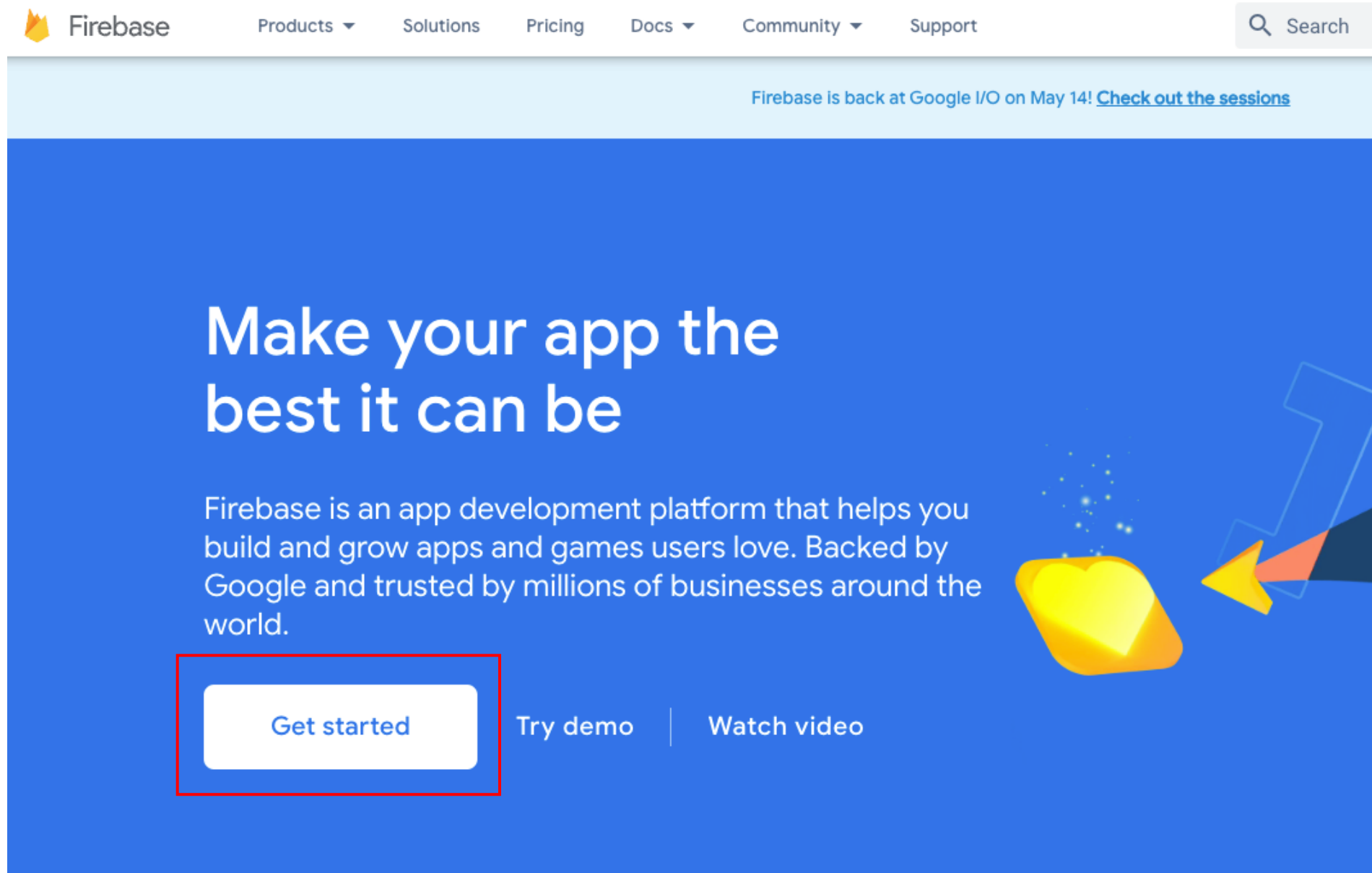


Firestore in Flutter setup

2024 SS

Go to firebase and cerate a firebase project



Welcome to Firebase!

Tools from Google for building app infrastructure, improving app quality, and growing your business

Create a project

 [View docs](#) 

Create firebase project




× Create a project (Step 1 of 3)

Let's start with a name for
your project[?]

Project name

todoapp

 todoapp-3d5c1

 Select parent resource

Continue

Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

Google Analytics enables:

✕ A/B testing ?

✕ Breadcrumb logs in Crashlytics ?

✕ User segmentation & targeting across
Firebase products ?

✕ Event-based Cloud Functions triggers ?

✕ Free unlimited reporting ?



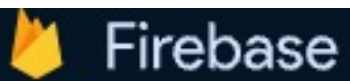
Enable Google Analytics for this project
Recommended

Disable first

Previous

Then create

Create project



Your Firebase projects



Add project

Go into the project you just created

todoapp

todoapp-3d5c1

 datalab.cs.nthu.edu....

todoapp

Spark plan

Get started by adding Firebase to your app



Add an app to get started



1 Prepare your workspace

The easiest way to get you started is to use the FlutterFire CLI.

Before you continue, make sure to:

- Install the [Firebase CLI](#) and log in (run `firebase login`)
- Install the [Flutter SDK](#)
- Create a Flutter project (run `flutter create`)

Next

2 Install and run the FlutterFire CLI

3 Initialize Firebase and add plugins

Install firebase CLI

- [Windows](#)
- [MacOS](#)
- **We recommend using npm install**
 - Install node version manager(nvm)
 - Add nvm to PATH([windows](#), MacOS: should be automatic)
 - \$ nvm install 20.13.0
 - \$ nvm use 20.13.0
 - \$ nvm run
 - \$ npm install -g firebase-tools ([Windows problem](#))

Login Firebase CLI with Google account

- After Firebase CLI is installed In terminal, run **\$ firebase login**
- Follow the pop-up login window and login to the Google account with a billing account

```
victor890915@VictordeMacBook-Pro ~ % firebase login
i  Firebase optionally collects CLI and Emulator Suite usage and error reporting
   information to help improve our products. Data is collected in accordance with
   Google's privacy policy (https://policies.google.com/privacy) and is not used to
   identify you.

?  Allow Firebase to collect CLI and Emulator Suite usage and error reporting
   information? No

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fgrhgmd47bqneki
j5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%
2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww
.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcl
oud-platform&response_type=code&state=253403360&redirect_uri=http%3A%2F%2Flocalh
ost%3A9005

Waiting for authentication...

✓  Success! Logged in as swcheng@datalab.cs.nthu.edu.tw
victor890915@VictordeMacBook-Pro ~ %
```

Go back to firebase page and click next

2

Install and run the FlutterFire CLI

From any directory, run this command:

```
$ dart pub global activate flutterfire_cli
```



Then, at the root of your Flutter project directory, run this command:

```
$ flutterfire configure --project=todoapp-3d5c1
```



This automatically registers your per-platform apps with Firebase and adds a `lib/firebase_options.dart` configuration file to your Flutter project.

[Previous](#)

[Next](#)

In your flutter project

- In project directory , run `$ dart pub global activate flutterfire_cli`
- Then run `$flutterfire configure --project=xxxxxxxxxx`
(copy from firebase page)

```
victor890915@VictordeMacBook-Pro todo % flutterfire configure --project=todoapp-3d5c1
i Found 1 Firebase projects. Selecting project todoapp-3d5c1.
? Which platforms should your configuration support (use arrow keys & space to select)? >
✓ android
✓ ios
✓ macos
✓ web
✓ windows
```

Choose the platform that you'll use

(arrow key to select , space to toggle, enter to choose)

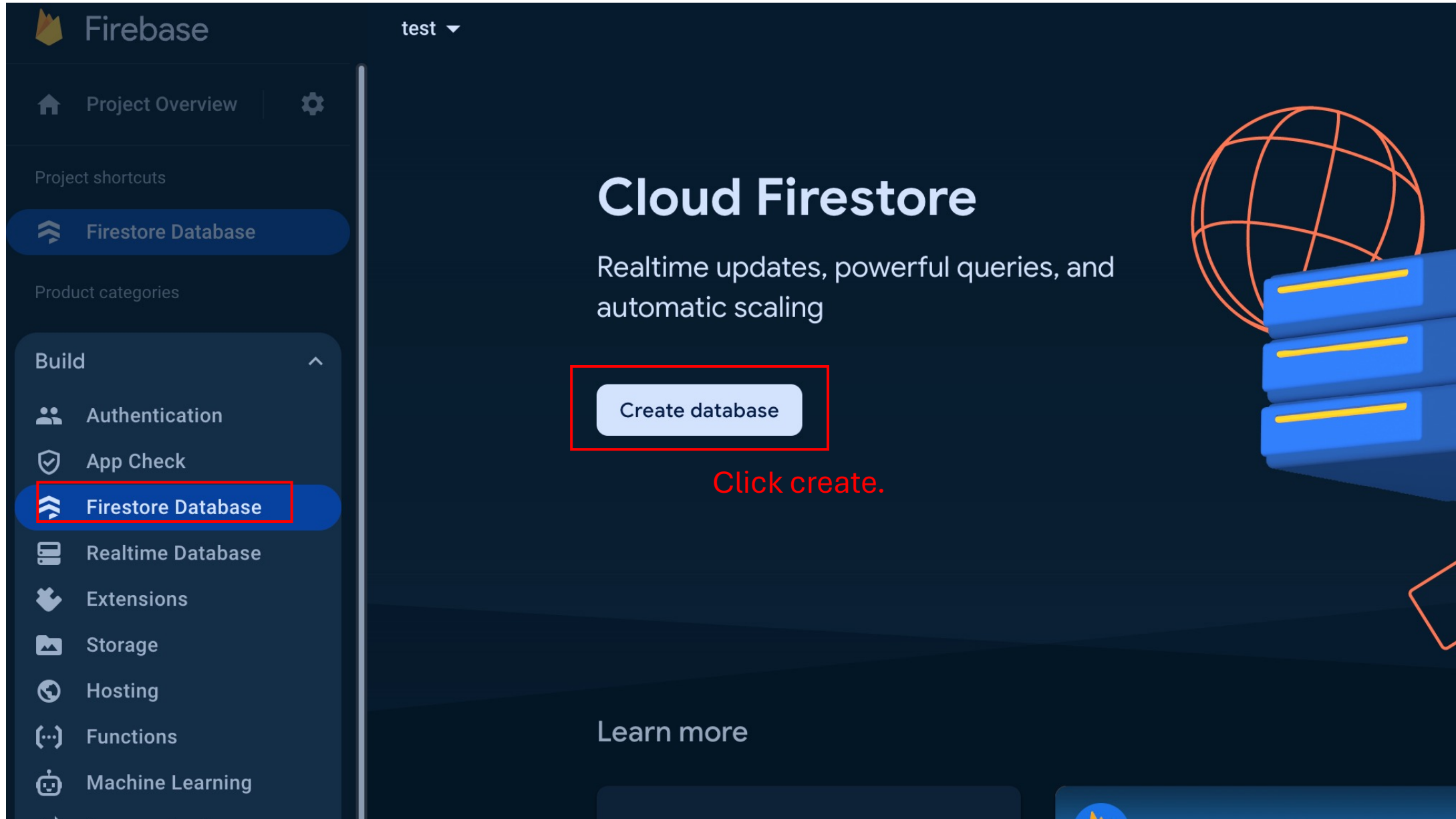
In your flutter project

- Choose the firebase project you've created
- After running flutter configure, you should see `firebase_options.dart` under `/lib`

Possible problems

- Insecure world writable dir in PATH, mode 040777
- Flutterfire not in PATH

Go back to FireBase console



Build/Firestore
Database

Click create.

Create database




- 1 Set name and location
- 2 Secure rules

Database ID

Location

 Your location setting is where your Cloud Firestore data will be stored

 After you set this location, you cannot change it later. If this is your first database, your default Cloud Storage location will also be set to this location.

[Learn more](#) 

Cancel

Next

Create database



Set name and location

2

Secure rules

After you define your data structure, **you will need to write rules to secure your data.**

[Learn more](#)



Start in **production mode**

Your data is private by default. Client read/write access will only be granted as specified by your security rules.



Start in **test mode**

Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2024, 6, 8);
    }
  }
}
```



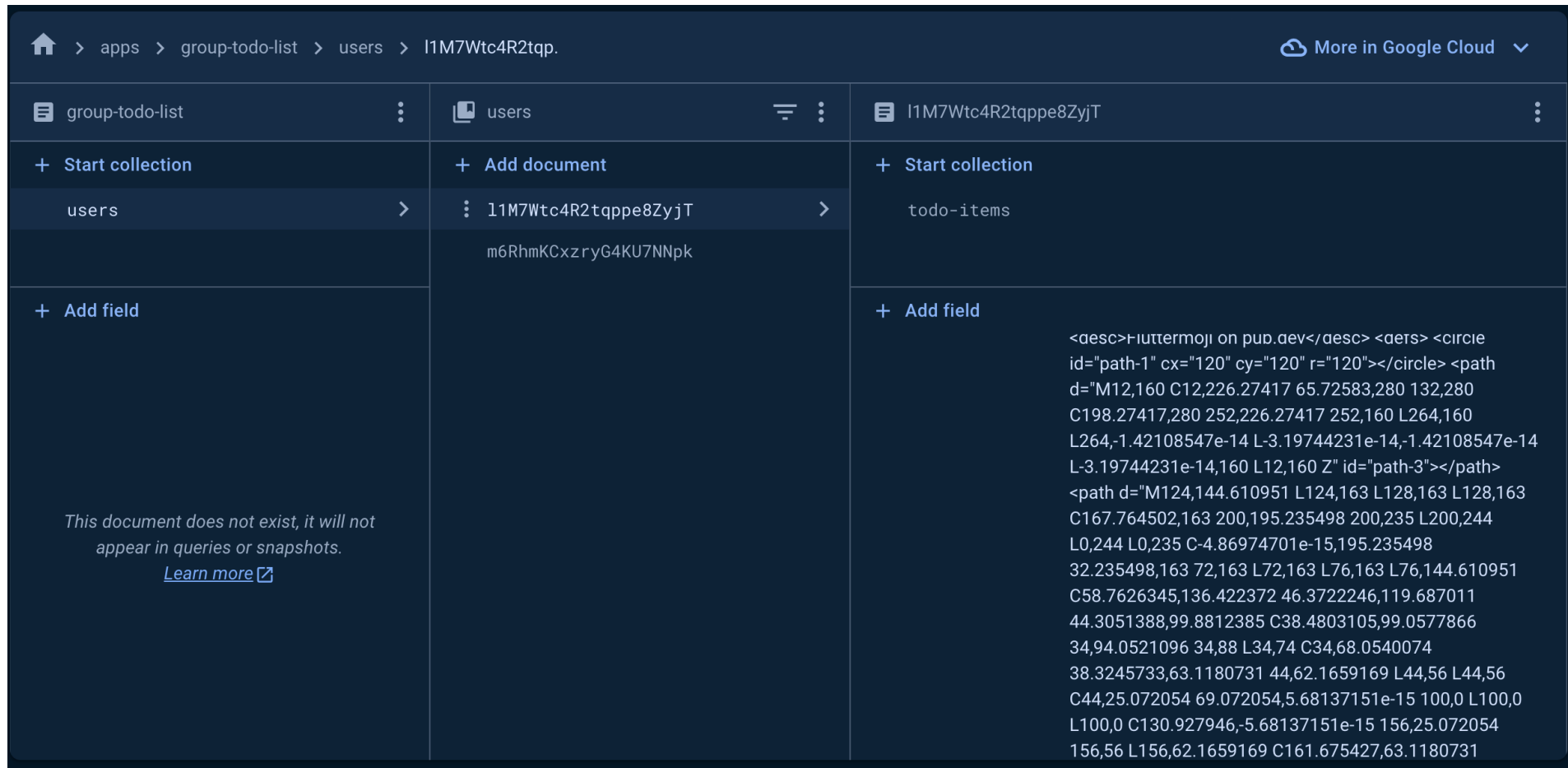
The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Cancel

Create

Test

- Add a new user, close app and reopen, the Users should persist



Users are now stored in Firebase database

Setup security rules

- Run `$ firebase init firestore`, there should be two new files (firestore.rules , firestore.indexes).
- These are system generated rules, but we have written our own rules, so delete them
- Move the “resources/firestore” folder out to the surface level
- Because we organized them in a folder, we must **update the firebase.json**

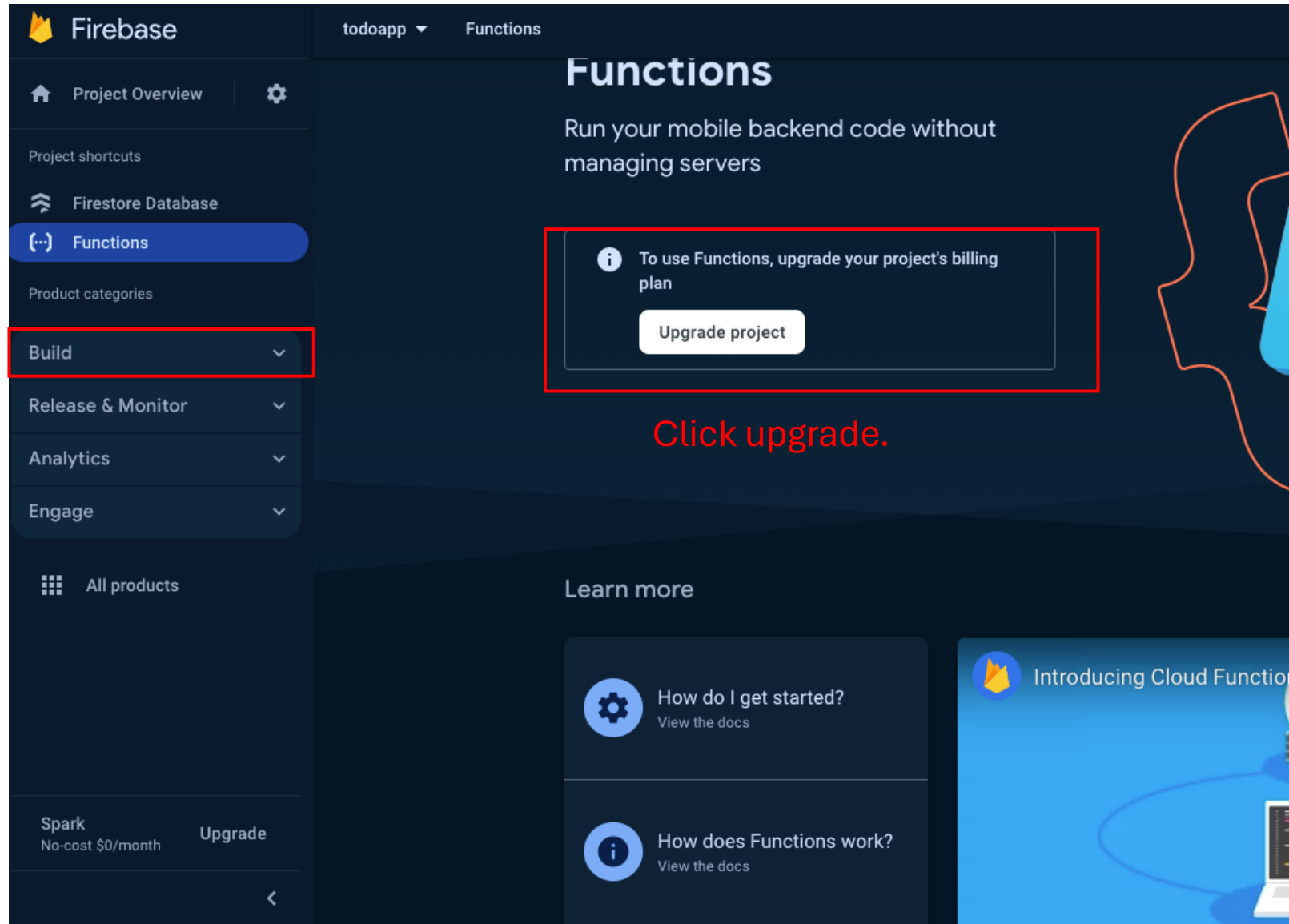
```
},  
"firestore": {  
  "rules": "firestore/firestore.rules",  
  "indexes": "firestore/firestore.indexes.json"  
},
```

Modify the path and save

- Run `$ flutterfire configure`
 - Reuse `firebase.json`
- **Deploy to firebase:** `$ firebase deploy --only firestore:rules`

Go back to FireBase console

Build/Functions



Your billing account



Selected plan

Blaze Pay as you go

[See full plan details](#)

To continue, please select a billing account

Your billing accounts

My Billing Account

[View account](#)


Cancel


Continue

Choose the
billing account
you've created

← Set a billing budget



Billing to 

My Billing Account [View account](#) 

Send an email to all administrators on the billing account when cost for the project approaches or exceeds this amount.

Budget amount in TWD

\$ 10

You'll get an email at \$5.00 (50%), \$9.00 (90%), \$10.00 (100%), and can customize this later in the [Google Cloud Console](#) .

Skip this step

Continue

Set a billing budget

Functions

Run your mobile backend code without managing servers

Get started

System will tell you to install a tool using npm

If you've done this, you can skip

Set up Functions

1

Install

2

Deploy

To use Functions, you need to install Firebase command line tools using npm ([Node.js](#))

Install Firebase tools:

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



Not working? You may need to [change npm permissions](#)

If you've previously installed Firebase command line tools, run the install command again to make sure you have the latest version

Cancel

Continue

Setup cloud functions

- Move the “resources/functions” folder out to the surface level
- Run `$ firebase init functions`
 - Language: JavaScript
 - Use ESLint: NO
 - Index.js exists, overwrite?: No
 - Install npm dependencies: Yes
- Run `$ flutterfire configure`
 - Reuse firebase.json

- Deploy function to firebase by using:

```
$firebase deploy --only functions
```



- In “/android/app/build.gradle”, change `minSdkVersion` to 21 or higher

Back to Firebase console.

Functions

Dashboard

Usage

Function	Trigger	Version	Requests (24 hrs)	Min / Max Instances	Timeout
shoppingAppDecrementUserItemCou... us-central1	 document.deleted	v2	1	0 / 100	1m
shoppingAppIncrementUserItemCou... us-central1	 document.created	v2	2	0 / 100	1m

Successfully deployed cloud functions

Items per page: 25 ▼ 1 – 2 of 2 < >

- Test by assigning to-do's to other user and see if the if the user's number of to-do items increases