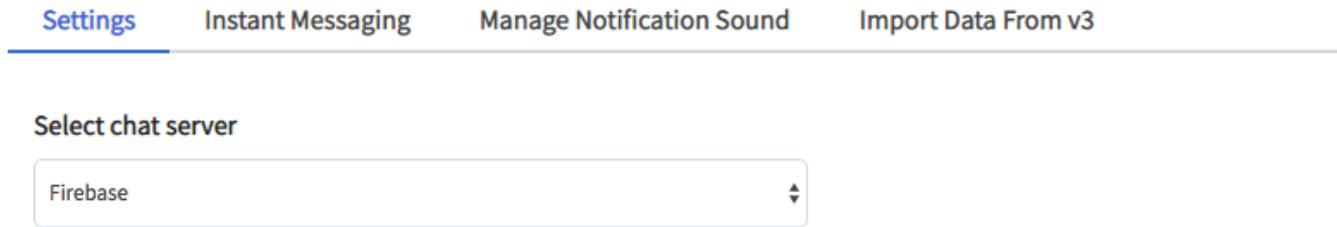


Server Setup for IM with Google Firebase

How to enable the Firebase server on IM App:

Go to AdminCP > Apps > Installed > Instant Messaging > Settings and select "Firebase" at setting **Select chat server**



Then, configure the required **Firebase Config Object** setting. We will guide you to get the value when adding Firebase project which will be instructed below.

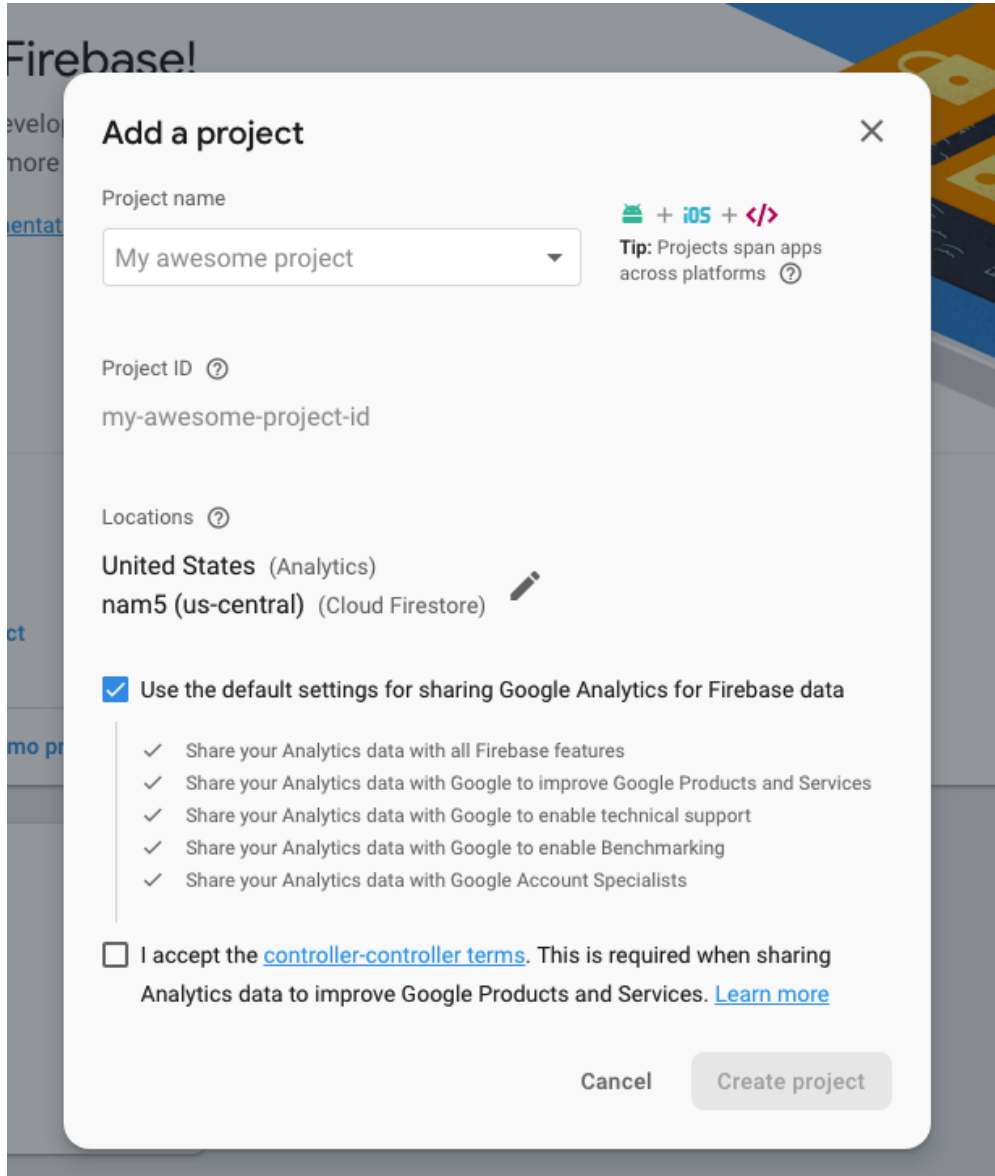
Also, you can configure **Algolia App ID** (Optional) and **Algolia API Key** (Optional) to support search messages. The instructions will be given later.

Setup Google Firebase App:

1. Add Firebase project

Go to [Firebase console](#)

Select **Add Project** if you don't have one



Go to App **Dashboard**, select **Project Overview > Project settings**

The screenshot shows the 'Settings' page for a project named 'IM Chat'. The page has a blue header with the project name and a 'Go to docs' link. Below the header is a navigation menu with tabs for 'General', 'Cloud Messaging', 'Integrations', 'Service accounts', 'Data privacy', and 'Users and permissions'. The 'General' tab is selected. The main content area is divided into two sections: 'Your project' and 'Your apps'. The 'Your project' section contains a table with project details: Project name (IM Chat), Project ID (im-chat-b8c34), Cloud Firestore location (nam5 (us-central)), and Web API Key (redacted). Below this is the 'Public settings' section, which includes a note about public-facing instances, a 'Public-facing name' field (project-[redacted]-1), and a 'Support email' dropdown menu (Not configured). The 'Your apps' section displays the message 'There are no apps in your project' and a prompt to 'Select a platform to get started'. Four platform icons are shown: iOS, Android, a code editor icon (highlighted with a red box), and another icon.

IM Chat Go to docs

Settings

General Cloud Messaging Integrations Service accounts Data privacy Users and permissions

Your project

Project name	IM Chat
Project ID	im-chat-b8c34
Cloud Firestore location	nam5 (us-central)
Web API Key	[REDACTED]

Public settings

These settings control instances of your project shown to the public

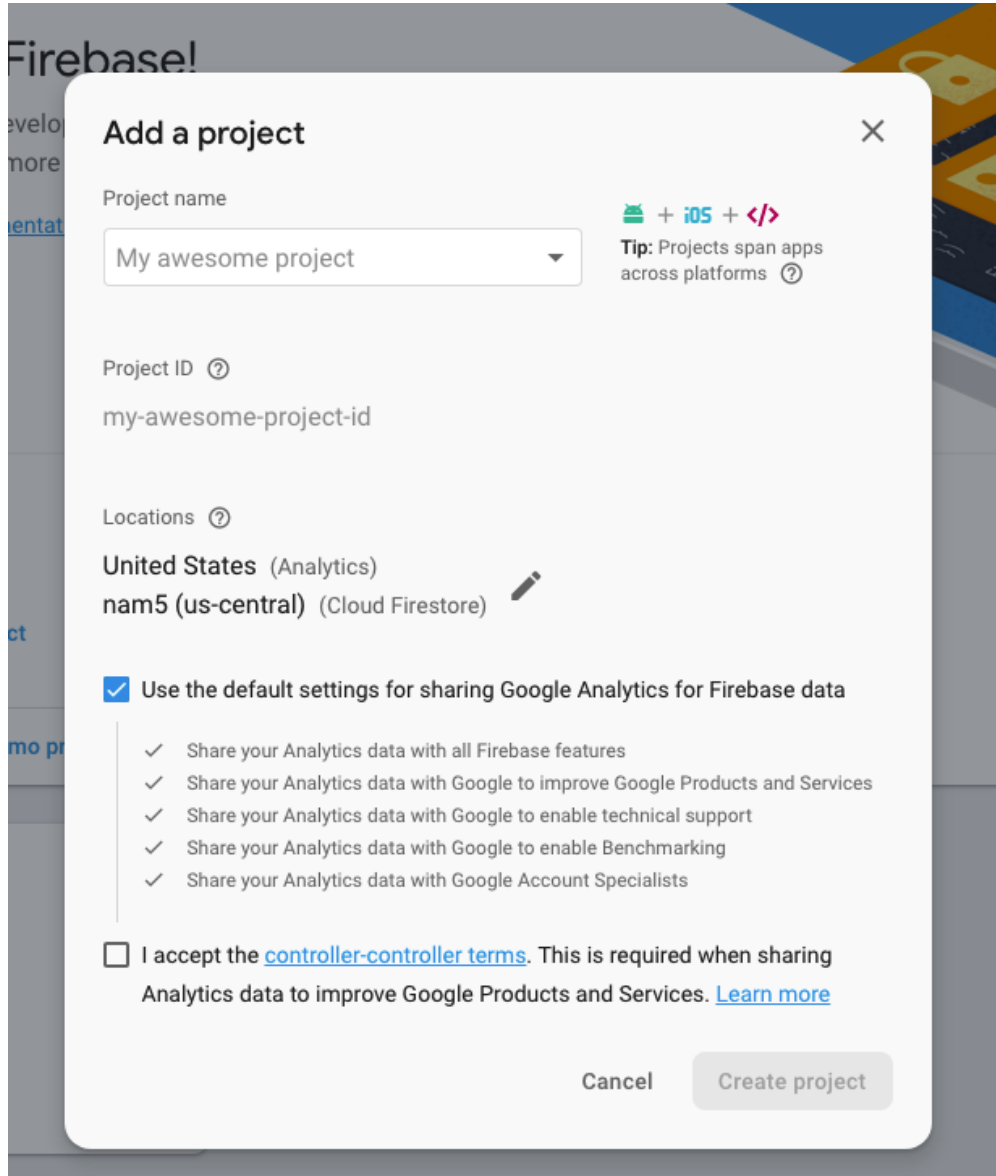
Public-facing name project-[REDACTED]-1

Support email Not configured

Your apps

There are no apps in your project
Select a platform to get started

In **Your apps** section click on icon `</>` to Add Firebase to web app



Add **App nickname**, then click **Register App**, then click **Continue to console** to back to **Project Settings** page

× **Add Firebase to your web app**

- ✓ Register app
- 2 Add Firebase SDK

Copy and paste these scripts into the bottom of your `<body>` tag, but before you use any Firebase services:

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/6.1.0/firebase-app.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
      https://firebase.google.com/docs/web/setup#config-web-app -->

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "[REDACTED]",
    authDomain: "[REDACTED].firebaseapp.com",
    databaseURL: "https://[REDACTED].firebaseio.com",
    projectId: "[REDACTED]",
    storageBucket: "[REDACTED].appspot.com",
    messagingSenderId: "[REDACTED]",
    appId: "1:555767431731:web:b5147f0326945398"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
</script>
```

Learn more about Firebase for web: [Get Started](#), [Web SDK API Reference](#), [Samples](#)

[Continue to console](#)

In **Your Apps** section, chose the app you just created, then click on **Config** option of **Firebase SDK snippet**. Copy all script code and paste to **Firebase Config Object** setting in AdminCP > Apps > Installed > Instant Messaging > Settings area.

Firebase SDK snippet

CDN ? Config ?

Copy and paste these scripts into the bottom of your `<body>` tag, but before you use any Firebase services:

```
const firebaseConfig = {
  apiKey: "[REDACTED]",
  authDomain: "[REDACTED].firebaseapp.com",
  databaseURL: "https://[REDACTED].firebaseio.com",
  projectId: "[REDACTED]",
  storageBucket: "[REDACTED].appspot.com",
  messagingSenderId: "[REDACTED]",
  appId: "1:555767431731:web:b5147f0326945398"
};
```

[Remove this app](#)

2. Enable Firebase Database

Back to Firebase App **Dashboard**, select **Develop > Database**. Then, right-click on **Create Database**. Select mode and click **Enable**

Cloud Firestore

The next generation of the Realtime Database with more powerful queries and

Security rules for Cloud Firestore

After you define your data structure, you will need to write rules to secure your data.
[Learn more](#)

- Start in locked mode**
Make your database private by denying all reads and writes
- Start in test mode**
Get set up quickly by allowing all reads and writes to your database

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if false;
    }
  }
}
```

i All third party reads and writes will be denied

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

Cancel **Enable**

After enabling, you will redirect to database page. In that page, select tab **Indexes** and add 2 **Composite** indexes to Database by click on **Create index manually**.

Then, configure fields as 2 screenshots below:

Create a composite index ✕

Cloud Firestore uses composite indexes for compound queries not already supported by single field indexes (ex: combining equality and range operators).

Collection ID

Fields to index

At least two fields required*

1	receiver	Ascending	▼
2	seen	Ascending	▼
3	time_stamp	Ascending	▼

[Add field](#)

Query scopes

Collection
For queries within a specific collection path

[Cancel](#) [Create index](#)

Creation time depends on the amount of data being indexed

Create a composite index

Cloud Firestore uses composite indexes for compound queries not already supported by single field indexes (ex: combining equality and range operators).

Collection ID

Fields to index
 At least two fields required*

1	deleted	Ascending
2	text	Ascending
3	time_stamp	Descending

[Add field](#)

Query scopes
 Collection
 For queries within a specific collection path

[Cancel](#) [Create index](#)

Creation time depends on the amount of data being indexed

Indexes enable:

IM Chat [Go to docs](#)

Database Cloud Firestore

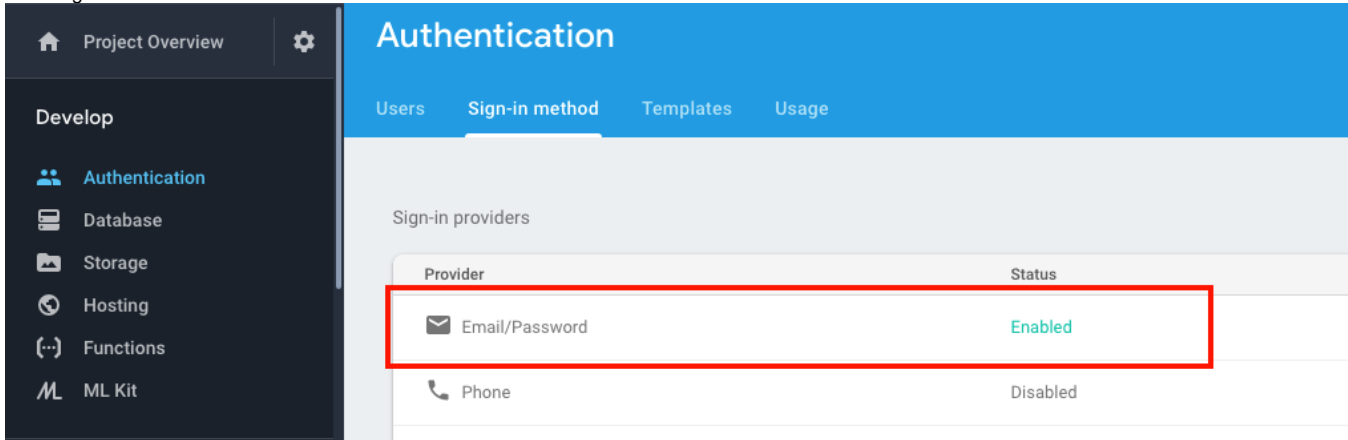
Data Rules **Indexes** Usage

[Composite](#) [Single field](#)

[Add index](#)

Collection ID	Fields indexed	Query scope	Status
messages	deleted Ascending text Ascending time_stamp Descending	Collection	Enabled
messages	receiver Ascending seen Ascending time_stamp Ascending	Collection	Enabled

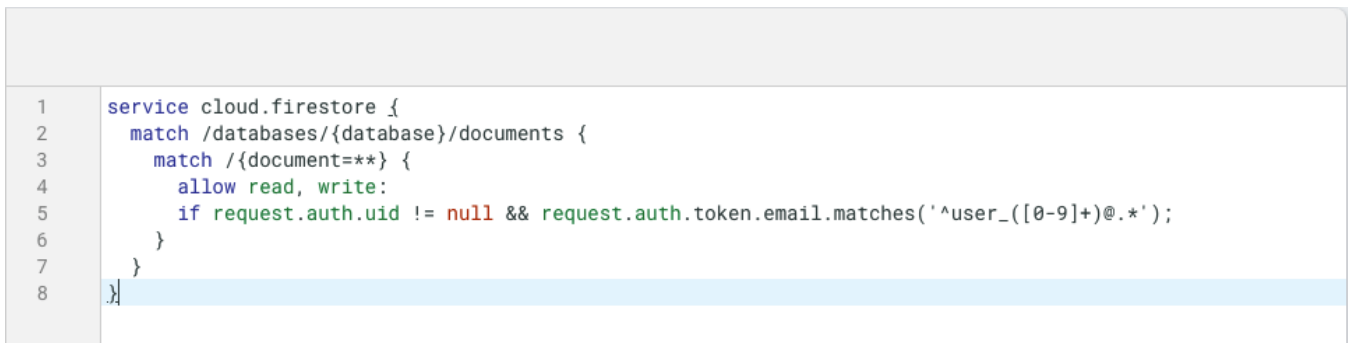
Go to Develop > Authentication, enable Sign-in method for **Email/Password provider** in your project to allow users access to the database, as the following screenshot:



Then, go to **Database > Rules** tab and add the following database rules:

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write:
        if request.auth.uid != null && request.auth.token.email.matches('^user_([0-9]+)@.*');
    }
  }
}
```

Rules table will be as the below image:



Now, you can start to use IM App with Firebase.

Setup Algolia to support search messages

Note: Without Algolia configure on IM App, users can only search for messages with prefix and case sensitive.

Go to [Algolia App Manage](#), sign up if you don't have an account

Note: Algolia is pay to use, so you should check their pricing at [Algolia Pricing](#)

Go to your **App Dashboard**

Applications

LTERNON5IJ **DASHBOARD** TEAM EDIT DANGER ZONE

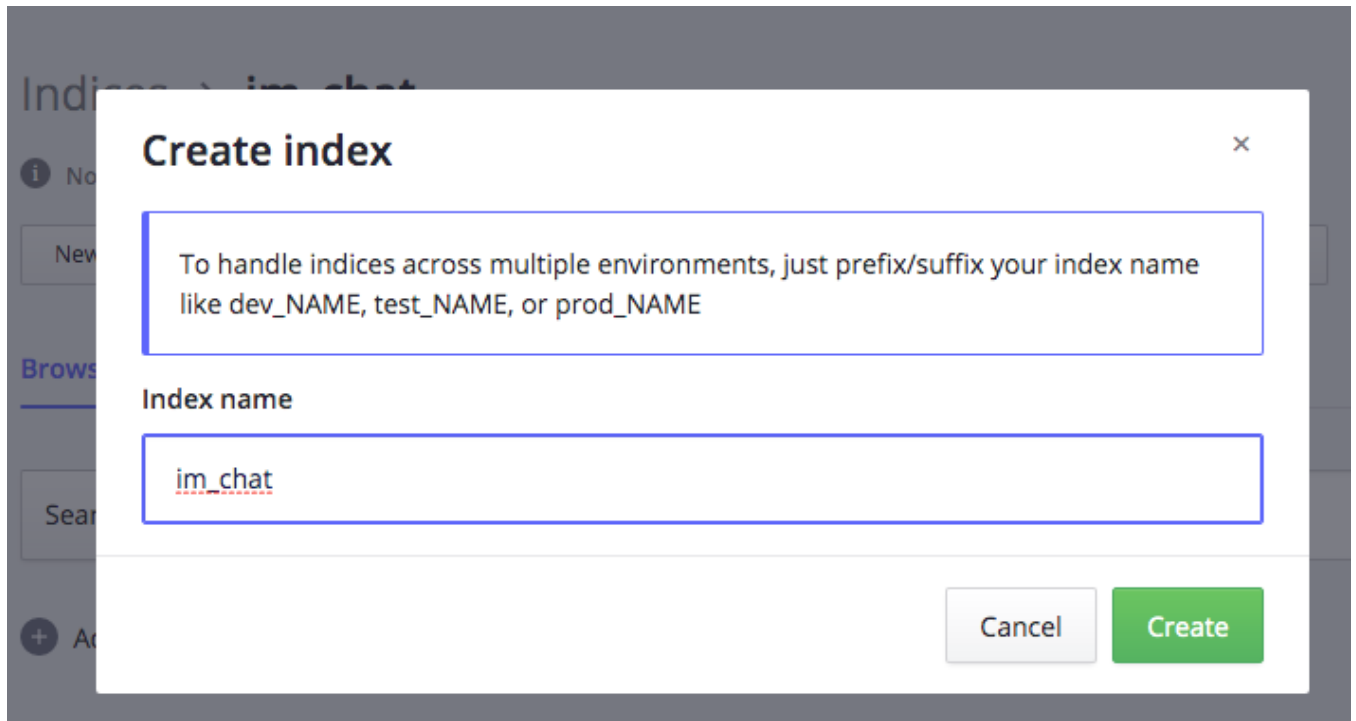
TRIAL 13 days remaining Hong-Kong

OPERATIONS 890 / 25M

RECORDS 1K / 5M

3 INDICES getstarted_actors, rating_a...

Select **Indices**, create new Index with name **im_chat**



After Create Index, select menu **API keys** on left

- Copy **Application ID** and paste to setting **Algolia App ID** on IM App
- Copy **Admin API Key** and paste to setting **Admin API Key** on IM App

algolia

LTERNON5IJ (UNNAMED APP)

Overview Indices Analytics Query suggestions Monitoring Infra API keys Team Billing

Your API Keys All API Keys

Application ID LTERNON5IJ

This is your unique application identifier. It's used to identify you when using Algolia's API.

Search-Only API Key

This is the public API key to use in your frontend code. This key is only usable for search queries.

Regenerate

Admin API Key

This is the ADMIN API key. Please keep it secret and use it ONLY from your backend: this key is used to create, update and DELETE your indices. You can also use it to manage your API keys.

Now, users can search full-text messages with IM App and Firebase