of notification, you need a key-value pair `click-action: FCM_PLUGIN_ACTIVITY` . But this is not possible view Firebase console.

Ideally, a notification comprises of a lot of data. Following JSON shows most of it. If you are using a custom back-end, you can send all of this data in the notification, and your app will then receive the callback with data.

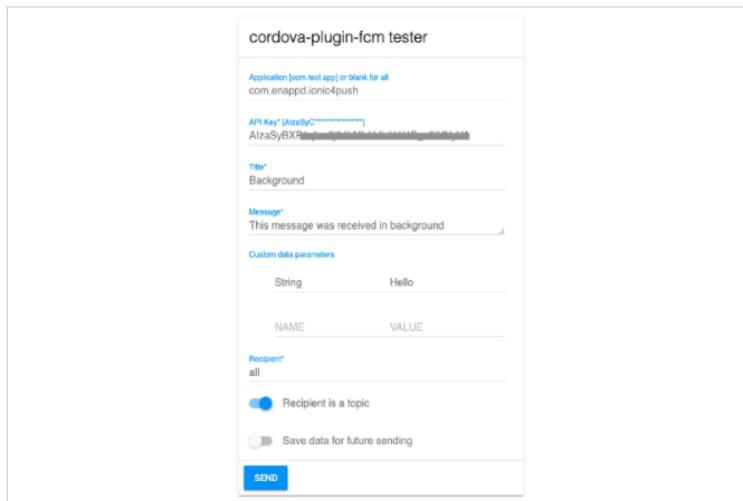
```
1 //POST: https://fcm.googleapis.com/fcm/send
2 //HEADER: Content-Type: application/json
3 //HEADER: Authorization: key=AIzaSy*****
4 {
5   "notification":{
6     "title":"Notification title",
7     "body":"Notification body",
8     "sound":"default",
9     "click_action":"FCM_PLUGIN_ACTIVITY",
10    "icon":"fcm_push_icon"
11  },
12  "data":{
13    "param1":"value1",
14    "param2":"value2"
15  },
16  "to":"/topics/topicExample",
17  "priority":"high",
18  "restricted_package_name": ""
19 }
20 //sound: optional field if you want sound with the notification
21 //click_action: must be present with the specified value for Android
22 //icon: white icon resource name for Android >5.0
23 //data: put any "param":"value" and retrieve them in the JavaScript notification callback
24 //to: device token or /topic/topicExample
25 //priority: must be set to "high" for delivering notifications on closed iOS apps
26 //restricted_package_name: optional field if you want to send only to a restricted app package
```



Ionic 5 Firebase push—Push notification syntax

You can find more details on push notification content syntax in [Firebase's official documentation](#).

Don't worry if you don't have custom back-end for now. You can use <https://cordova-plugin-fcm.appspot.com/> to send notifications for testing.



Send test push from cordova-plugin-fcm.appspot.com

Get the API key from Firebase console → Project Settings → Cloud Messaging → Legacy server Key

Now when you receive the notification in background, and tap the notification, you receive the data inside the app like following



Data receive in Background push

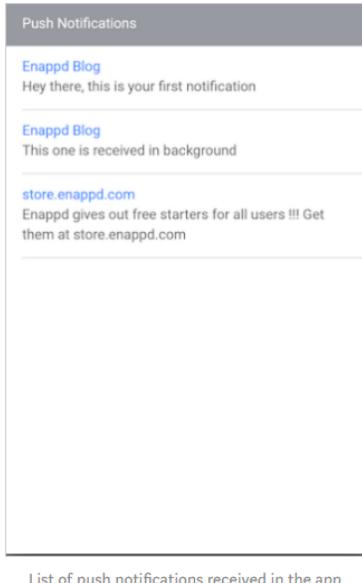
7c. When app is closed

When the app is closed, your phone shows a standard notification popping from the top (for most devices !). If you have sent the notification from Firebase, again, the data won't show up in the callback.

But if you are sending the notification from cordova-plugin-fcm.appspot.com, or with your custom back-end (as we'll do later in the blog), you receive the data in the same format as above. Now you can use the data to perform any action inside the app.

You are probably wondering how does our push notification list look like now. Since we are not saving anything in local storage, it will only record current session's push notification. The list looks like this after few

notifications.



List of push notifications received in the app

8. Custom notification using NodeJS server

In this tutorial we have used the NodeJS as the server but you can use any of the server scripting language for serving the push notification using FCM.

So you are wondering how we can send the notification to a specified person not the others or to all. So for this the FCM token comes into the picture so we will use that unique token to identify the user to whom we want to send the push notification.

In first step we will use the below piece of code to generate the FCM token using [@ionic-native/fcm/ngx](#)

```
1 getToken() {
2   this.fcm.getToken().then(token => {
3     // Register your new token in your back-end if you want
4     // backend.registerToken(token);
5   });
6 }
```

(partial) home.page.ts hosted with ❤ by GitHub

[view raw](#)

Ionic 5 Firebase push—Get push token

So you can save this **token** variable in your database in `users` model and later use this for sending push notification to that specific user.

As explained earlier, once the token is registered, next time onwards it is always good practice to use `onTokenRefresh` to get the latest token in the app. You can compare this to your previously saved token in DB, and update if required

So you can use any of the database with which you are convenient or you can use cloud Firestore to save the token in users model.(Which is not in scope for the tutorial or you can check [this](#) to get started with firebase)

Push notification code for node server

Node is a great scripting language where you can create REST APIs and

can do many more useful stuff.

```
1 const message = {
2   registration_ids: [...req.body.userFcmToken], // array required
3   notification: {
4     title: req.body.notificationTitle,
5     body: req.body.notificationBody,
6     sound: "default",
7     icon: "ic_launcher",
8     badge: req.body.notificationBadge ? req.body.notificationBadge : "1",
9     click_action: 'FCM_PLUGIN_ACTIVITY',
10   },
11   priority: req.body.notificationPriority ? req.body.notificationPriority : 'high',
12   data: {
13     action: req.body.actionType, // Action Type
14     payload: req.body.payload // payload
15   }
16 };
```

pushAPI.js hosted with ❤ by GitHub

[view raw](#)

Ionic 5 Firebase push—Compose push notification on nodeJS server

So this will be the message object which will be later passed to the FCM API, In this **registration_ids** is the array of FCM tokens which we have generated in previous section so we will pass all the tokens in an array to whom we want to send the push notifications.

The **notification** object will contain the other general information like title, body, sound and other stuff.

And if we want to send the notification to a user with some data we can pass that in the data object we have—

1. **action**—Basically it can be used to redirect to screen when we click on a notification e.g if we send '**/orders**' in the action and after that when we receive the notification we can simply do this **this.route.navigate([data.action])**.
2. **payload**—It can be the any other useful data which we want to send to the user like it can be a object containing message from other user.

Now finally we will send the data into the FCM API to further send the notification to the users phones

```
1 ....
2 const fcm = new FCM(serverKey);
3 return new Promise((resolve, reject) => {
4   fcm.send(message, (err, response) => {
5     if (err) {
6       console.log("Something has gone wrong!", JSON.stringify(err));
7       reject(err);
8     } else {
9       console.log("Successfully sent with response: ", response);
10      resolve(response)
11    }
12  })
13})
```

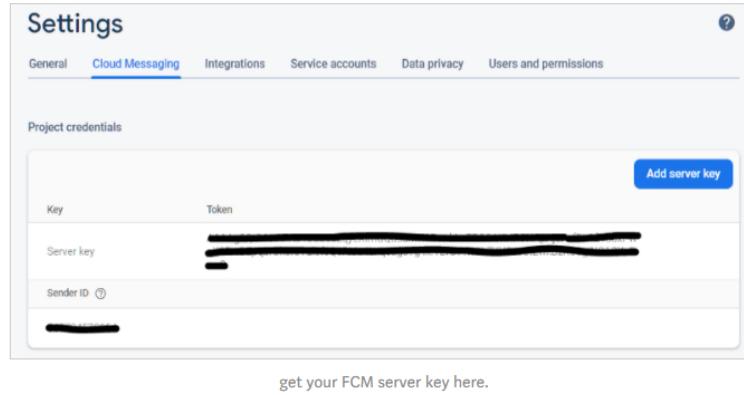
pushAPI.js hosted with ❤ by GitHub

[view raw](#)

Ionic 5 Firebase push—Send push from nodeJS server

Here we will use the server key which you can get from firebase console and then go to **Project overview settings** and click on **Cloud Messaging** and get the key on right now copy it.

and get the key on right name **server key**.



The screenshot shows the 'Cloud Messaging' tab selected in the Firebase console's navigation bar. Under the 'Project credentials' section, there is a table with two columns: 'Key' and 'Token'. A single row contains 'Server key' in the 'Key' column and a long blacked-out string in the 'Token' column. Below the table is a 'Sender ID' field containing a blacked-out string. At the top right of the table area is a blue 'Add server key' button.

get your FCM server key here.

And pass the above **message** object in this API to send the message to a particular user.

Now you can bind your complete code in a REST API which can be called from the client side with the message object data.

```
1 const express = require("express");
2 const app = express();
3 const FCM = require('fcm-node');
4 const serverKey = '*****' //put your server key here
5 app.post('/send-push',(req,res) =>
6   const message = {
7     registration_ids: [...req.body.userFcmToken] , // array required
8     notification: {
9       title:req.body.notificationTitle ,
10      body: req.body.notificationBody,
11      sound: "default",
12      icon: "ic_launcher",
13      badge: req.body.notificationBadge ? req.body.notificationBadge : "1",
14      click_action: 'FCM_PLUGIN_ACTIVITY',
15    },
16    priority: req.body.notificationPriority ? req.body.notificationPriority : 'high',
17    data: {
18      action:req.body.actionType, // Action Type
19      payload:req.body.payload // payload
20    }
21
22   const fcm = new FCM(serverKey);
23
24   fcm.send(message, (err, response) => {
25     if (err) {
26       console.log("Something has gone wrong!", JSON.stringify(err));
27       res.send(err);
28     } else {
29       console.log("Successfully sent with response: ", response);
30       res.send(response)
31     }
32   })
33 });

pushAPI.js hosted with ❤ by GitHub
```

[view raw](#)

Ionic 5 Firebase push—nodeJS (Express) code for push server

Now we can call this `/send-push` from our client side application wherever we need to send the notification to the other user.

Handling Received Notifications

so we have learnt how to send the notification using both firebase console and the custom back end APIs. Now we see how we will handle the received notification.

```

1  this.fcm.onNotification().subscribe(data => {
2      //Here we can get the data using data.actionType and data.payload and use it accordingly
3      if (data.wasTapped) {
4          console.log("Received in background");
5      } else {
6          console.log("Received in foreground");
7      };
8  });

```

(partial) home.page.ts hosted with ❤ by GitHub

[view raw](#)

Ionic 5 Firebase push—Handle push notification data in app

Here we can get the data we have passed in the message object and use it according to the user's need.

Conclusion

In this post we learnt how to setup Firebase push notifications in an Ionic 5 app, how to send notifications using Firebase console and a nodeJS server, how to receive different types of notifications, how to handle them in the app and send a message to particular user. We also created a small Ionic 5 app with provisions to show the received push notifications as a list.

Stay tuned for more Ionic blogs !

... . . .

Next Steps

If you liked this blog, you will also find the following blogs interesting and helpful. Feel free to ask any questions in the comment section

Ionic Capacitor

- Basic—[Geolocation](#) | [Barcode & QR code](#) | [Facebook Login](#) (Angular) | [Facebook Login \(React\)](#) | [Icon and Splash](#) | [Camera & Photo Gallery](#) | [Debugging with browser](#) | Theming in Ionic apps
- Advanced—[AdMob](#) | [Local Notifications](#) | [Google Login](#) | [Twitter Login](#) | [Games using Phaser](#) | [Play music](#) | [Push Notifications](#)

Ionic Cordova

- [Taxi Booking App example with Ionic, Node, Express and MySQL](#)
- Ionic Payment Gateways—[Stripe with Firebase](#) | [Stripe with NodeJS](#) | [PayPal](#) | [Apple Pay](#) | [RazorPay](#)
- Ionic Charts with—[Google Charts](#) | [HighCharts](#) | [d3.js](#) | [Chart.js](#)
- Ionic Authentications—[Via Email](#) | [Anonymous](#) | [Facebook](#) | [Google](#) | [Twitter](#) | [Via Phone](#)
- Ionic Features—[Geolocation](#) | [QR Code reader](#) | [Pedometer](#) | [Signature Pad](#) | [Background Geolocation](#)
- Media in Ionic—[Audio](#) | [Video](#) | [Image Picker](#) | [Image Cropper](#)

... . . .

- Ionic Essentials—[Debugging with browser](#) | [Native Storage](#) | [Translations](#) | [RTL](#) | [Sentry Error Monitoring](#) | [Social sharing](#)
 - Ionic messaging—[Firebase Push](#) | [Reading SMS](#) | [Local Notifications](#)
 - Ionic with Firebase—[Basics](#) | [Hosting and DB](#) | [Cloud functions](#) | [Deploy App to Firebase](#) | [Firebase simulator](#)
 - Unit Testing in Ionic—[Part 1](#) | [Mocks & Spies](#) | [Async Testing](#)
- . . .

Ionic React Full App with Capacitor

If you need a base to start your next Ionic 5 React Capacitor app, you can make your next awesome app using [Ionic 5 React Full App in Capacitor](#)



Ionic 5 React Full App in Capacitor from Enappd

Ionic Capacitor Full App (Angular)

If you need a base to start your next Angular Capacitor app, you can make your next awesome app using [Capacitor Full App](#)

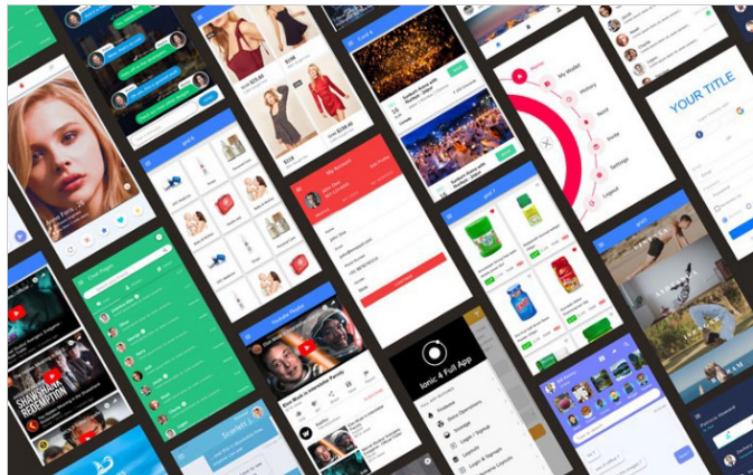




Capacitor Full App with huge number of layouts and features

Ionic Full App (Angular and Cordova)

If you need a base to start your next Ionic 5 app, you can make your next awesome app using [Ionic 5 Full App](#)



Ionic Full App in Cordova, with huge number of layouts and features



4 Comments Enappd [Disqus' Privacy Policy](#) [Login](#) ▾

Heart Recommend 3 [Tweet](#) [Share](#) Sort by Best ▾

 Join the discussion...

LOG IN WITH OR SIGN UP WITH DISQUS [?](#)

 Name

 **Ramya Sai** • 18 days ago
Truepush is a globally free push notification tool to send unlimited push notifications for unlimited subscribers. The web push notification service is available for WordPress, API, and Shopify. It provides features like segmentation, triggers, RSS feed, simple notifications, campaign analysis, and more. You can also join the Truepush monetization plan to add additional revenue to websites using push ads.
^ | v • Reply • Share >

 **Cris** • 7 months ago
Does it work on iOS? Does it work on PWA in iOS?
^ | v • Reply • Share >

 **Miqueas Gutierrez** • 8 months ago
Hola, tambien el servicio de Notificaciones Push INDIGITALL, es Gratis para 30000 Dispositivos, permite enviar Notificaciones Push con imágenes también geolocalizadas, programadas entre muchas más funciones, aquí está el enlace de la documentación para instalarlo fácilmente:<https://docs.indigitall.com...>
^ | v • Reply • Share >

 **anRoswell Harris Maiden** • a year ago
And with capacitor????
^ | v • Reply • Share >

[Subscribe](#) [Add Disqus to your site](#) [Do Not Sell My Data](#) 

Sponsored



Three phases of Indian renaissance

Frontline



The cost of hearing aids in Chandigarh might surprise you

Hear.com



This Is What A New Hearing Aid Should Actually Cost

Best Hearing Aids



Invest in Amazon CFD from 200\$. You could earn, up to \$2000!

Vici



Egypt throws a spatula in Indian Basmati rice exports

BusinessLine



How To Get H1 Visa Job in USA from India

Job In Usa



ENAPPD

Ionic, React native, Vue, Flutter and Firebase App starters, themes and templates

CONNECT WITH US



[About Enappd](#) | [Terms](#) | [Privacy Policy](#)
All rights reserved © Enappd 2018-2019

Browse Templates and Starters

[Ionic 5 Starters](#)

[React Native Starters](#)

[Firebase Starters](#)

[Free Starters](#)

[Full App Starters](#)

BestSellers

[Ionic 5 Full App - Angular Cordova](#)

[Ionic 5 Full App - Angular Capacitor](#)

[Ionic 5 Full App - React Capacitor](#)

[Ionic 4 Taxi Booking Complete Platform](#)

[Ionic 4 Grocery Shopping Complete Platform](#)

[Ionic 4 Spotify Clone](#)

Other Links

[Privacy Policy](#)

[Terms and Conditions](#)

[Licensing Options](#)