

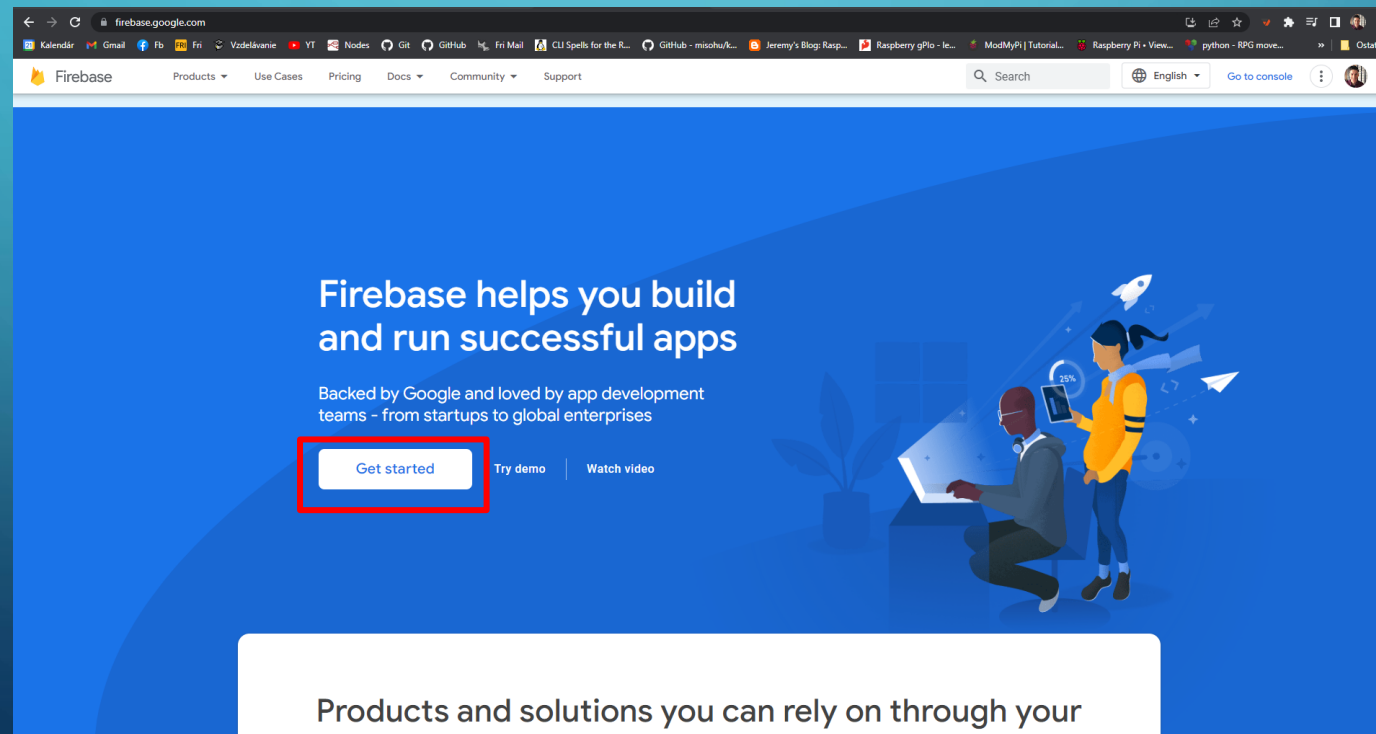
A decorative graphic on the left side of the slide consisting of white lines and circles on a blue gradient background, resembling a circuit board or a stylized tree structure.

GOOGLE FIREBASE

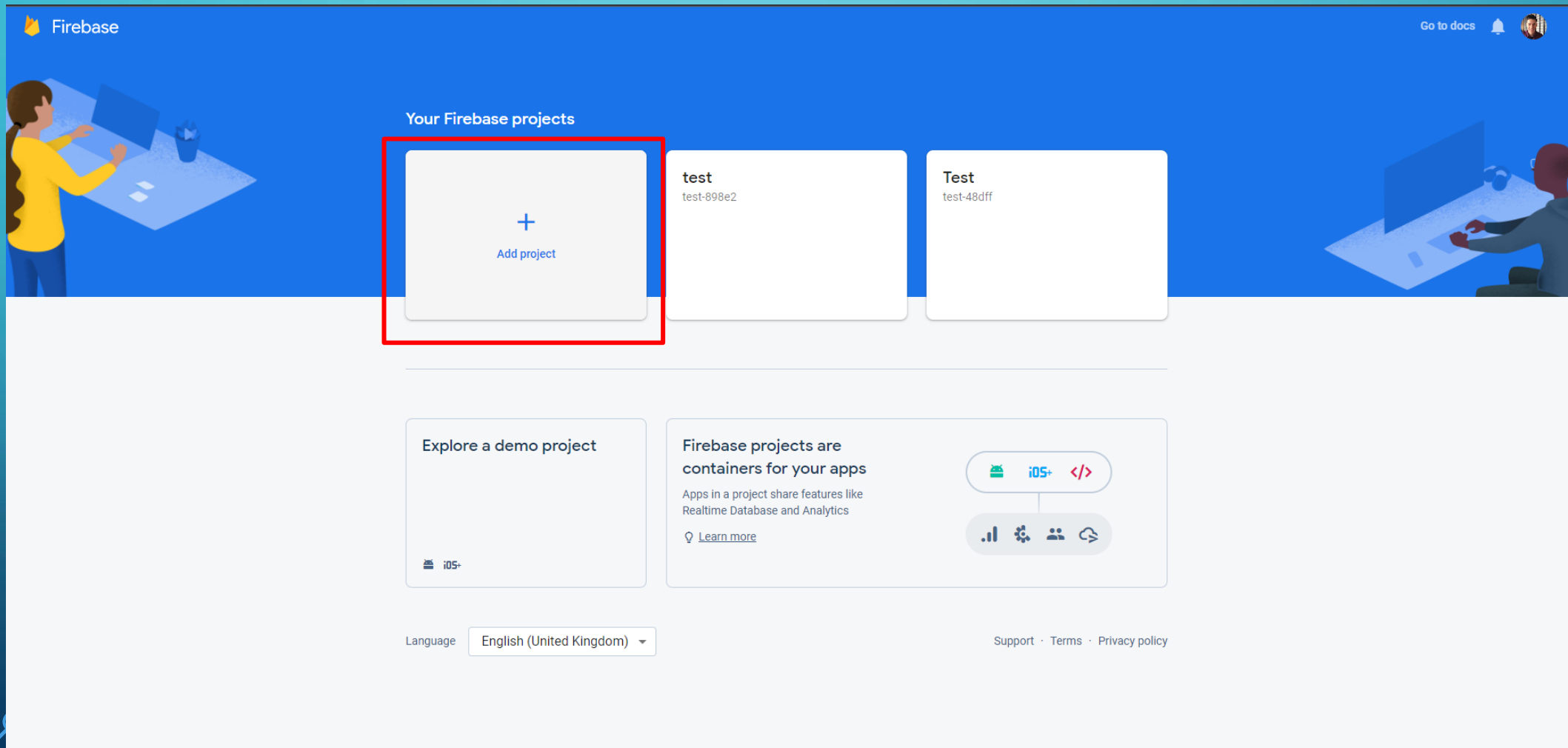
FRI UNIZA

GOOGLE FIREBASE

- <https://firebase.google.com/>
- Realtime NoSQL database.



PROJECT CREATION



The screenshot shows the Firebase Project Creation interface. At the top, the Firebase logo is on the left, and 'Go to docs' with a bell icon and a user profile icon are on the right. The main section is titled 'Your Firebase projects'. It contains three cards: a white card with a blue plus sign and the text 'Add project' (highlighted with a red border), a white card titled 'test' with ID 'test-898e2', and a white card titled 'Test' with ID 'test-48dff'. Below these cards, there are two promotional boxes. The left box is titled 'Explore a demo project' and features an 'iOS+' icon. The right box is titled 'Firebase projects are containers for your apps' and mentions 'Apps in a project share features like Realtime Database and Analytics', with a 'Learn more' link. To the right of this text are two rows of icons: the top row has Android, iOS+, and code symbols; the bottom row has analytics, settings, users, and share icons. At the bottom of the page, there is a language selector set to 'English (United Kingdom)' and a footer with links for 'Support', 'Terms', and 'Privacy policy'.

Firebase

Go to docs

Your Firebase projects

+
Add project

test
test-898e2

Test
test-48dff

Explore a demo project

iOS+

Firebase projects are containers for your apps

Apps in a project share features like Realtime Database and Analytics

[Learn more](#)

Language: English (United Kingdom)

Support · Terms · Privacy policy

PROJECT CREATION

× Create a project (Step 1 of 3)

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

Project name

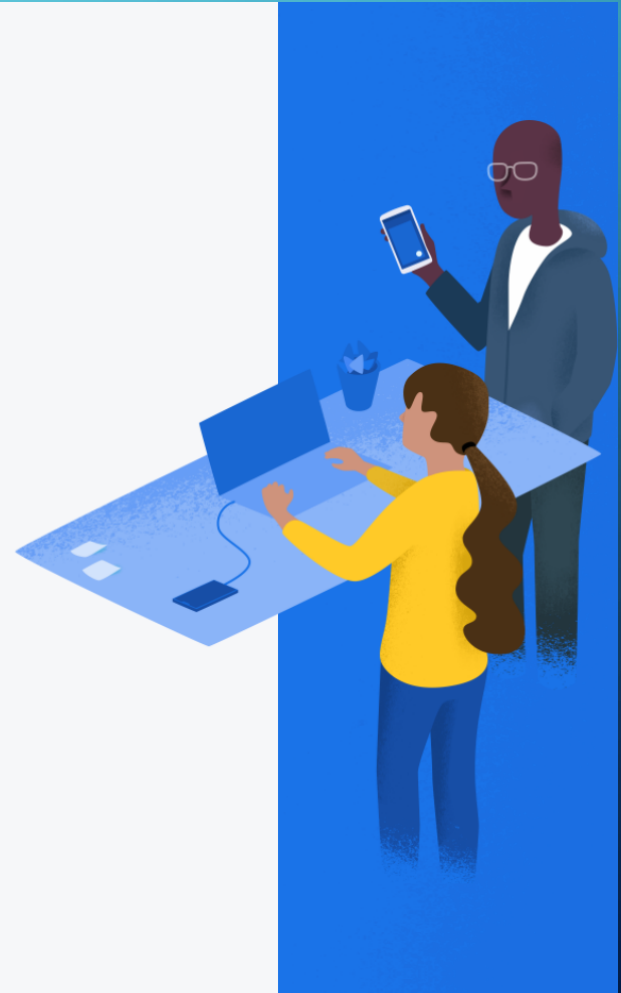
IoT-ESP32

✎ iot-esp32-5c3f0



I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft or profession.

Continue













PROJECT CREATION

× 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 and Cloud Functions.

Google Analytics enables:

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

☒ Enable Google Analytics for this project
Recommended

[Previous](#)

[Continue](#)



PROJECT CREATION

× Create a project(Step 3 of 3)

Configure Google Analytics

Choose or create a Google Analytics account ⓘ

Default Account for Firebase

Automatically create a new property in this account ✎

Upon project creation, a new Google Analytics property will be created in your chosen Google Analytics account 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



DATABASE CREATION

The screenshot displays the Firebase console interface. On the left, a dark sidebar contains a menu with categories: 'Project Overview', 'Build', 'Release and monitor', 'Analytics', 'Engage', and 'Extensions'. The 'Build' section is expanded, showing 'Authentication', 'Firestore Database', 'Realtime Database' (highlighted with a red rectangle), 'Storage', 'Hosting', 'Functions', and 'Machine Learning'. The main content area has a dark header with the 'Realtime Database' title and the subtitle 'Store and sync data in real time'. A 'Create Database' button is highlighted with a red rectangle. To the right of the button is an illustration of three server racks. Below the header, a white banner asks 'Is Realtime Database right for you?' with a 'Compare Databases' link. Further down, a 'Learn more' section contains two cards: 'How do I get started?' with a 'View the docs' link, and 'How much will Realtime Database cost?' with a 'View pricing' link. On the right side of the 'Learn more' section is a video player titled 'Introducing Firebase Realtime Database' with a play button icon and a 'Pozriť nes...' (Watch later) button.

Firebase

IoT-ESP32

Project Overview

Build

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

Release and monitor

Analytics

Engage

Extensions

Spark

No cost \$0/month

Upgrade

Realtime Database

Store and sync data in real time

Create Database

Go to docs

Is Realtime Database right for you? [Compare Databases](#)

Learn more

- How do I get started?
[View the docs](#)
- How much will Realtime Database cost?
[View pricing](#)

Introducing Firebase Realtime Database

Pozriť nes... Zdieľať

Prehľad na YouTube

DATABASE CREATION

The screenshot shows the Firebase console interface for creating a Realtime Database. The left sidebar contains navigation links for Project Overview, Build (Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning), Release and monitor, Analytics, Engage, and Extensions. The main content area is titled 'Realtime Database' with the subtitle 'Store and sync data in real time'. A 'Create Database' button is visible. A modal dialog titled 'Set up database' is open, showing two steps: '1 Database options' and '2 Security rules'. The 'Database options' step is active, displaying the text 'Your location setting is where your Realtime Database data will be stored.' Below this, a dropdown menu for 'Realtime Database location' is set to 'United States (us-central1)'. At the bottom of the modal, there are 'Cancel' and 'Next' buttons. The background of the console shows a server rack illustration and some project details like 'IoT-ESP32'.

Firebase

IoT-ESP32

Project Overview

Build

- Authentication
- Firestore Database
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Release and monitor

Crashlytics, Performance, Test Lab ...

Analytics

Dashboard, Realtime, Events, Conve...

Engage

A/B Testing, Cloud Messaging, In-A...

Extensions

Spark

No cost \$0/month Upgrade

Realtime Database

Store and sync data in real time

Create Database

Set up database

1 Database options 2 Security rules

Your location setting is where your Realtime Database data will be stored.

Realtime Database location

United States (us-central1)

Cancel Next

How do I get started?
View the docs

How much will Realtime Database cost?
View pricing

Pozrief nes... Zdieľat

DATABASE CREATION

The screenshot shows the Firebase console interface for creating a Realtime Database. A modal dialog titled "Set up database" is open, with two tabs: "1 Database options" and "2 Security rules". The "Database options" tab is active, showing two radio button options: "Start in locked mode" (selected) and "Start in test mode". The "Start in locked mode" option is highlighted with a red box. Below the options, there is a code block showing the default security rules for locked mode:

```
{  "rules": {    ".read": false,    ".write": false  }}
```

. An information icon and text state: "All third party reads and writes will be denied". At the bottom of the dialog, there are "Cancel" and "Enable" buttons, with the "Enable" button highlighted by a red box. The background shows the Firebase console sidebar with navigation links like "Project Overview", "Authentication", "Firestore Database", "Realtime Database", "Storage", "Hosting", "Functions", and "Machine Learning". The main content area behind the dialog says "Realtime Database" and "Store and sync data in real time".

Set up database

1 Database options — 2 Security rules

Once you have defined your data structure, you will have to write rules to secure your data.
[Learn more](#)

☒ **Start in locked mode**
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

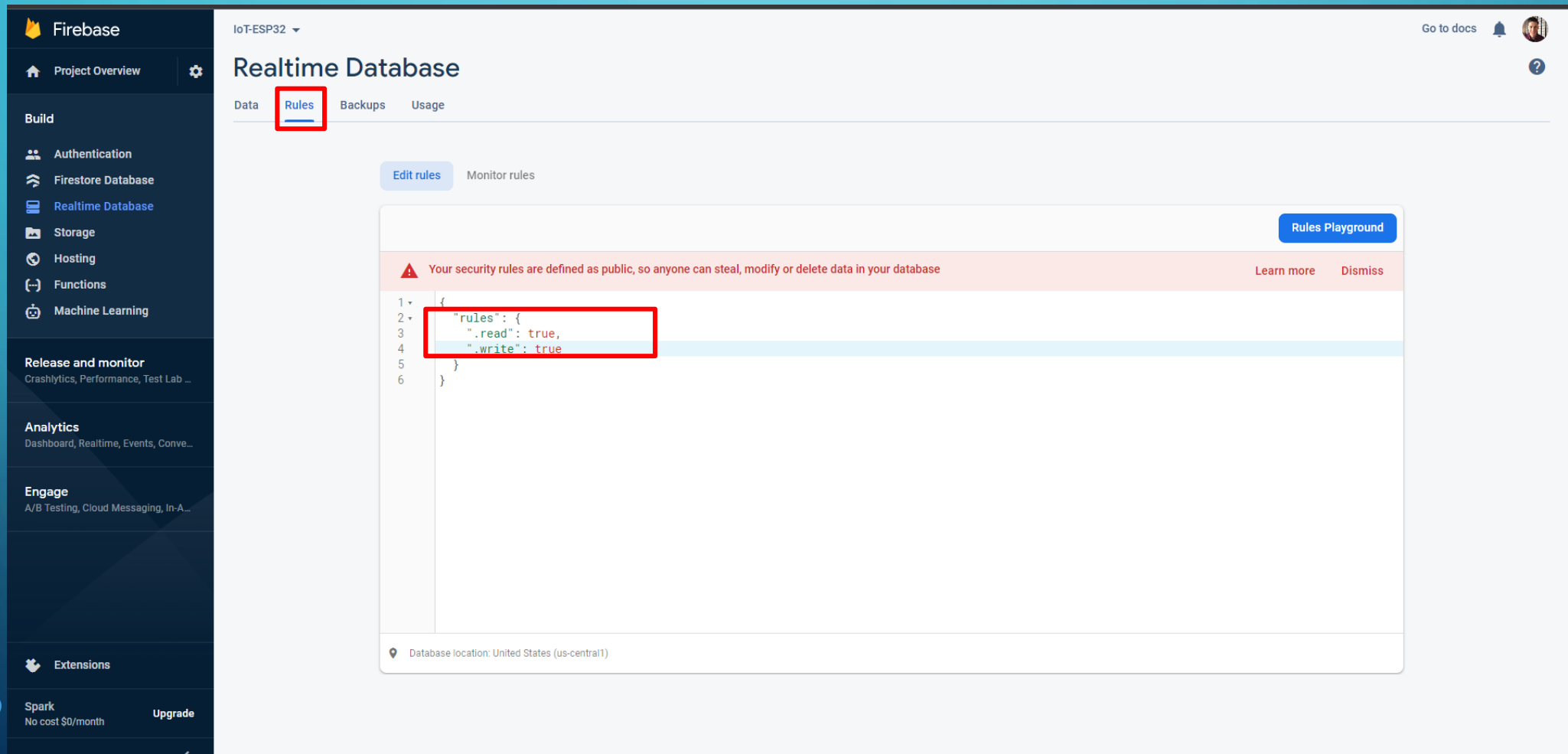
☐ **Start in test mode**
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
{  "rules": {    ".read": false,    ".write": false  }}
```

i All third party reads and writes will be denied

Cancel **Enable**

RULES



Firestore

Project Overview

Build

- Authentication
- Firestore Database
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Release and monitor

Crashlytics, Performance, Test Lab ...

Analytics

Dashboard, Realtime, Events, Conve...

Engage

A/B Testing, Cloud Messaging, In-A...

Extensions

Spark

No cost \$0/month Upgrade

IoT-ESP32

Realtime Database

Data **Rules** Backups Usage

Edit rules Monitor rules

Rules Playground

⚠ Your security rules are defined as public, so anyone can steal, modify or delete data in your database [Learn more](#) [Dismiss](#)

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

Database location: United States (us-central1)

URL & DATA

The screenshot displays the Firebase Realtime Database interface. On the left is a dark sidebar with the 'Firebase' logo and a menu including 'Project Overview', 'Build' (with sub-items like Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning), 'Release and monitor', 'Analytics', 'Engage', 'Extensions', and 'Spark'. The main area is titled 'Realtime Database' for project 'IoT-ESP32'. It has tabs for 'Data', 'Rules', 'Backups', and 'Usage'. A warning banner at the top states: 'Protect your Realtime Database resources from abuse, such as billing fraud or phishing. Configure App Check'. Below this, a red box highlights the 'Data' tab, which contains a red box around the URL 'https://iot-esp32-5c3f0-default-rtdb.firebaseio.com'. A red arrow labeled 'URL' points to this box. Below the URL is a red box around the data entry 'test: "1234"', with a red arrow labeled 'Data' pointing to it. A red box also encompasses the entire data entry area. At the bottom, it shows 'Database location: United States (us-central1)'.

URL

Data

MICROPYTHON-FIREBASE LIBRARY

- <https://github.com/ckoevery/micropython-firebase-realtime-database>
- Commands:
 - get,
 - getfile,
 - put,
 - patch,
 - addto,
 - delete.

Download : ufirebase.py and
save on device (esp32).

GUIDE

1. Upload library to ESP32.
2. Connect ESP32 to AP with internet connection.
3. Import library: `import ufirebase as firebase.`
4. Set URL : `firebase.setURL(„URL“).`
5. Data upload: `firebase.put(„test“, „1 234“, bg=False).`
`firebase.addto („vals“, {„temp“:20, „hum“:50}, bg=False).`
6. Data download: `firebase.get(„test“, „testVariable“, bg=False).`
`print(firebase.testVariable).`

Optional run in the background with the keyword `bg`.

TASKS

1. Create your own firebase database.
2. Upload measured values (temperature & humidity & photoresistor) to database in periodic interval.
3. Upload measured values to database when button is pressed.
4. Make pairs. If the button on the first board is pressed, the state of the led on the second board will change. Use a database for synchronization.