



# News Hour Admin Doc (v4.0.7)

09-03-22 Last updated

---

*//only for existing buyers*

**Migrate v4.0.X : follow this doc : [doc](#)**

It's an offline document. There is an online documentation [here](#). We are highly recommending the online version. Because it will always be updated."

## Introduction

Welcome to News Hour documentation & thank you for buying our product. So there are two setup steps here. Admin panel & App. Both are developed on Flutter. So, You have to install flutter on your computer. You can use Both **Visual Studio Code** & **Android Studio** for Flutter. The steps are the same for both IDE.. We have used Visual Studio Code. So, Our Setup will be based on this IDE.

## 1. Flutter Installation

First, You have to install flutter on your computer. To install flutter on your computer, follow the official documentation from Google.

Flutter Official Site : <https://flutter.dev/>.

You can follow these youtube videos to install flutter also.

1. For Mac: <https://www.youtube.com/watch?v=9GuzMsZQUYs>
2. For Windows: <https://www.youtube.com/watch?v=T9LdScRVhv8>

Make sure you have installed the latest stable version of flutter. If everything is okay then you can follow the further steps below.

## 2. Admin Panel Setup

After successfully installing Flutter on your computer, You have to enable Flutter web support. Make sure you have installed the stable channel.

### 2.1 Flutter Web Setup

From Flutter stable version 2.0.0+, it will support the web. So, If you are already on the latest stable channel of flutter, you don't have to do anything for the flutter web. **You should run all the commands on your IDE terminal from the project directory.**

1. Open the **source/news\_admin** folder with your terminal and wait some moment to load the admin project.
2. After that run the following command to clean the whole project.

```
flutter clean
```

3. After that run the following command to get the required packages.

```
flutter pub get
```

4. After that run the following command to enable web support for your project.

```
flutter config --enable-web
```

And then it will enable the web feature for your project. That's it. Your admin project initialization is complete.

## 2.2 Firebase(Database) Setup for Admin Panel :

1. We have used the firestore **database** as the backend for this project. First go to the [Firebase Console](#) and sign in with your gmail account and go to the console.
2. Create a project by your app name. And go to the project overview.
3. Click on the **plus icon** and then click on the **web icon**. You will see a popup like the second picture below :

The image shows two screenshots from the Firebase console. The top screenshot displays the 'News Hour' project overview with analytics for daily active users and day 1 retention. The bottom screenshot shows the 'Add Firebase to your web app' wizard, specifically the 'Register app' step where an app nickname is entered as 'My web app'.

**Analytics Data (Top Screenshot):**

Metric	Value	Change
Daily active users	48	+41.2%
Day 1 retention	21.4%	+39.3%

**App Performance Data (Bottom Screenshot):**

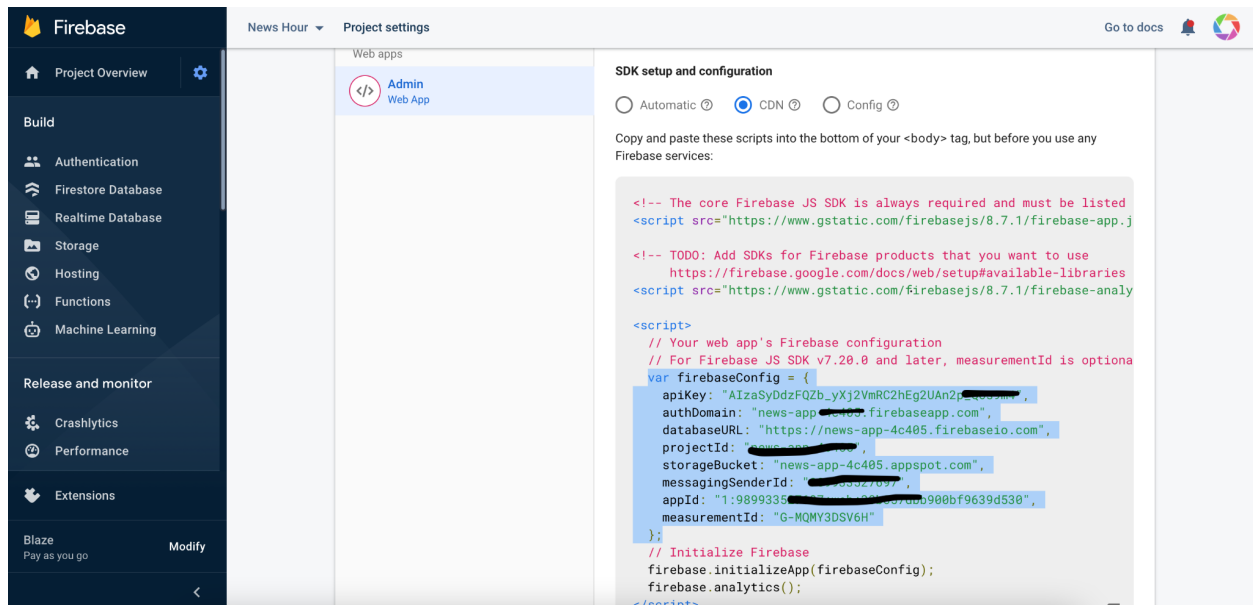
App	DAU	Day 1 retention	Revenue
News App Android	39	21.4%	
Admin Android	0	0%	

**Setup Wizard (Bottom Screenshot):**

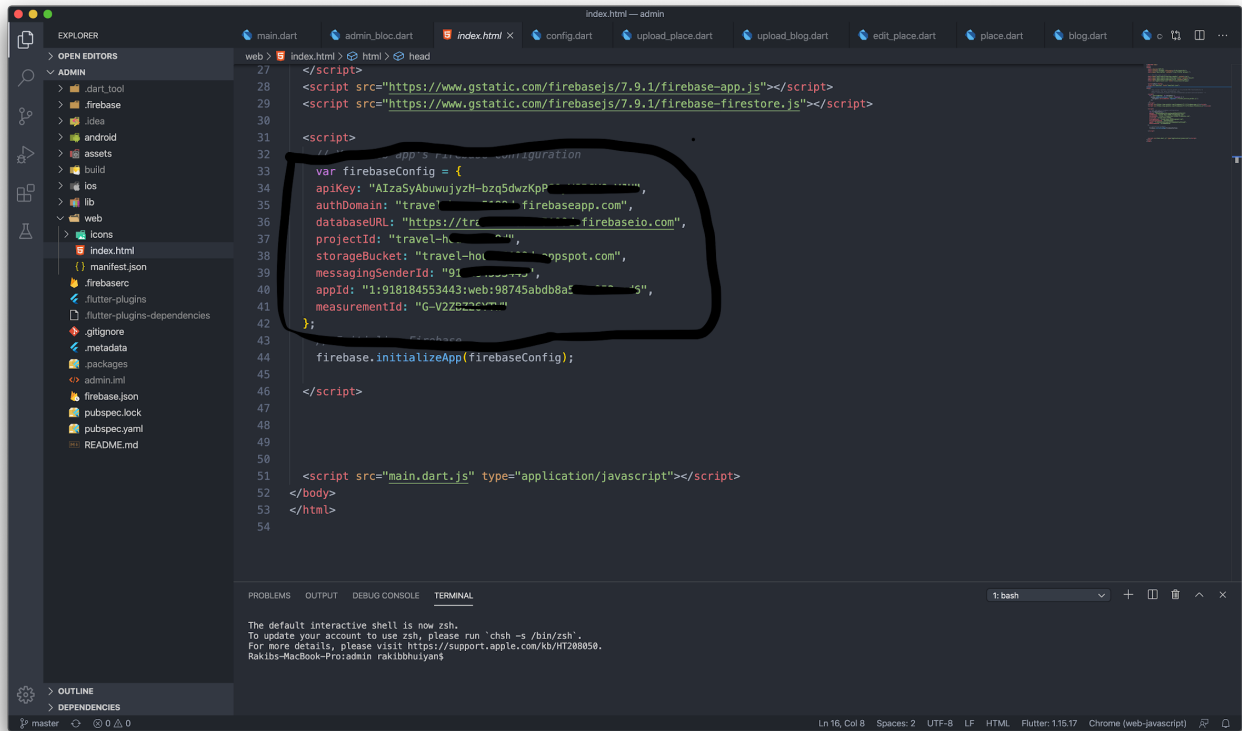
- Register app**
  - App nickname: My web app
  - ☐ Also set up **Firebase Hosting** for this app. [Learn more](#)
  - Register app
- Add Firebase SDK**

- Input the name like as **Admin Panel** and fill the checkbox because you need hosting too and then click on **register app**. After that, Skip the other options by clicking next. (Don't input custom name for hosting else you need extra setup for this which is not recommended)

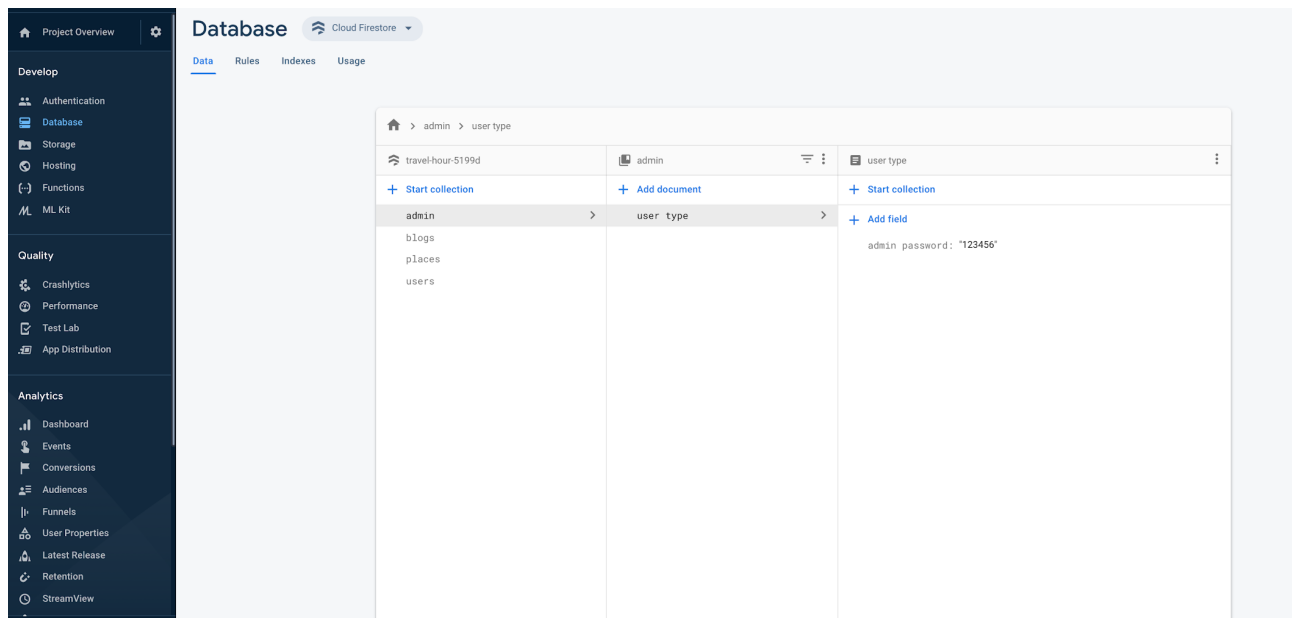
- Now go to the Project settings and in the bottom section click on the **Admin panel** that you have just created.
- After that click on **CDN** and copy the selected code which is shown in the picture below.



- Now, go to your admin panel project with your IDE.
- Go to **web/index.html** file and replace the selected code with your config code (what you have just copied from the firebase console). Look at the picture below:



9. Now go to the **Firestore Database** and create a **database** as test mode.
10. Now, Create a collection as **admin** and create a document as **user type** and create a textfield as **admin password** and use your password as **value**. Just like the picture below (Only the admin section). Try to use a unique password. This will be the password of your admin panel. You can change the password from here anytime. You can also change the password from your admin website.



## 2.3 Database Rules Setup :

Go to your **firebase console** > **project overview** > **database** > **firestore database** > **rules** and then edit the rules like this and click publish.

```

1  rules_version = '2';
2  service cloud.firestore {
3    match /databases/{database}/documents {
4      match /{document=**} {
5        allow get: if true;
6        allow list: if true;
7        allow create: if request.auth != null;
8        allow delete: if false;
9        allow update: if request.auth != null;
10   }
11
12   match /users/{userId}{
13     allow update: if (request.auth.id == userId);
14     allow delete: if false;
15   }
16 }
17 }
```



## 2.4 Database Index Setup :

Now go to the right tab (**Indexes**) of the same page. You have to create 4 indexes one by one. Click on **Add Index** and then create an index by following the list.

1. Enter **contents** as Collection ID, **`content type`** into the first field and set **Ascending**, Enter **loves** in the second field and set **Descending** and then select **Collection** as query scope.
2. Enter **contents** as Collection ID, **`content type`** into the first field and set **Ascending**, Enter **timestamp** in the second field and set **Descending** and then select **Collection** as query scope.
3. Enter **contents** as Collection ID, **category** into the first field and set **Ascending**, Enter **timestamp** in the second field and set **Descending** and then select **Collection** as query scope.
4. Enter **contents** as Collection ID, **category** into the first field and set **Ascending**, Enter **timestamp** in the second field and set **Ascending** and then select **Collection** as query scope.

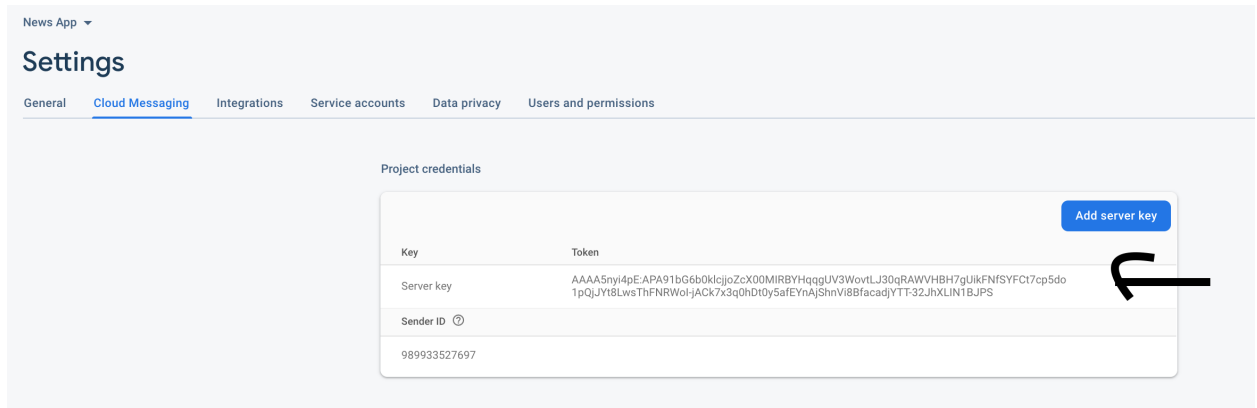
After completing this, your index page should look like this :

Composite		Single field	
Collection ID	Fields indexed	Query scope	Status
contents	`content type` Ascending loves Descending	Collection	Enabled
contents	`content type` Ascending timestamp Descending	Collection	Enabled
contents	category Ascending timestamp Descending	Collection	Enabled
contents	category Ascending timestamp Ascending	Collection	Enabled

That's it. Your database setup is complete.

## 2.5 Push Notification Setup

1. Now, You have to set up push notifications to notify your users. To do that, go to **project settings > cloud messaging** tab and copy the **Server Key**.



2. Now go to your project and open **lib/config/config.dart** file and replace the **server token** that you have just copied.

```

1
2 class Config{
3   final String appName = 'News Hour'; //app name
4   final String testerPassword = 'tester12345'; //testing password - don't use this
5
6   //firebase server token for push notification
7   final String serverToken = 'AAAA5nyi4pE:APA91bG6b0klcjjoZcX0*****';
8   final String icon = 'assets/images/icon.png'; // app icon
9
10
11
12
13   //don't edit or remove this
14   final List contentTypes = [
15     'image',
16     'video'
17   ];
18 }

```

## 2.6 Other Setup

1. Change the **app name**. (Look at the previous picture)
2. Now you have to change the app icon. First rename your icon to **icon** and make sure the icon is in **.png** format. Now drag & drop your icon to the **assets/images** folder and replace it with our icon.
3. That's it. Your Admin Setup is complete. You can run the admin app now.

## 3. Run The Admin App

1. You can run the admin app now running the following command in the terminal.

```
flutter run -d chrome --web-renderer html
```

2. After running the admin app, Use the password that you have saved in the database before and check if everything is okay or not.

## 4. Upload the Admin App to the Firebase Hosting

1. You have to upload your site to your firebase hosting. For this, you have to install firebase tools on your IDE. Run this command on the terminal:

It will download firebase tools for you. Or you directly download from [here](#) and install this manually.

```
npm install -g firebase-tools
```

2. After that, You need to login to the firebase console with your google account. To do that, Run:

```
firebase login
```

Follow the command prompt and the link in your browser. Accept the permissions by clicking Ok. When you return to your terminal you should see that you are now logged in:

```
Waiting for authentication...  
✓ Success! Logged in as [REDACTED]  
antrob-macbookpro3:flutter_web_test antrob$
```

3. Now you have to initialize the firebase. Run this command:

```
firebase init
```

```
antrob-macbookpro3:flutter_web_test antrob$ firebase init

*****
**                                     **
** 00 00 00 00 00 00 00 00 00 00 00 **
** 00 00 00 00 00 00 00 00 00 00 00 **
** 00 00 00 00 00 00 00 00 00 00 00 **
** 00 00 00 00 00 00 00 00 00 00 00 **
** 00 00 00 00 00 00 00 00 00 00 00 **
*****

You're about to initialize a Firebase project in this directory:

/Users/antrob/StudioProjects/flutter_web_test

Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices. (Press <space> to select, <tab> to toggle all, <shift> to invert selection)
o Database: Deploy Firebase Realtime Database Rules
o Firestore: Deploy rules and create indexes for Firestore
o Functions: Configure and deploy Cloud Functions
x> Hosting: Configure and deploy Firebase Hosting sites
o Storage: Deploy Cloud Storage security rules
```

4. Use the arrow keys to navigate the cursor to Hosting and hit the spacebar to select it, then press enter. Now you will see this screen:

```
You're about to initialize a Firebase project in this directory:

/Users/antrob/StudioProjects/flutter_web_test

Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices. Hosting: Configure and deploy Firebase Hosting sites

== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

Please select an option: (Use arrow keys)
> Use an existing project
  Create a new project
  Add Firebase to an existing Google Cloud Platform project
  Don't set up a default project
```

5. Select an existing project by pressing Enter. Use the arrow keys to select the project you have made in the firebase console.
6. Next, enter **build/web** as the public directory and press enter, then enter y (for yes) to select the single page app option:

Your **public** directory is the folder (relative to your project directory) that will contain Hosting assets to be uploaded with **firebase deploy**. If you have a build process for your assets, use your build's output directory.

```
? What do you want to use as your public directory? build/web
? Configure as a single-page app (rewrite all urls to /index.html)? (y/N) y
```

7. Now Run this command to build the web app.

```
flutter build web --web-renderer html --release
```

8. After that, run this command to upload the app to your hosting and wait for some moment.

```
firebase deploy
```

That's it. Your Admin Panel is 100% complete. You will get a url from the terminal and it will be the domain of your admin panel and you can use this domain for further use of the admin panel. You can also get the domain from your **firebase console > your app > hosting page**.

Now, Go for the app setup!

## 5. Before the App Setup (Impotant!!!!!!!!!!)

You have to upload some categories & articles. Otherwise the app will show nothing at all and can cause errors. In order to upload articles, you have to upload at least **4 categories which are mandatory!!** Because these 4 categories will be static in the app for UI purposes.

*Make sure you have uploaded at least 4 categories, 1 video article & 1 image article.  
And also add at least 1 item to the featured list.*