

React Advance

Bài 5: Giới thiệu Firebase



Nội dung

1. Giới thiệu Firebase
2. Tạo project Storage với Firebase
3. Kết nối NodeJS app với Firebase



1. Giới thiệu Firebase



Giới thiệu Firebase

- Firsebase là một nền tảng ứng dụng di động và web với các công cụ và hạ tầng được thiết kế để giúp các lập trình viên xây dựng các ứng dụng chất lượng cao.
- Firebase (tiền thân là Evolve) trước đây là một startup được thành lập vào năm 2011 bởi Andrew Lee và James Tamplin.
- Ban đầu, Evolve chỉ cung cấp cơ sở dữ liệu để các lập trình viên thiết kế các ứng dụng Chat. Tuy nhiên họ nhanh chóng nhận ra tiềm năng sản phẩm của mình khi nhận thấy các khách hàng không sử dụng Cơ sở dữ liệu để làm ứng dụng Chat. Mà thay vào đó, họ sử dụng Firebase để lưu các thông tin như Game Progress



Giới thiệu Firebase

- Bộ đôi Lee và Tamplin quyết định tách mảng realtime ra để thành lập một công ty độc lập – chính là Firebase vào tháng 4 năm 2012.
- Sau nhiều lần huy động vốn và gặt hái được những thành công nổi bật, Firebase đã được Google để ý.
- Tháng 10 năm 2014, Firebase chính thức gia nhập vào gia đình Google
- Từ đây, Firebase có điều kiện để phát triển, mở rộng số lượng dịch vụ con.
- Hiện tại, Google đã chuyển các dịch vụ nền tảng hỗ trợ các lập trình viên bên ngoài về cho Firebase quản lý chẵn hạn như Cloud Mesageing, AdMod và Analytics



Giới thiệu Firebase





Giới thiệu Firebase

- Firebase có thể cung cấp rất nhiều dịch vụ hỗ trợ các lập trình viên xây dựng các ứng dụng (Web/Mobile)
- Các dịch vụ tiêu biểu mà hệ thống Firebase hỗ trợ
 - **Firebase Hosting & Storage** → chứa cơ sở dữ liệu
 - **Firebase Authentication** → quản lý phần quyền và đăng nhập
 - **Firebase Realtime Database** → cơ sở dữ liệu theo thời gian thực, đáp ứng phản hồi cho các loại ứng dụng yêu cầu xử lý đồng bộ thời gian thực như Chat
 - **Firebase Cloud Messaging** → hỗ trợ gửi thông báo (notification) đến các thiết bị
 - **Test Lab, Crash Reporting, App Indexing** → hỗ trợ báo cáo các vấn đề có thể xảy ra khi ứng dụng đang chạy trên nhiều máy thật.
 - **Remote Config** → giúp thay đổi các thiết lập trên ứng dụng của người dùng mà không cần phải public version mới



Giới thiệu Firebase

- Data (dữ liệu) được lưu trữ dưới dạng JSON và được đồng bộ theo thời gian thực cho tất cả các connection từ Client
- Hỗ trợ Offline → khi có kết nối lại với Server, thì những dữ liệu chưa được đồng bộ về Server thì Firebase sẽ tự động đồng bộ lại về Server
- Dễ dàng chứng thực người dùng từ các ứng dụng Android, iOS và JavaScript SDKs
- Hỗ trợ Authorization với mạng xã hội như Facebook, Twitter, Github, Google ...
- Xử lý user login với Client-side code

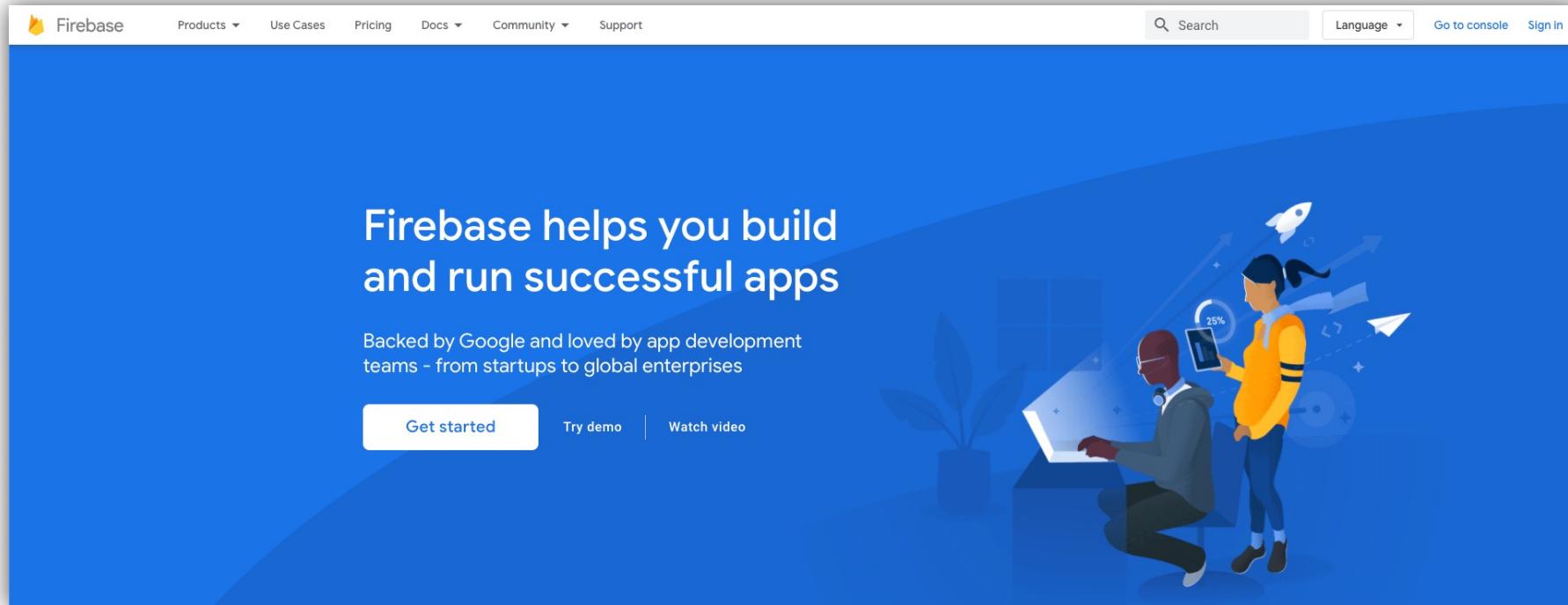


2. Tạo project Storage với Firebase



Tạo project Storage với Firebase

- Truy cập địa chỉ <https://firebase.google.com/>
- Đăng nhập với tài khoản Gmail --> Sau đó chọn *Get started*





Khởi tạo Project

The screenshot shows the Firebase welcome page. At the top left is the Firebase logo (a yellow flame icon) and the word "Firebase". At the top right are links for "Go to docs", a bell icon, and a user profile icon with the letter "E". The main heading is "Welcome to Firebase!" followed by a subtext: "Tools from Google for building app infrastructure, improving app quality, and growing your business". Below this are two buttons: "Create a project" (highlighted with an orange arrow from a callout box) and "View docs". To the right is a stylized illustration of two people working: one sitting at a desk with a laptop, and another standing and writing on a whiteboard. The bottom section contains two cards: "Explore a demo project" (with an "iOS" icon) and "Firebase projects are containers for your apps" (with icons for TV, iOS, code, signal, users, and cloud). A "Learn more" link is also present.

Chọn vào đây để khởi tạo Project



Điền thông tin Project

X Create a project (Step 1 of 3)

Let's start with a name for your project®

Project name
HelloWorld

I accept the [Firebase terms](#)

Continue

1. Điền tên Project

2. Chọn để đồng ý

3. Chọn vào đây để sang bước kế tiếp



Thông tin kích hoạt Google Analytics

×

Create a project (Step 2 of 3)

Google Analytics for your Firebase project

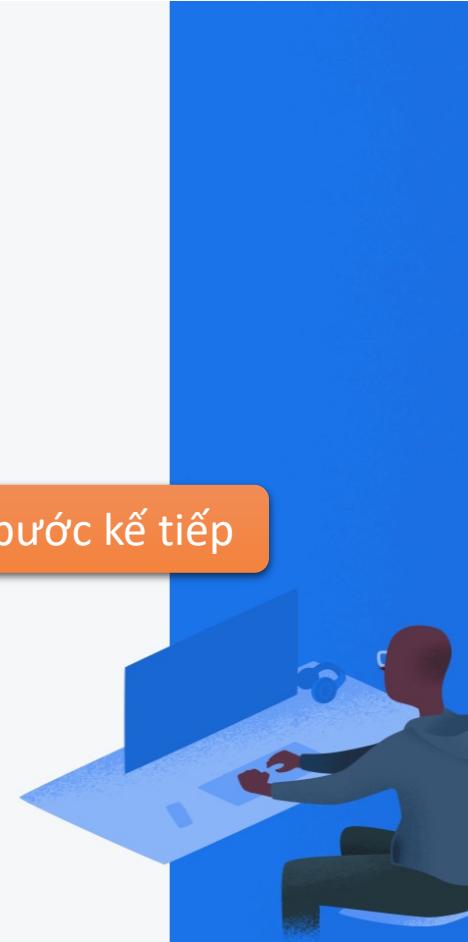
Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions, and Cloud Functions.

Google Analytics enables:

- A/B testing
- Crash-free users
- User segmentation & targeting across Firebase products
- Event-based Cloud Functions triggers
- Predicting user behavior
- Free unlimited reporting

Enable Google Analytics for this project
Recommended

[Previous](#) [Continue](#)



Chọn vào đây để sang bước kế tiếp



Setup thông tin Google Analytics

X Create a project (Step 3 of 3)

Configure Google Analytics

Analytics location ⑦

Vietnam

Data sharing settings and Google Analytics terms

Use the default settings for sharing Google Analytics data. [Learn more](#)

- Share your Analytics data with Google to improve Google Products and Services
- Share your Analytics data with Google to enable Benchmarking
- Share your Analytics data with Google to enable Technical Support
- Share your Analytics data with Google Account Specialists

I accept the [Google Analytics terms](#)

Upon project creation, a new Google Analytics property will be created and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#)

Previous

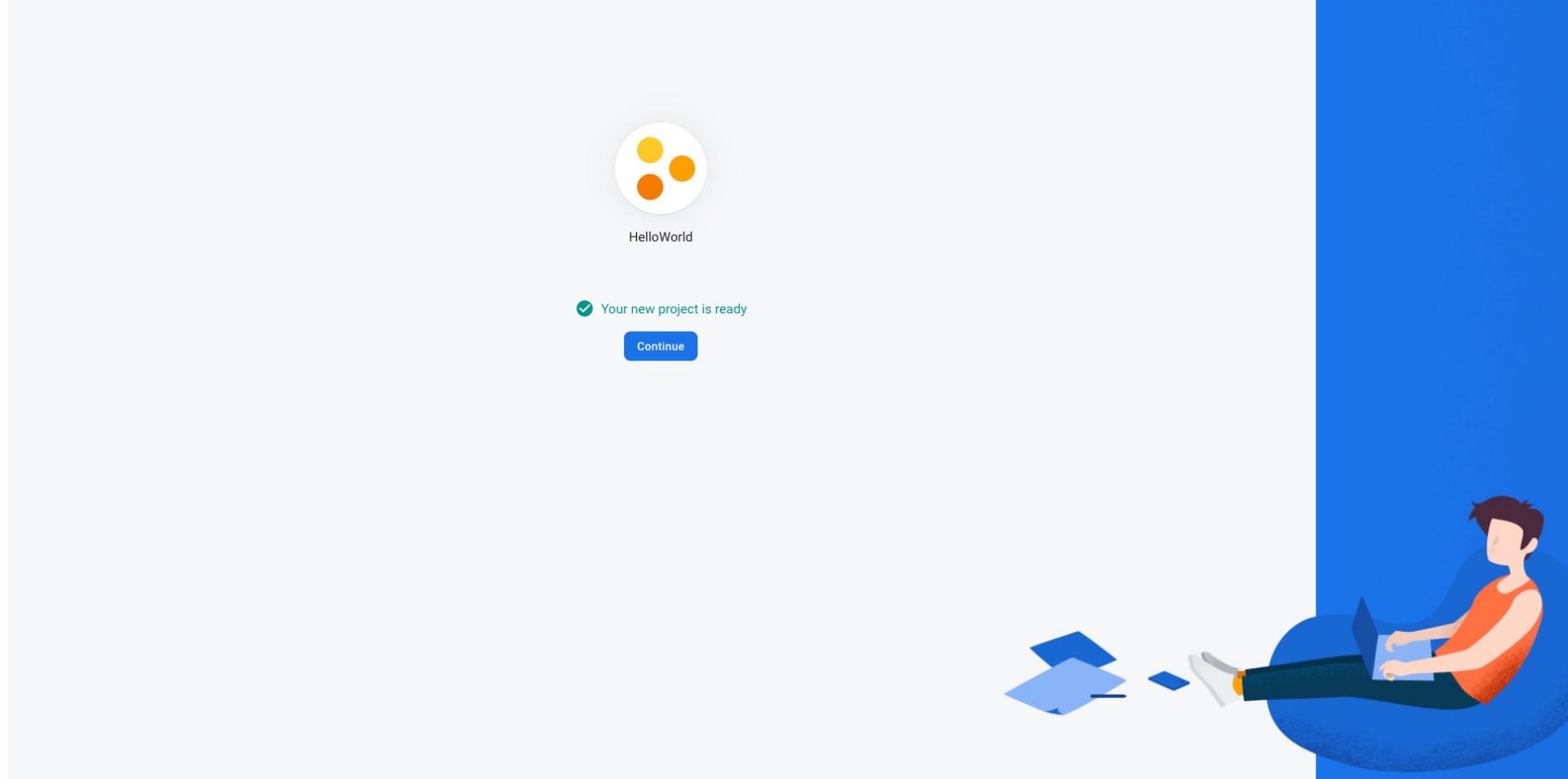
Create project

1. Chọn Quốc Gia

2. Chọn khởi tạo Project



Chọn continue để kết thúc quá trình khởi tạo Project





Dashboard của Project “HelloWorld” vừa khởi tạo

The screenshot shows the Firebase Project Overview dashboard for a project named "HelloWorld". The left sidebar contains navigation links for Project Overview, Build, Release & Monitor, Analytics, Engage, Extensions, and a Spark plan section indicating a free \$0/month usage. The main content area features a "Get started by adding Firebase to your app" section with icons for iOS, Android, Web, and Cloud Functions, along with a "Spark plan" button. It also includes a "Store and sync app data in milliseconds" section with cards for Authentication (purple background) and Cloud Firestore (orange background). A top banner offers email updates and a "Sign up" button.

Firebase

HelloWorld ▾

Project Overview

Build

Release & Monitor

Analytics

Engage

Extensions

Spark Free \$0/month Upgrade

Receive email updates about new Firebase features, research, and events

Sign up X

HelloWorld Spark plan

Get started by adding Firebase to your app

iOS Android </> Cloud Functions

Add an app to get started

Store and sync app data in milliseconds

Authentication

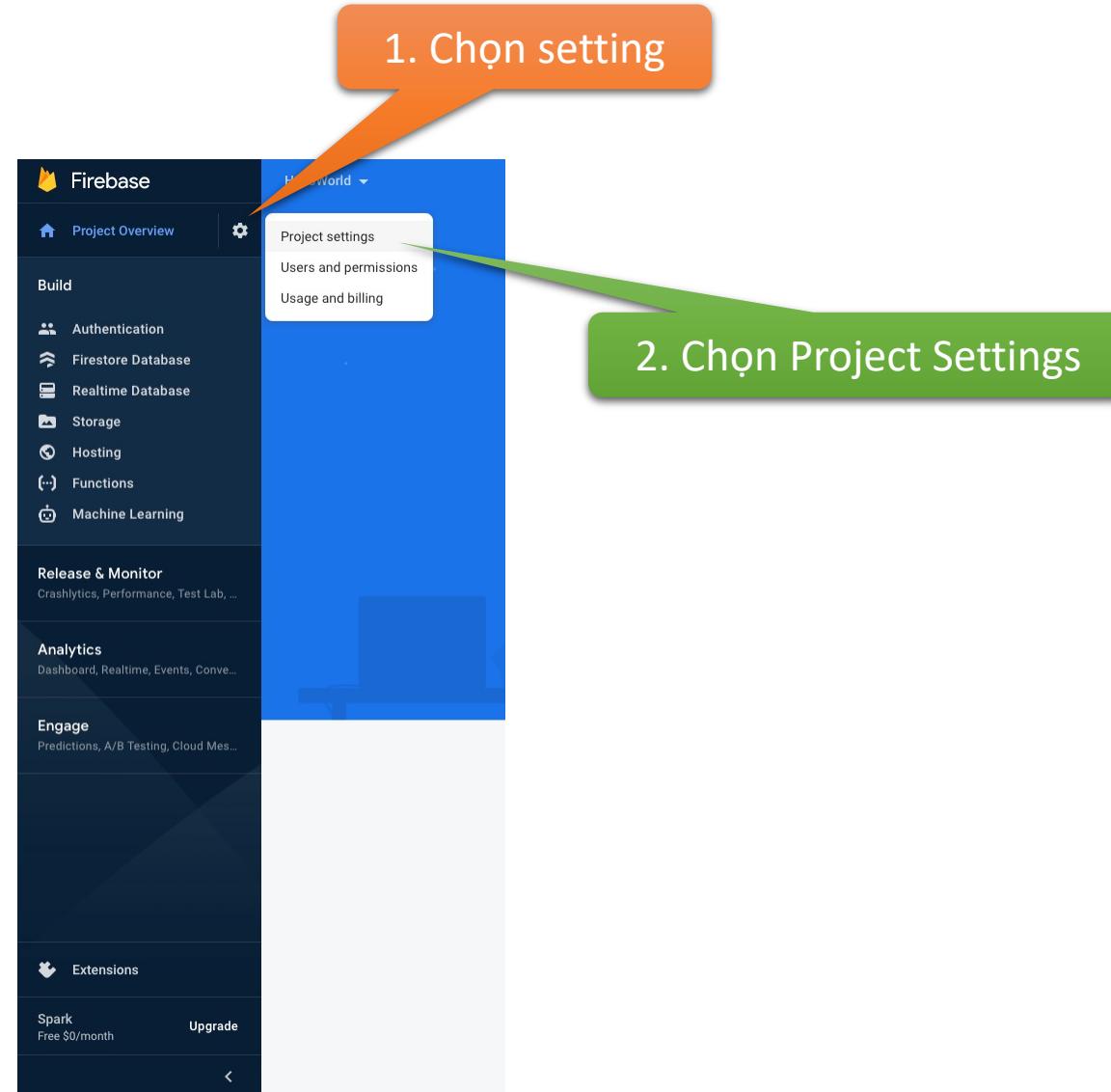
Authenticate and manage users

Cloud Firestore

Realtime updates, powerful queries, and automatic scaling



Lưu lại thông tin của Project





Lưu lại các thông tin của Project

The screenshot shows the Firebase Project settings page for a project named "HelloWorld". A yellow callout points to the "General" tab. An orange callout highlights the "Project ID" field, which contains "helloworld-620fb". A blue callout highlights the "Project Number" field, which contains "364134102513". A green callout highlights the "Web API Key" field, which contains "AlzaSyDVvsC16mvElj6oV6Cr_LHNm500C4ph0bl". The "General" tab is selected, and other tabs like "Cloud Messaging", "Integrations", "Service accounts", "Data privacy", "Users and permissions", and "App Check (BETA)" are visible.

Project settings

General Cloud Messaging Integrations Service accounts Data privacy Users and permissions App Check (BETA)

Your project

Project name: HelloWorld

Project ID: helloworld-620fb

Project number: 364134102513

Default GCP resource location: Not yet selected

Web API Key: AlzaSyDVvsC16mvElj6oV6Cr_LHNm500C4ph0bl

Public settings

These settings control instances of your project shown to the public

Public-facing name: project-364134102513

Support email: Not configured

Your apps

There are no apps in your project

Select a platform to get started

iOS Android </> Universal

Tab General

Project ID

Project Number

Web API Key



Lưu lại Server key

Tab Cloud Messaging

Lưu lại Server Key

Project settings

General Cloud Messaging Integrations Service accounts Data privacy Users and permissions App Check BETA

Project credentials

Key	Token
Server key	AAAAAVMgVdfE:APA91bFp5e9A6x775DMXQHr0psJ3m5miwCNY6qj6BLRGiumt4BN4G7VglmgqiCB71N-JhJVayqU95ySTDkaD8EQPBwZQkuaiDR17q-x_pNtSDF0b-VnHRanYcB2uwuOWG03BeVaPzKd

Sender ID ⓘ

364134102513

iOS app configuration

Web configuration

Web Push certificates	Web Push certificates
	Firebase Cloud Messaging can use Application Identity key pairs to connect with external push services. Learn more
	Key pair Date added
	Generate key pair

You can also [import an existing key pair](#)



Cấu hình Service Account

Tab Cloud Service Accounts

The screenshot shows the Firebase Project settings interface for a project named 'HelloWorld'. The 'Service accounts' tab is highlighted with a blue underline. On the right, there's a section titled 'Firebase Admin SDK' which includes a 'Legacy credentials' section with a 'Database secrets' option, and a 'All service accounts' section showing '3 service accounts'. Below this is an 'Admin SDK configuration snippet' for Node.js, Java, Python, and Go. A blue button at the bottom right of this section says 'Generate new private key'. The entire screenshot is framed by a large yellow border.

Manage service account permissions [\[\]](#)

Firebase Admin SDK

Legacy credentials

Database secrets

All service accounts

3 service accounts [\[\]](#)

Firebase Admin SDK

Your Firebase service account can be used to authenticate multiple Firebase features, such as Database, Storage and Auth, programmatically via the unified Admin SDK. [Learn more \[\]](#)

Firebase service account
firebase-adminsdk-qtij8@helloworld-620fb.iam.gserviceaccount.com

Admin SDK configuration snippet

Node.js Java Python Go

```
var admin = require("firebase-admin");
var serviceAccount = require("path/to/serviceAccountKey.json");

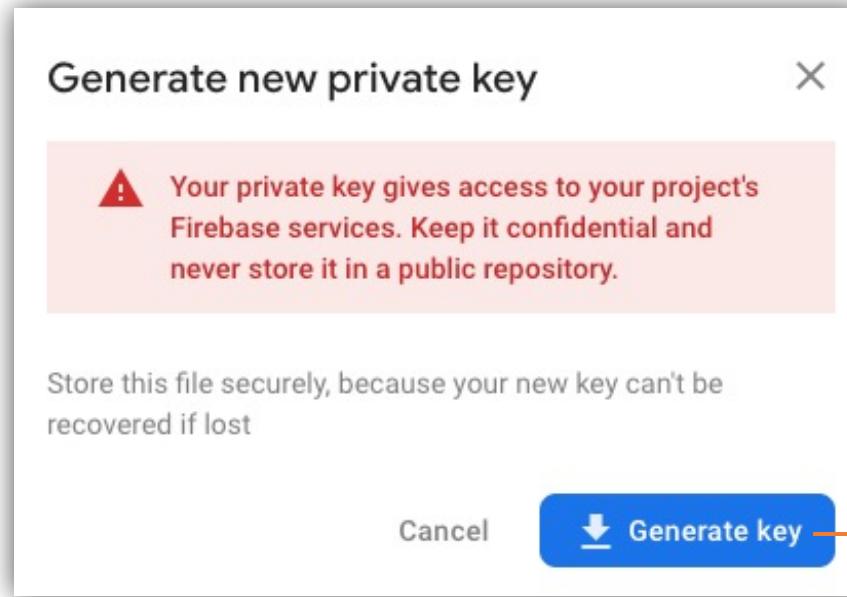
admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: "https://helloworld-620fb-default-rtdb.firebaseio.com"
});
```

Generate new private key

Chọn vào đây để tạo key



Download file Private Key



Download file Private Key.json





Khởi tạo Firebase Realtime Database

The screenshot shows the Firebase console interface for creating a Realtime Database. A yellow arrow points from the left towards the 'Realtime Database' section in the sidebar, with the text '1. Chọn Realtime Database'. Another orange arrow points from the right towards the 'Create Database' button, with the text '2. Chọn Create Database'.

1. Chọn Realtime Database

2. Chọn Create Database

Realtime Database
Store and sync data in real time

Create Database

Is Realtime Database right for you? Compare Databases

Introducing Firebase Realtime Database

Watch later Share

Watch on YouTube

How do I get started? View the docs

How much will Realtime Database cost? View pricing

Project Overview

Build

- Authentication
- Firebase Database
- Realtime Database**
- Storage
- Hosting
- Functions
- Machine Learning

Release & Monitor

Analytics

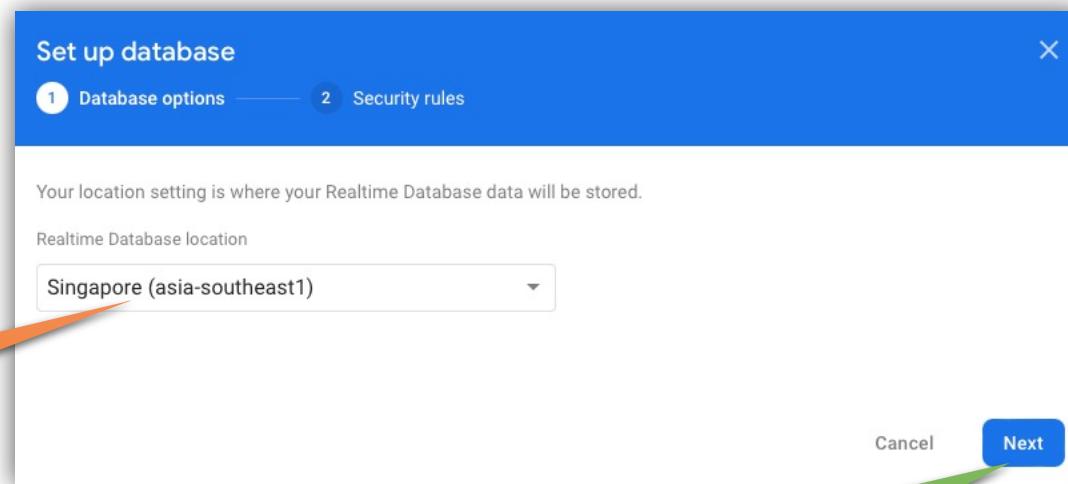
Engage

Extensions

Spark Free \$0/month Upgrade



Setup Database

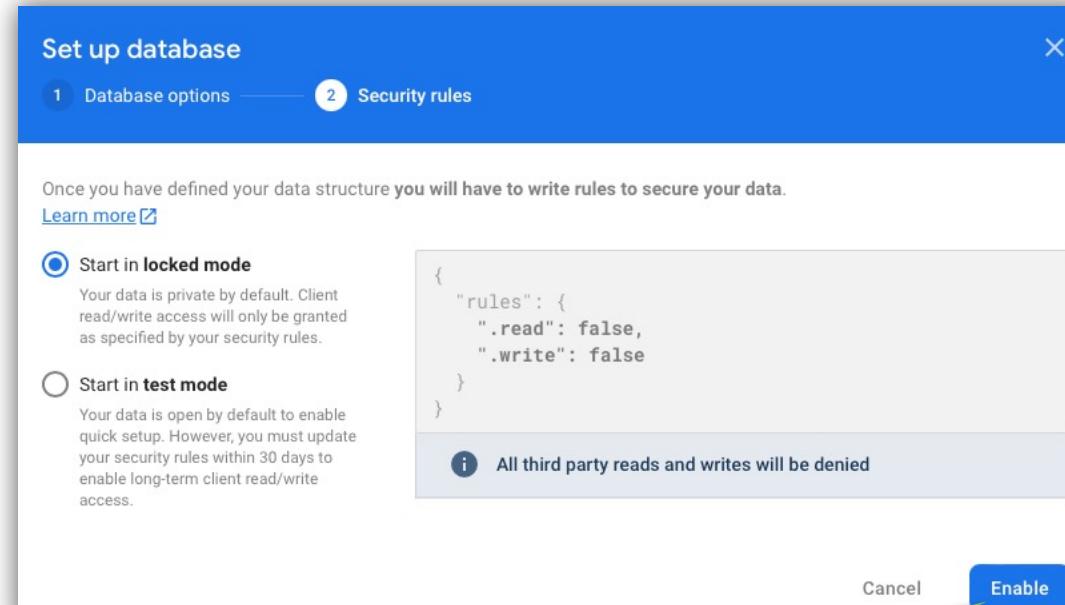


Nên chọn location là Singapore vì
sẽ gần về khoản cách địa lý

Chọn Next để chuyển sang bước kế tiếp



Setup Database



Chọn Enable để kích hoạt Database



Màn hình quản lý dữ liệu của Database

A screenshot of the Firebase Realtime Database console for a project named "HelloWorld". The left sidebar shows various services: Build, Authentication, Firestore Database, Realtime Database (selected), Storage, Hosting, Functions, Machine Learning, Release & Monitor, Analytics, Engage, and Extensions. The main area is titled "Realtime Database" and has tabs for Data, Rules, Backups, and Usage. A single node is listed under "Data": "helloworld-620fb-default-rtdb: null". Below the node, there is a URL: "https://helloworld-620fb-default.firebaseio.com/.json" and a note: "Database location: Singapore (asia-southeast1)". A green callout bubble with the text "Chú ý lưu lại thông tin database ID" (Note: Save the database ID) points to the node ID "helloworld-620fb-default-rtdb".



Cấu hình cho phép được ghi dữ liệu trên Database từ Client App

Chọn tab Rules

Thay đổi giá trị của read và write

```
1 {
2   "rules": {
3     ".read": false,
4     ".write": false
5   }
6 }
```

Database location: Singapore (asia-southeast1)



Update Rules

Chọn public để lưu lại cài đặt

Giá trị mới

```
1 ▾ {  
2 ▾   "rules": {  
3       ".read": "now < 1619283600000",  
4       ".write": "now < 1619283600000"  
5     }  
6 }
```

Database location: Singapore (asia-southeast1)



Tạo dữ liệu cho Database

https://helloworld-620fb-default-rtbd.firebaseio.southeast1.firebaseio.app/

helloworld-620fb-default-rtbd: null

Name products + ×

- Name 1 + ×
 - Name name Value iPhone ×
 - Name price Value 1000 ×
- Name 2 + ×
 - Name name Value iPad ×
 - Name price Value 1200 ×
- Name 3 + ×
 - Name name Value macBook ×
 - Name price Value 1500 ×

Cancel Add

Database location: Singapore (asia-southeast1)



https://helloworld-620fb-default-rtbd.firebaseio.southeast1.firebaseio.app/

helloworld-620fb-default-rtbd

products

- 1
 - name: "iPhone"
 - price: 1000
- 2
 - name: "iPad"
 - price: 1200
- 3
 - name: "macBook"
 - price: 1500

Database location: Singapore (asia-southeast1)



3. Kết nối NodeJS app với Firebase



Khởi tạo project NodeJS & ExpressJS

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows a project structure with a folder named "LAB06" containing "node_modules", "index.js", "package-lock.json", and "package.json". A file named "Private Key.json" is highlighted with an orange callout.
- Code Editor:** An "index.js" file is open, showing the following code:

```
const express = require("express");
const app = express();

app.use((req, res, next) => {
  console.log(` ${req.method} - ${req.originalUrl}`);
  next();
});

app.get("/", (req, res) => {
  res.end("Hello world");
});

app.listen(3000, () => {
  console.log("Server is starting at http://localhost:3000");
});
```
- Terminal:** Shows the command "npm start" being run, followed by the output: "Server is starting at http://localhost:3000" and "GET - /".
- Annotations:** An orange bracket on the right side of the slide points to the "app.use" line in the code editor, with the text "Cài đặt middleware để log thông tin của request".



Setup Firebase Module cho Project

- Cài đặt module firebase-admin

```
> npm install firebase-admin --save
```

- Sử dụng đối tượng admin từ module firebase-admin để xử lý trong mã nguồn của Project

```
const admin = require('firebase-admin');
```



Cấu hình firebase-admin object trong Project

```
JS index.js > ...
1  const express = require('express');
2  const app = express();
3
4  const bodyParser = require('body-parser');
5
6  const admin = require('firebase-admin');
7  const serviceAccount = require('./helloworld-620fb.firebaseio-adminsdk-qttj8-e3ba3a8599.json');
8
9  //? Khởi tạo app với service account
10 admin.initializeApp({
11   credential: admin.credential.cert(serviceAccount),
12   databaseURL: 'https://helloworld-620fb-default.firebaseio.firebaseio.com',
13   databaseAuthVariableOverride: {
14     uid: "my-service-worker"
15   }
16 });
17
18 //? Sử dụng quyền admin để app đọc và ghi dữ liệu trên database
19 var db = admin.database();
20 var ref = db.ref("/");
21
22 app.use((req, res, next) => {
23   console.log(` ${req.method} - ${req.originalUrl}`);
24   next();
25 });
26
27 app.use(bodyParser.json());
28 app.use(bodyParser.urlencoded({extended: true})); } Cấu hình bodyParser
29
```



Xây dựng API – Get – đọc danh sách Product từ Firebase

The diagram illustrates the flow of data from a Firebase database to a Node.js application and finally to a Postman API call.

Top Left: A screenshot of a browser window showing the Firebase Realtime Database at <https://helloworld-620fb-default.firebaseioapp.com/>. The database structure under 'products' shows three items: iPhone (name: "iPhone", price: 1000), iPad (name: "iPad", price: 1200), and MacBook (name: "macBook", price: 1500).

Middle: A code editor window displaying Node.js code for handling a GET request to '/products'. The code uses the Firebase Realtime Database SDK to read data once and return it as JSON.

```
34 app.get('/products', (req, res) => {
35   ref.once('value', (data) => {
36     res.writeHead(200, {'Content-Type': 'text/json'});
37     res.end(JSON.stringify(data.val()));
38   });
39 })
```

Bottom Right: A screenshot of the Postman application showing a GET request to 'localhost:3000/products'. The response status is 200 OK, and the response body is displayed in JSON format, matching the data structure shown in the Firebase database.

A screenshot of the Postman application interface. The request URL is 'localhost:3000/products' with a 'GET' method. The response status is 200 OK, and the response body is displayed in JSON format, showing the list of products from the Firebase database.

```
1 "products": [
2   "-MfuJhrIyyTM2_6gNUj5": {
3     "name": "iPhone",
4     "price": 1000
5   },
6   "-MfuJk4wrJhwXB61WRbA": {
7     "name": "iPad",
8     "price": 1200
9   },
10  "-MfuJm5DKdR089oiEuGc": {
11    "name": "macBook",
12    "price": 1500
13  }
14]
15}
16]
```



Xây dựng API – Thêm Object mới vào Firebase database

POST localhost:3000/products

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 ... "name": "iMac",  
2   "price": 2000  
3 ]
```

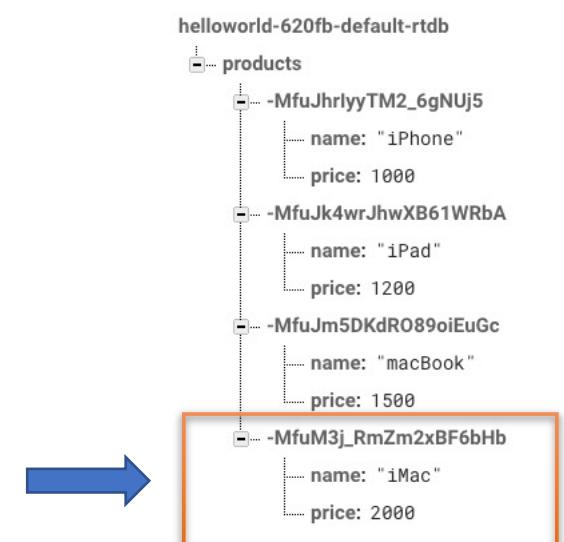
Body Cookies Headers (5) Test Results Status: 200 OK Time: 8 ms Size: 147 B Save Response

Pretty Raw Preview Visualize Text

1 OK



```
41 app.post('/products', (req, res) => {  
42   var name = req.body.name;  
43   var price = req.body.price;  
44  
45   var productRef = ref.child('products');  
46   productRef.push().set({  
47     name,  
48     price  
49   });  
50  
51   res.end("OK");  
52 });
```





Bài tập về nhà

- Xây dựng một ứng dụng Express thực hiện các yêu cầu sau
 - Tạo một Firebase Database quản các bài POST. Thông tin POST gồm (4 điểm)
 - Title – tiêu đề của bài Post
 - Content - chứa nội dung của bài Post
 - Tạo project Express để thực hiện các thao tác (6 điểm)
 - Hiển thị tất cả các bài Post
 - Thêm một bài Post mới
 - Xóa một bài Post

