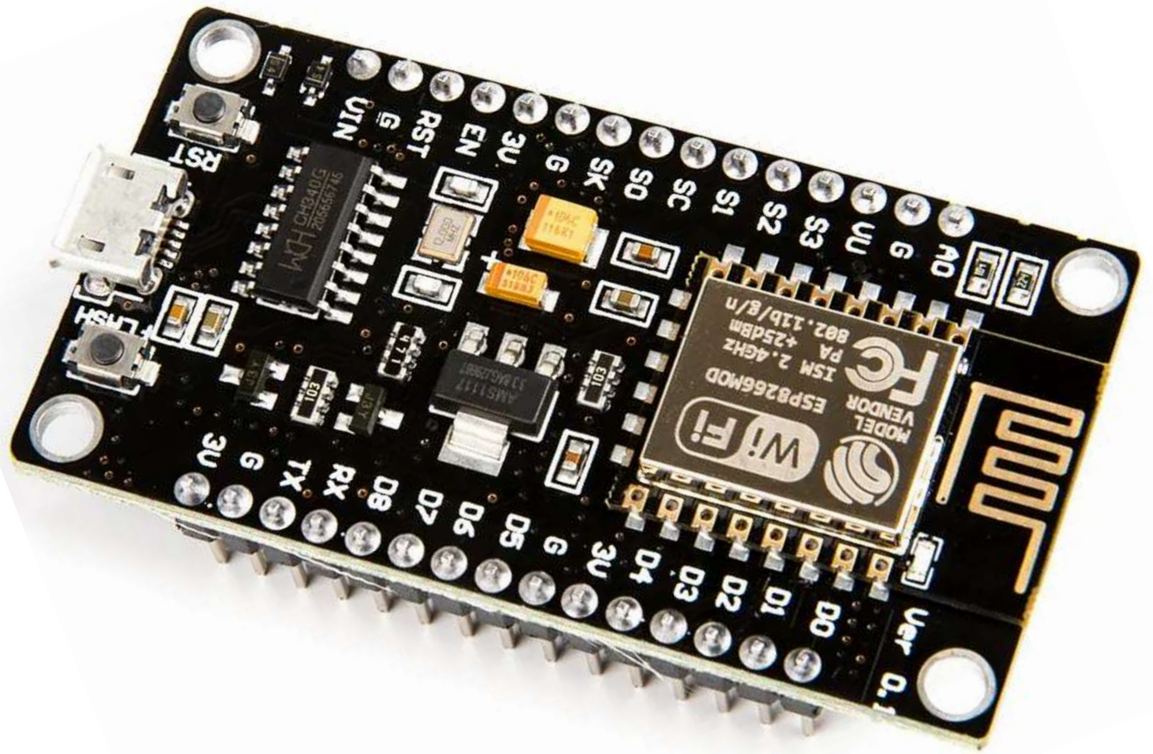
Internet of Things

NodeMCU: Connecting to the Internet

IoT Team, BFCAI

NodeMCU ESP8266

- NodeMCU is a low-cost open-source IoT platform based on the ESP8266 Wi-Fi system on a chip.
- NodeMCU Version 3 runs on the ESP-12E (ESP8266MOD) module.



IoT Application: Smart Umbrella

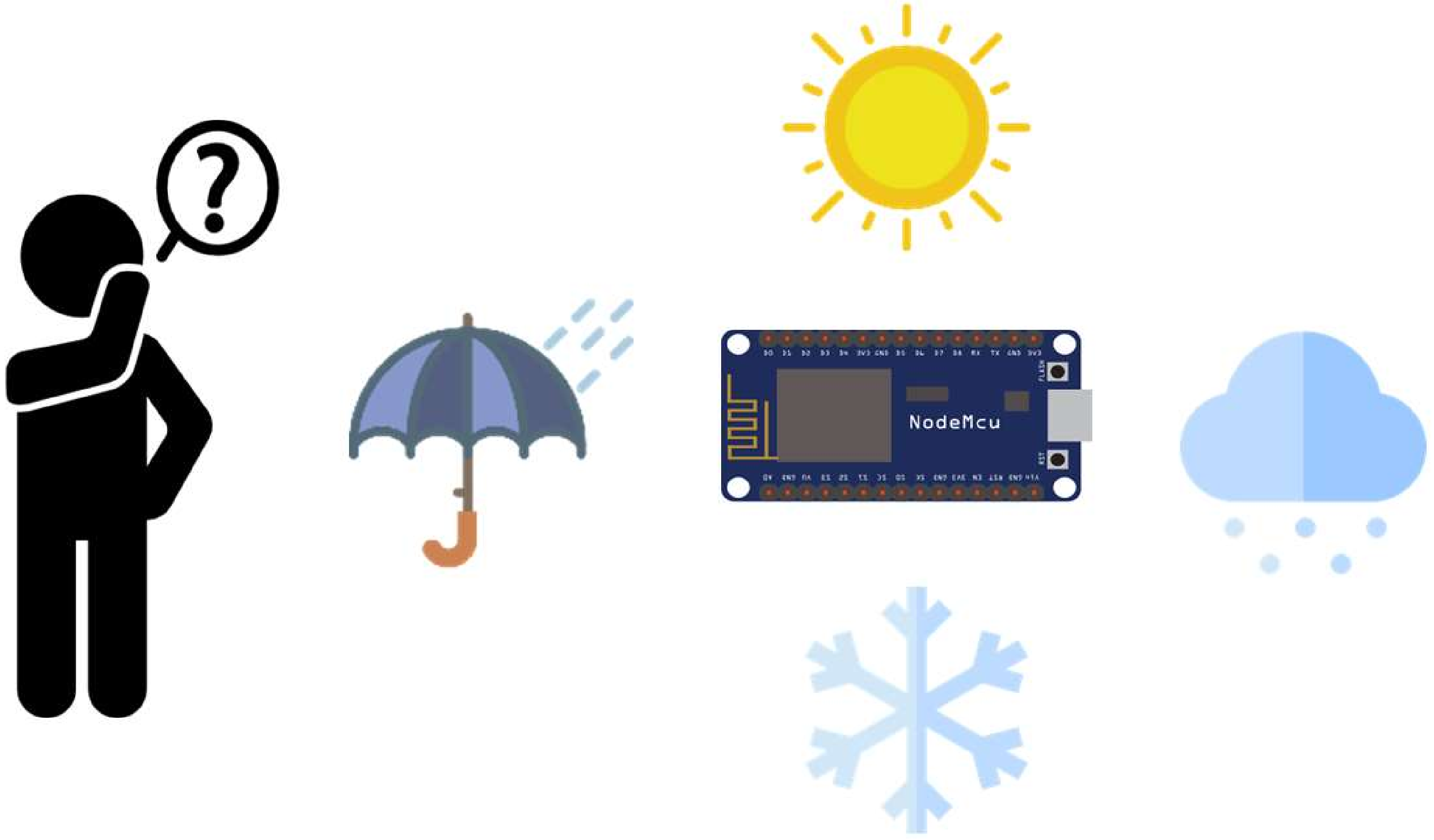
- An umbrella that provides information about the likelihood of rain so that users can make a simple decision about whether to take the umbrella with them as they leave their home.

IoT Application: Smart Umbrella

- The umbrella has a handle that would illuminate when snow or rain was in the forecast.

IoT Application: Smart Umbrella

- Using existing Wi-Fi technology to pull information about the weather from the Internet.

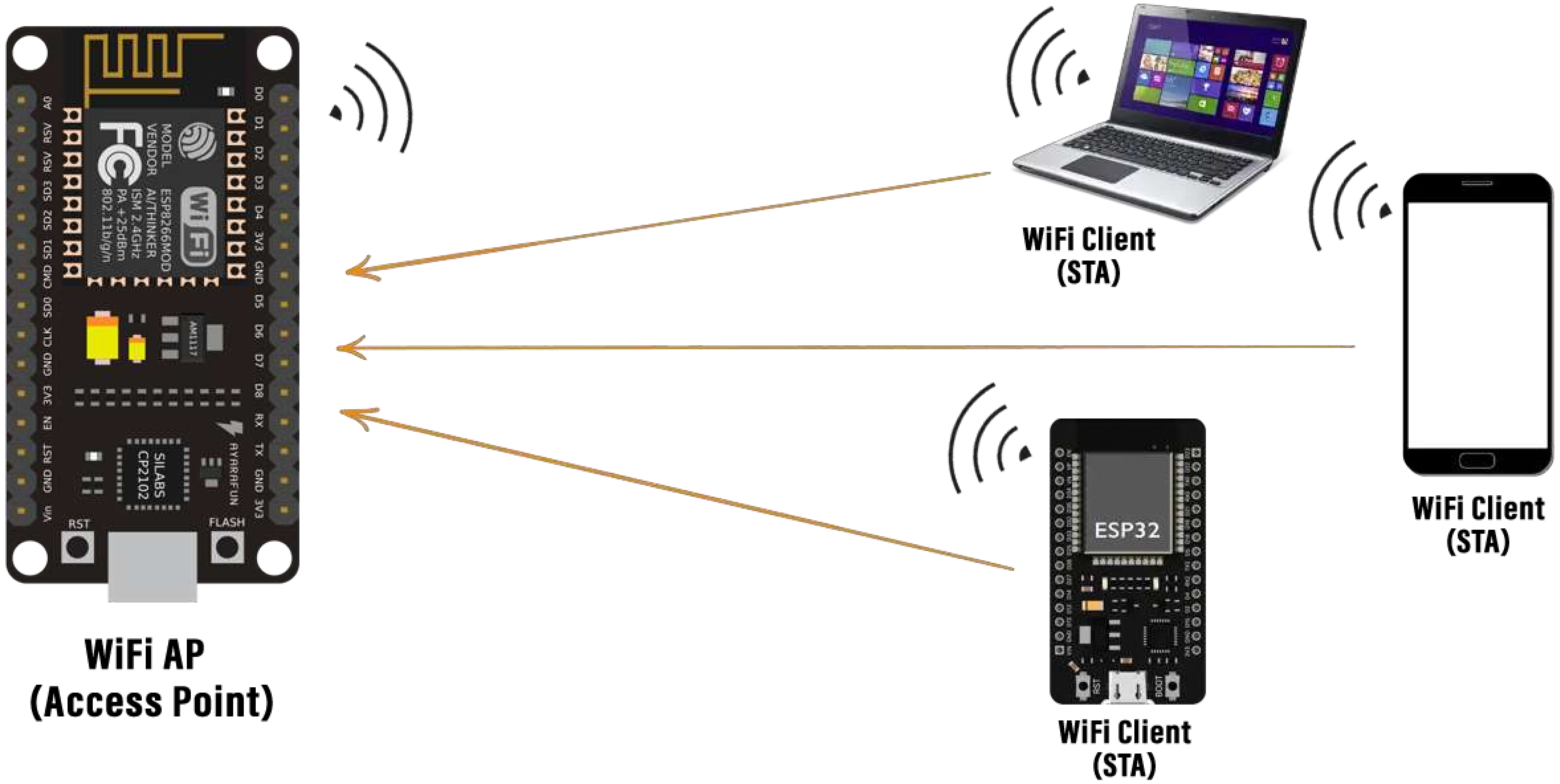


NodeMCU ESP8266: WiFi Modes

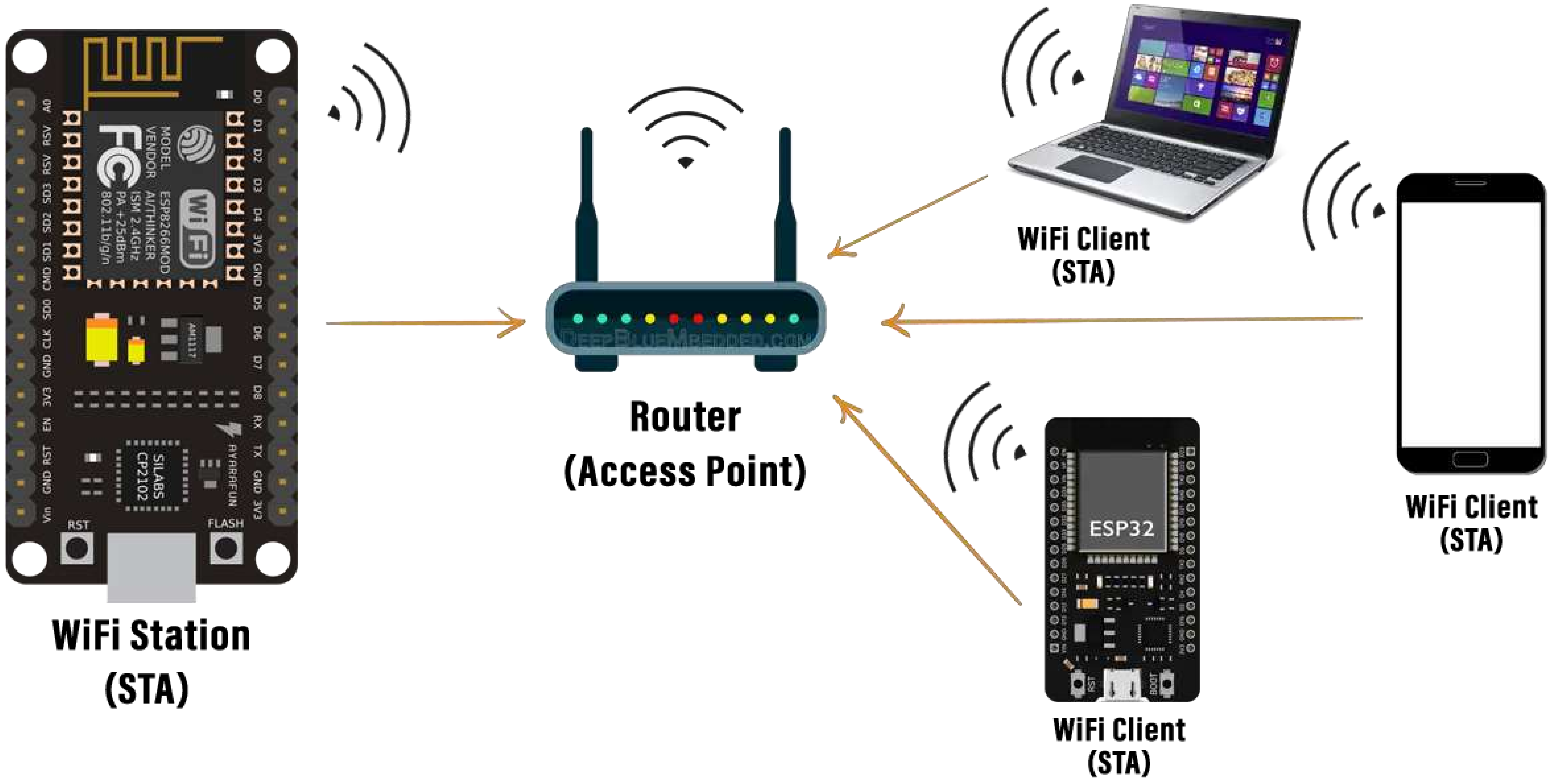
- The NodeMCU WiFi can run in three modes: WiFi Station, Access Point, or both at the same time.
- To set the NodeMCU WiFi mode, you can use the `WiFi.mode()` function.

WiFi Mode	**Description**
`WiFi.mode(WIFI_STA);`	**Station Mode (STA)** NodeMCU connects to other networks
`WiFi.mode(WIFI_AP);`	**Access Point Mode (AP)** NodeMCU creates its own network, and other WiFi stations can connect to it
`WiFi.mode(WIFI_AP_STA);`	**Access Point + Station Mode (AP_STA)** NodeMCU WiFi will act as both Access Point and WiFi Station at the same time

NodeMCU ESP8266: Access Point Mode



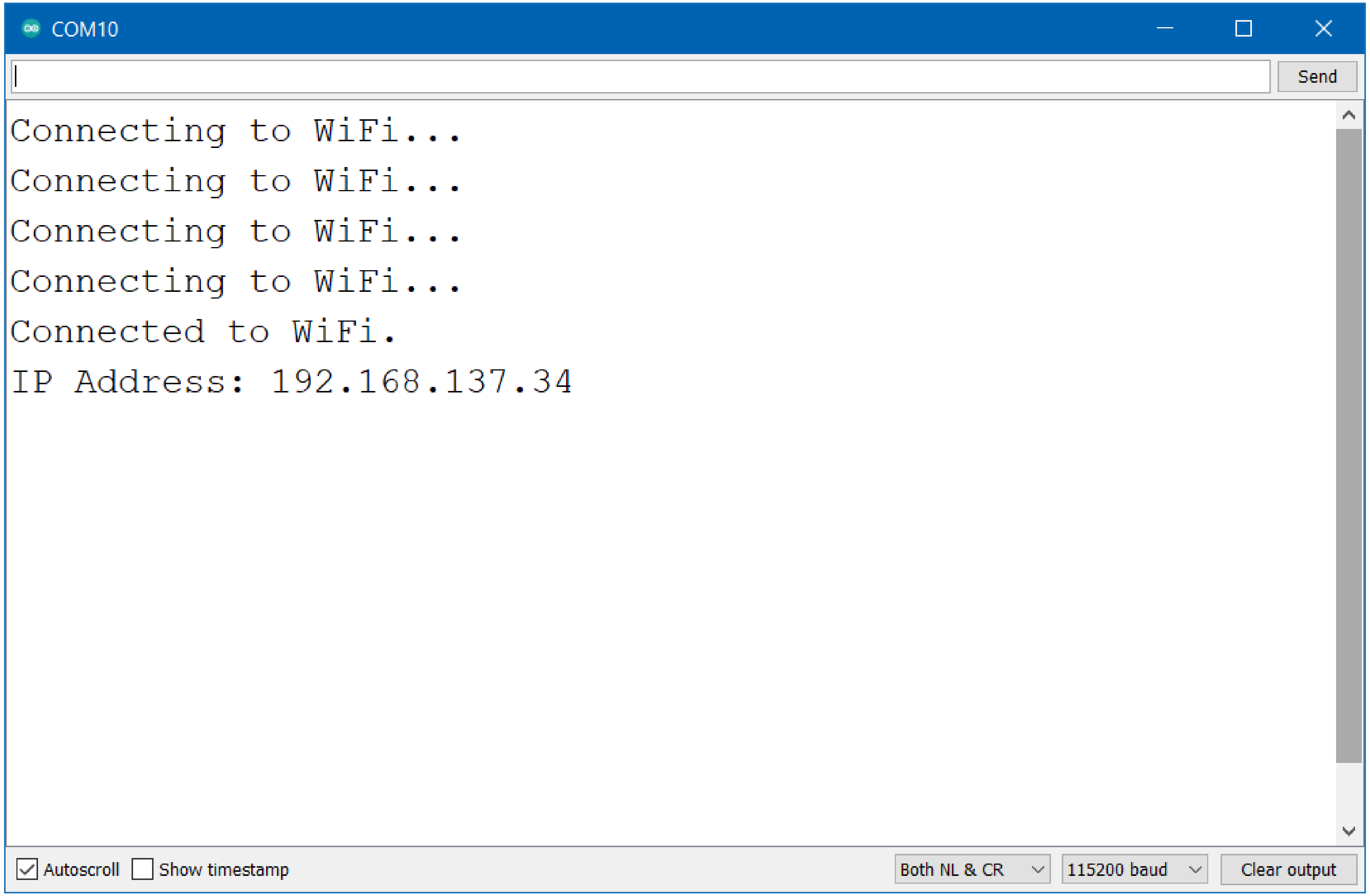
NodeMCU ESP8266: WiFi Station Mode



NodeMCU ESP8266: WiFi Station Mode – Code

```
#include <ESP8266WiFi.h>                          // Include ESP8266WiFi library for WiFi features

const char* WIFI_SSID = "iotlab";                 // Define the WiFi network SSID
const char* WIFI_PASS = "hostiotlab";             // Define the WiFi network password

void setup() {
  Serial.begin(115200);                           // Start serial communication at 115200 baudrate
  WiFi.begin(WIFI_SSID, WIFI_PASS);               // Begin WiFi connection using SSID and password

  while(WiFi.status() != WL_CONNECTED){           // Check if the WiFi status is not connected
    delay(1000);                                  // Wait until the WiFi connection is established
    Serial.println("Connecting to WiFi...");      // Print message indicating an attempt to connect to WiFi
  }

  Serial.println("Connected to WiFi.");           // Print message when WiFi connection is successful
  Serial.print("IP Address: ");                   // Print the label for the IP address
  Serial.println(WiFi.localIP());                 // Print the assigned IP address
}

void loop() {

}
```

NodeMCU ESP8266: WiFi Station Mode – Output



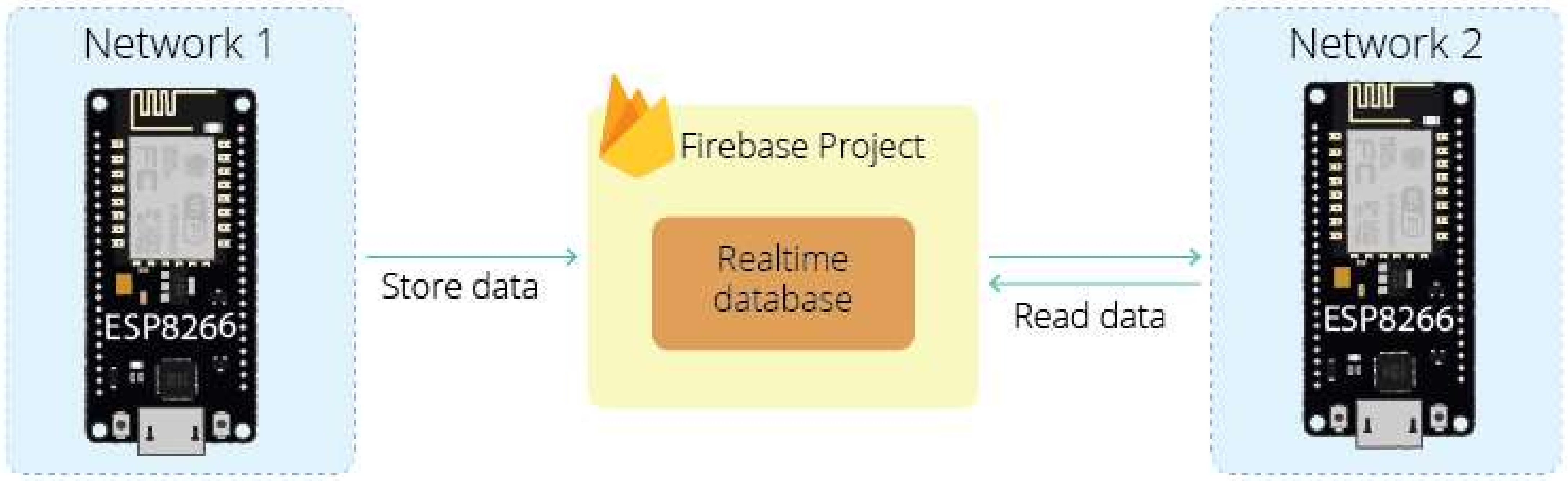
```
Connecting to WiFi...
Connecting to WiFi...
Connecting to WiFi...
Connecting to WiFi...
Connected to WiFi.
IP Address: 192.168.137.34
```

Firebase

- Firebase is Google’s mobile and web application development platform that includes many services to manage data from IOS, Android, or web applications.
- This includes things like analytics, authentication, databases, configuration, file storage, push messaging, and the list goes on.
- The services are hosted in the cloud and scale with little to no effort on the part of the developer.
- You can use the ESP8266 to connect and interact with your Firebase project, and you can create applications to control the ESP8266 via Firebase from anywhere in the world.

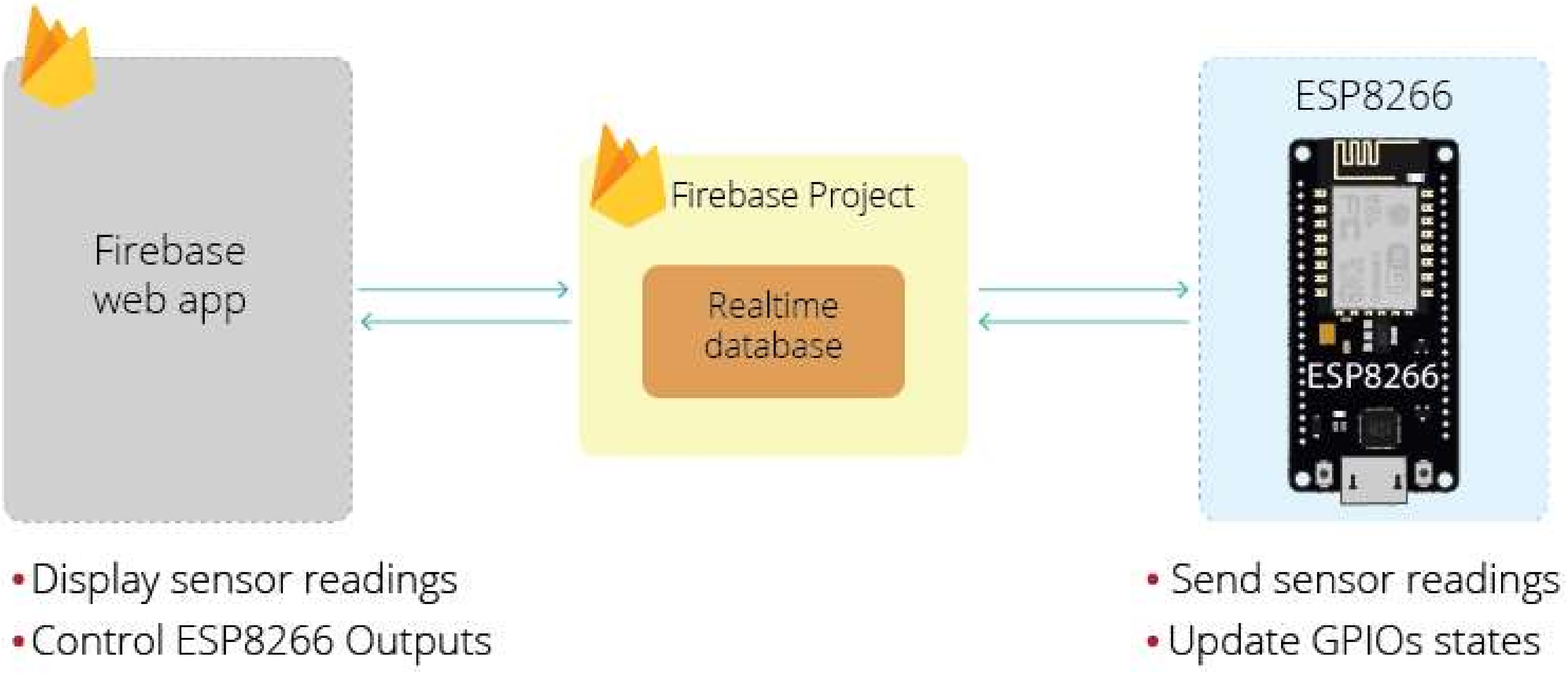
Firebase

- The ESP8266 can interact with the database from anywhere in the world.
- You can have two ESP8266 boards in different networks, with one board storing data and the other board reading the most recent data, for example.



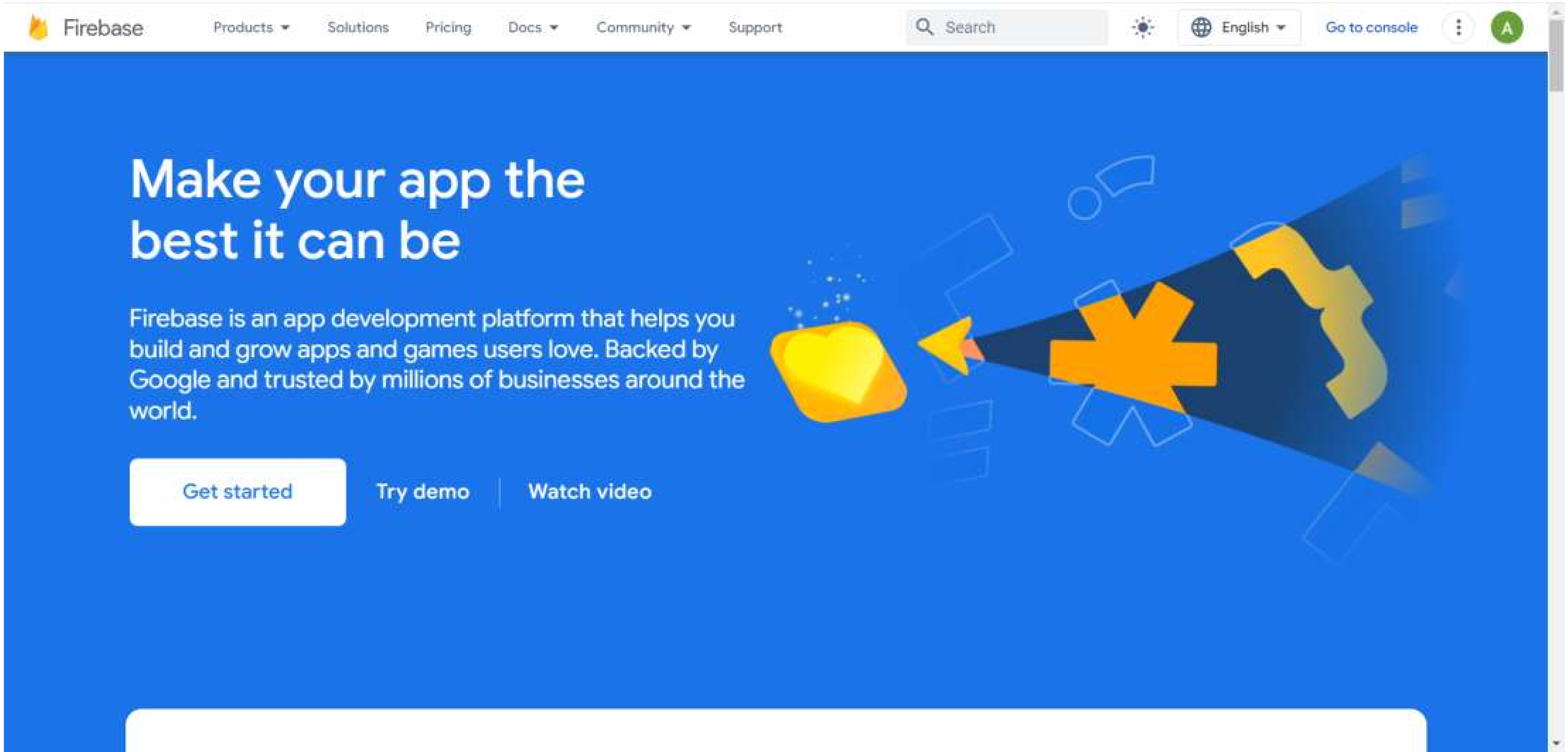
Firebase

- You can have a web or mobile app using Firebase that will use ESP8266 to display sensor readings or control outputs from anywhere in the world.



Firebase: Creating a New Project

- Go to https://firebase.google.com

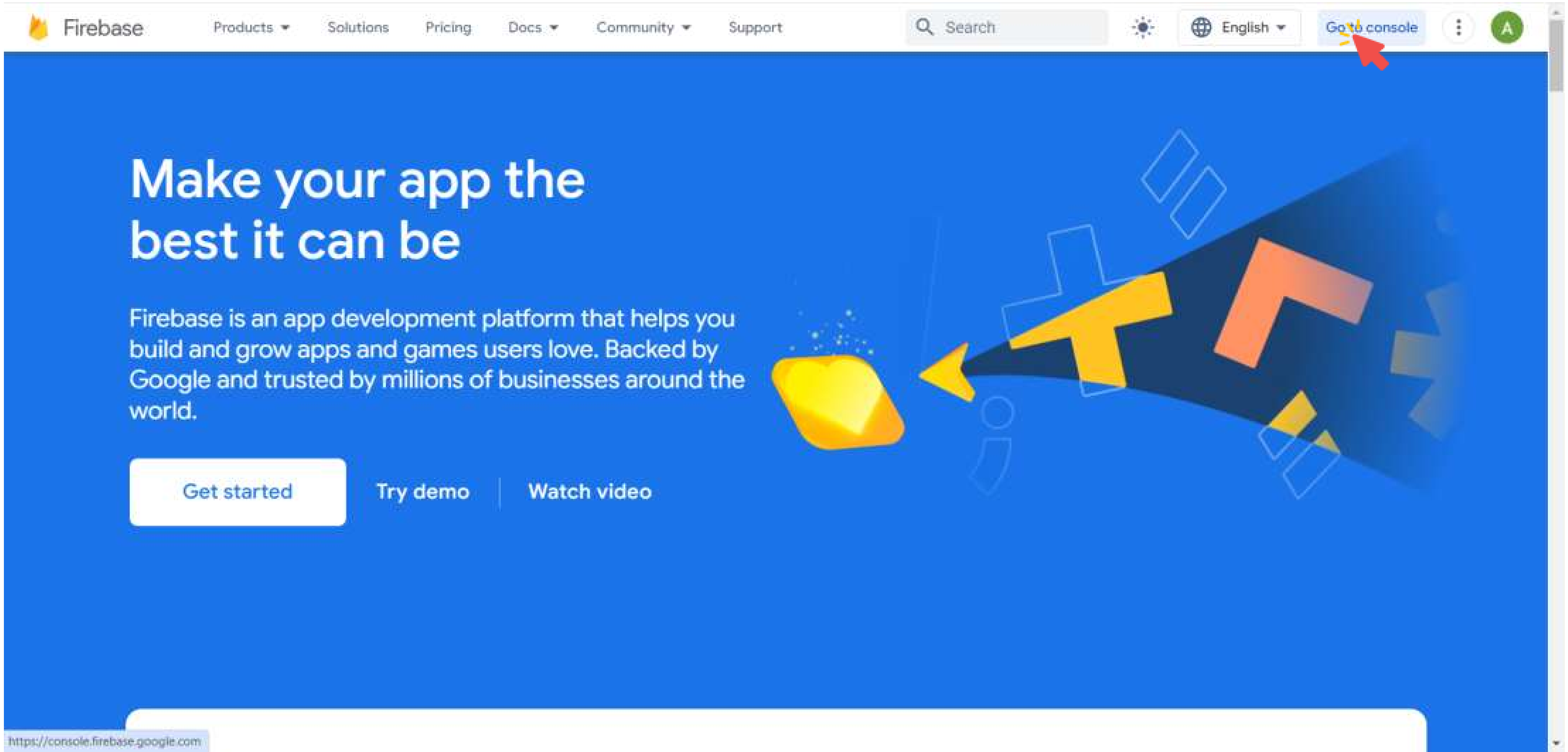


Make your app the best it can be

Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

Firebase: Creating a New Project

- Click on Go to Console.

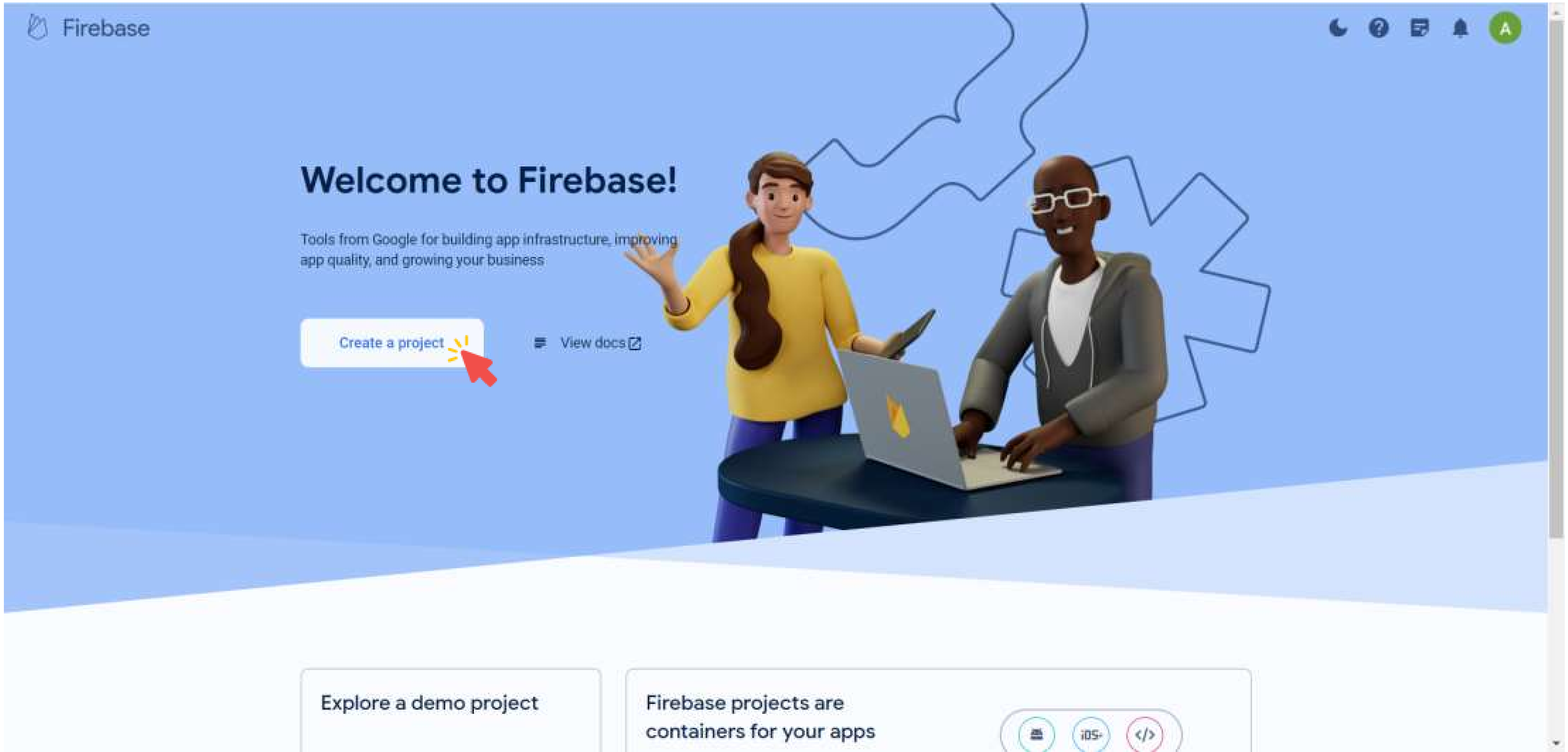


Make your app the best it can be

Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

Firebase: Creating a New Project

- Click on Create a Project.



Firebase

Welcome to Firebase!

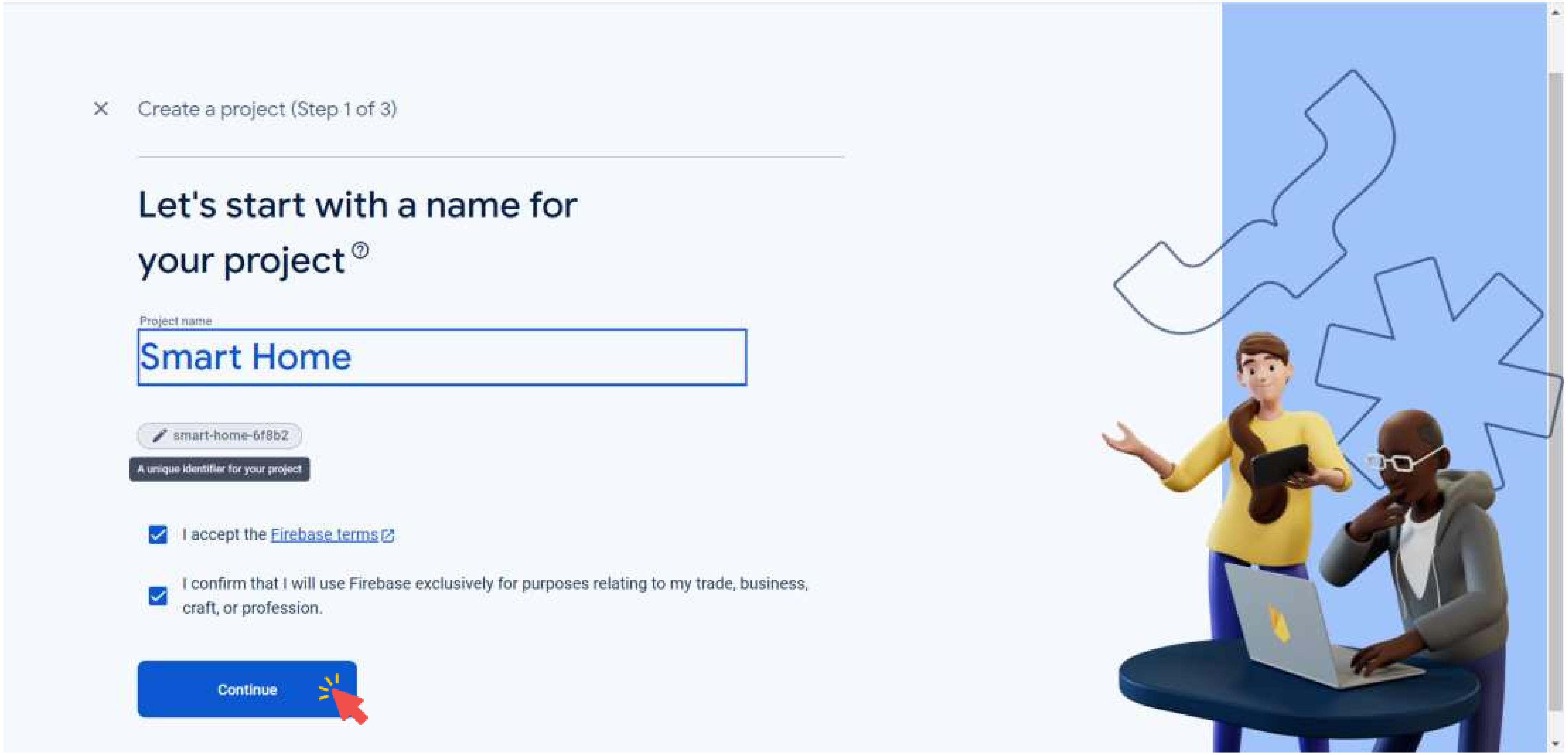
Tools from Google for building app infrastructure, improving app quality, and growing your business

Explore a demo project

Firebase projects are containers for your apps

Firebase: Creating a New Project

- Enter the name of your project, accept terms, and click Continue.



× Create a project (Step 1 of 3)

Let's start with a name for your project

Project name: Smart Home

smart-home-6f8b2

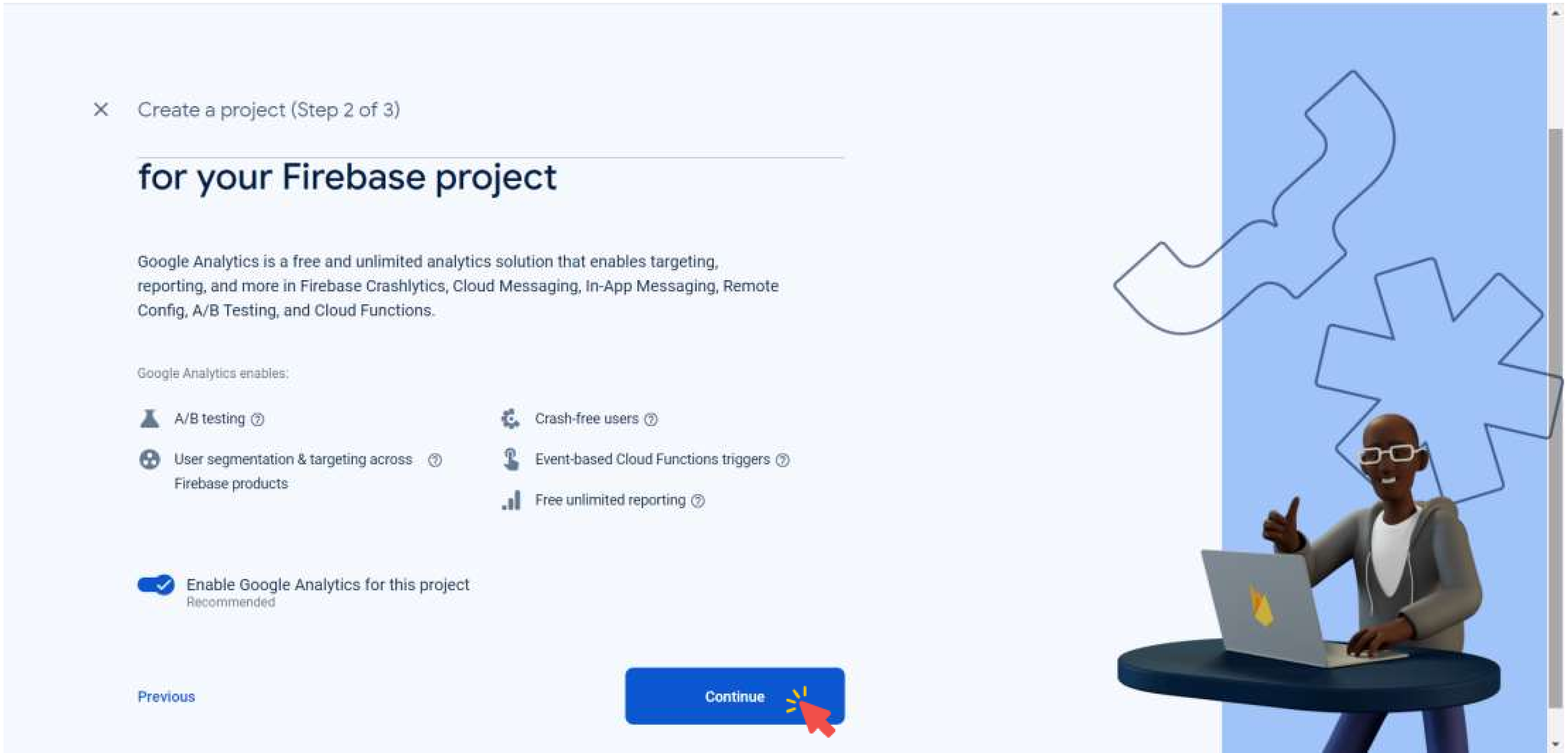
A unique identifier for your project

[✓] I accept the Firebase terms

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

Firebase: Creating a New Project

- Click Continue.


Create a project (Step 2 of 3)

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.

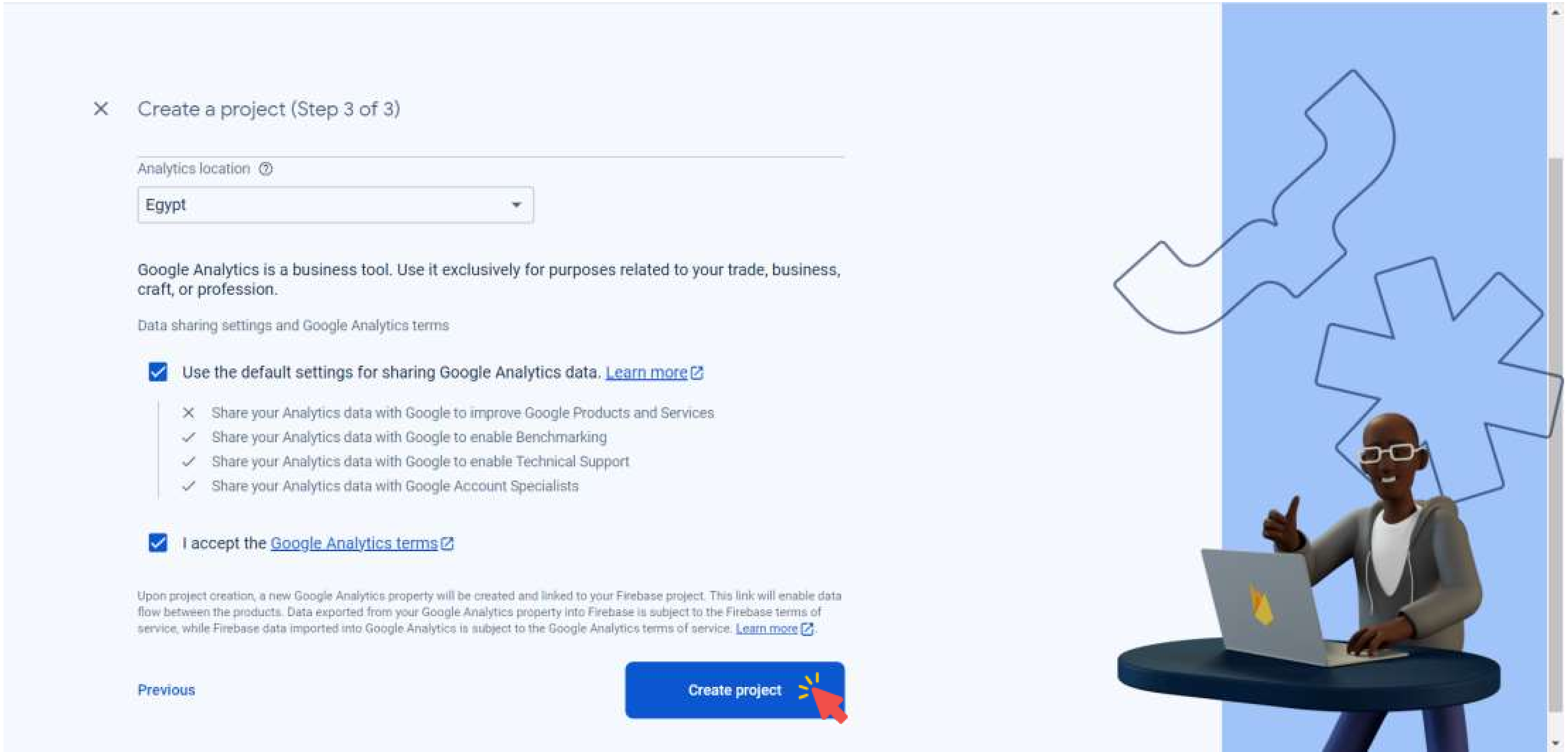
Google Analytics enables:

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

Enable Google Analytics for this project ✓
Recommended

Firebase: Creating a New Project

- Choose the location, accept terms, and click Create Project.



Create a project (Step 3 of 3)

Analytics location

Egypt

Google Analytics is a business tool. Use it exclusively for purposes related to your trade, business, craft, or profession.

Data sharing settings and Google Analytics terms

✓ Use the default settings for sharing Google Analytics data. Learn more

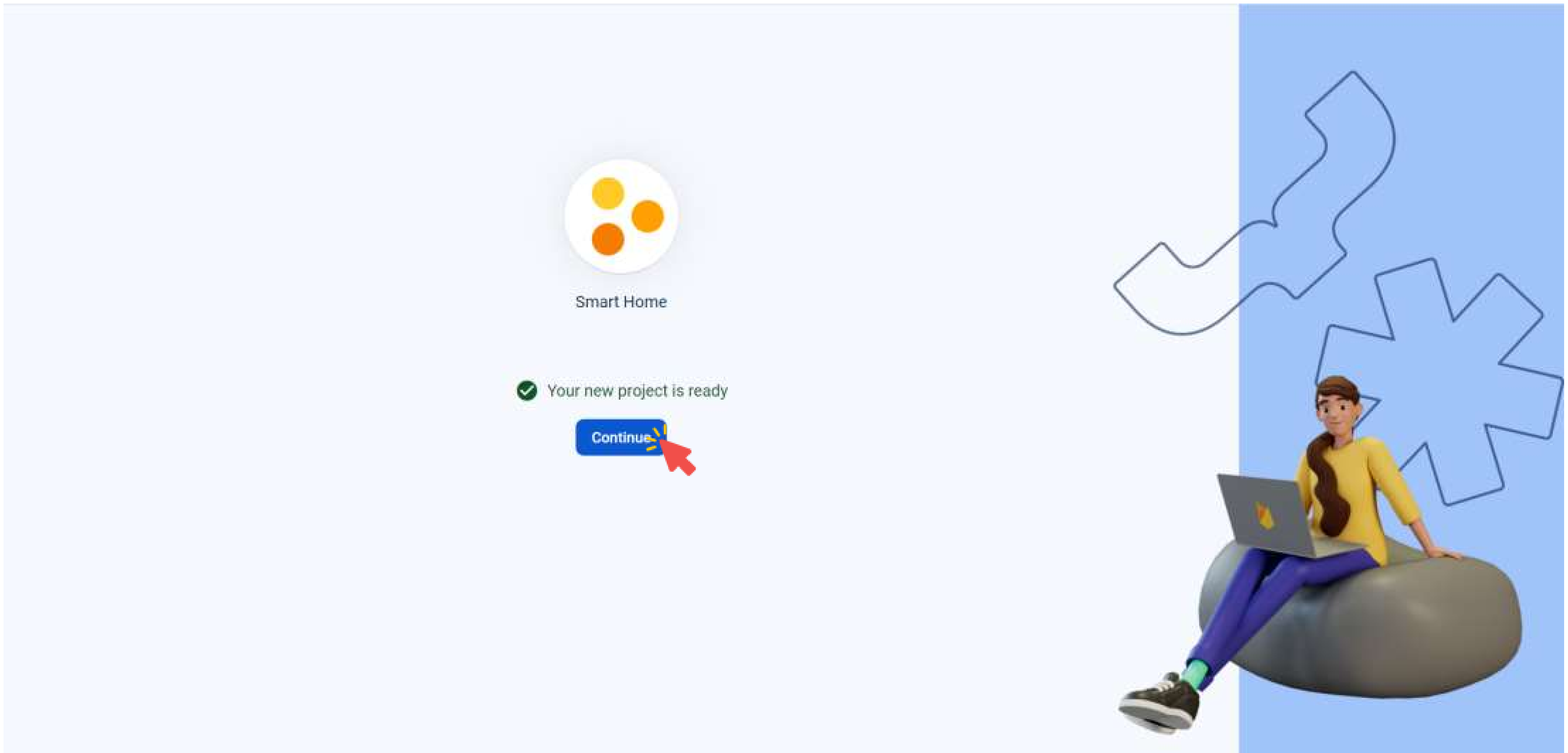
- ✗ Share your Analytics data with Google to improve Google Products and Services
- ✓ Share your Analytics data with Google to enable Benchmarking
- ✓ Share your Analytics data with Google to enable Technical Support
- ✓ Share your Analytics data with Google Account Specialists

✓ 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.

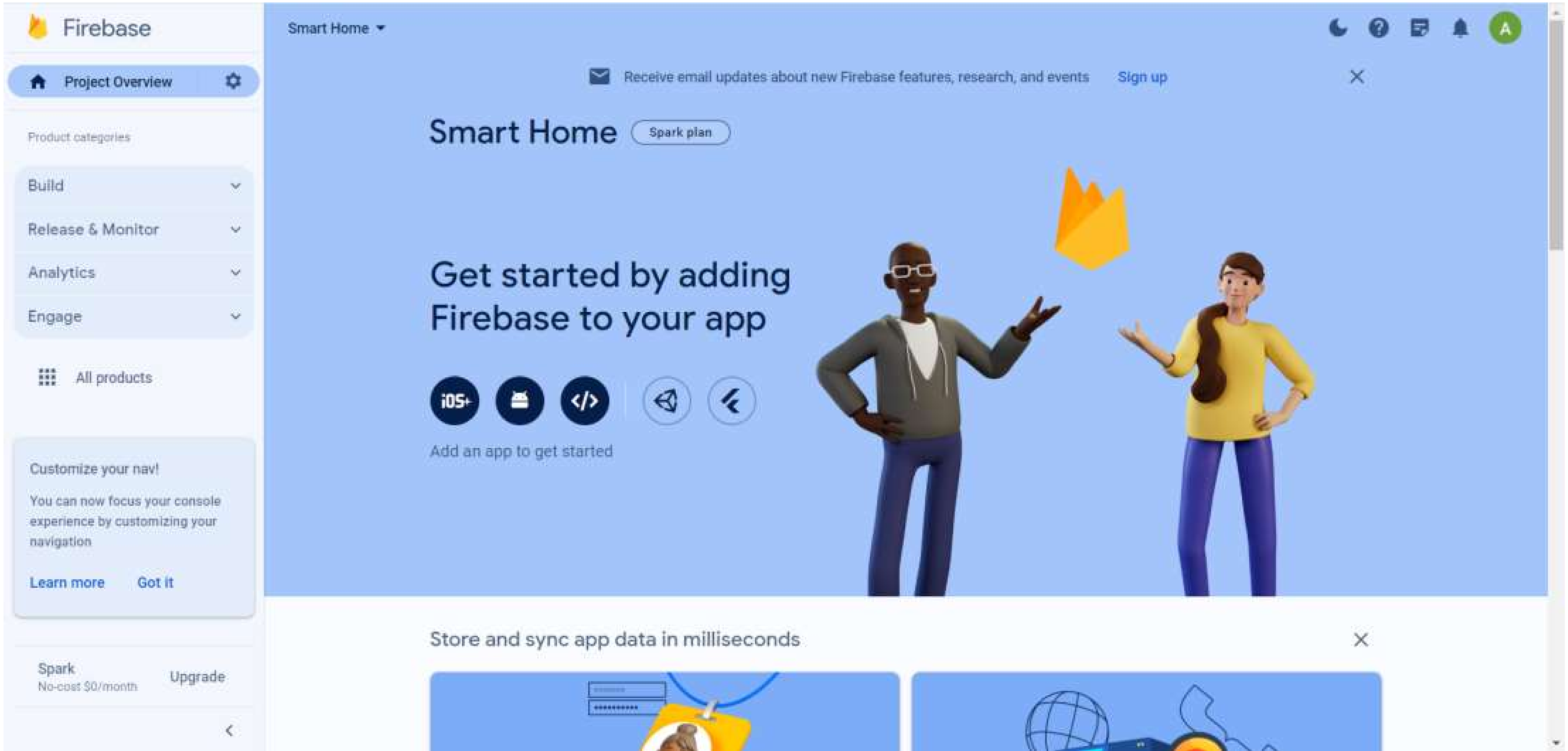
Firebase: Creating a New Project

- Wait until the project is created, and click Continue.



Firebase: Creating a New Project

- You'll be redirected to your project console page.



Firebase

Project Overview

Product categories

Build

Release & Monitor

Analytics

Engage

All products

Customize your nav!

You can now focus your console experience by customizing your navigation

Learn more Got it

Spark
No-cost $0/month

Upgrade

Smart Home

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

Smart Home Spark plan

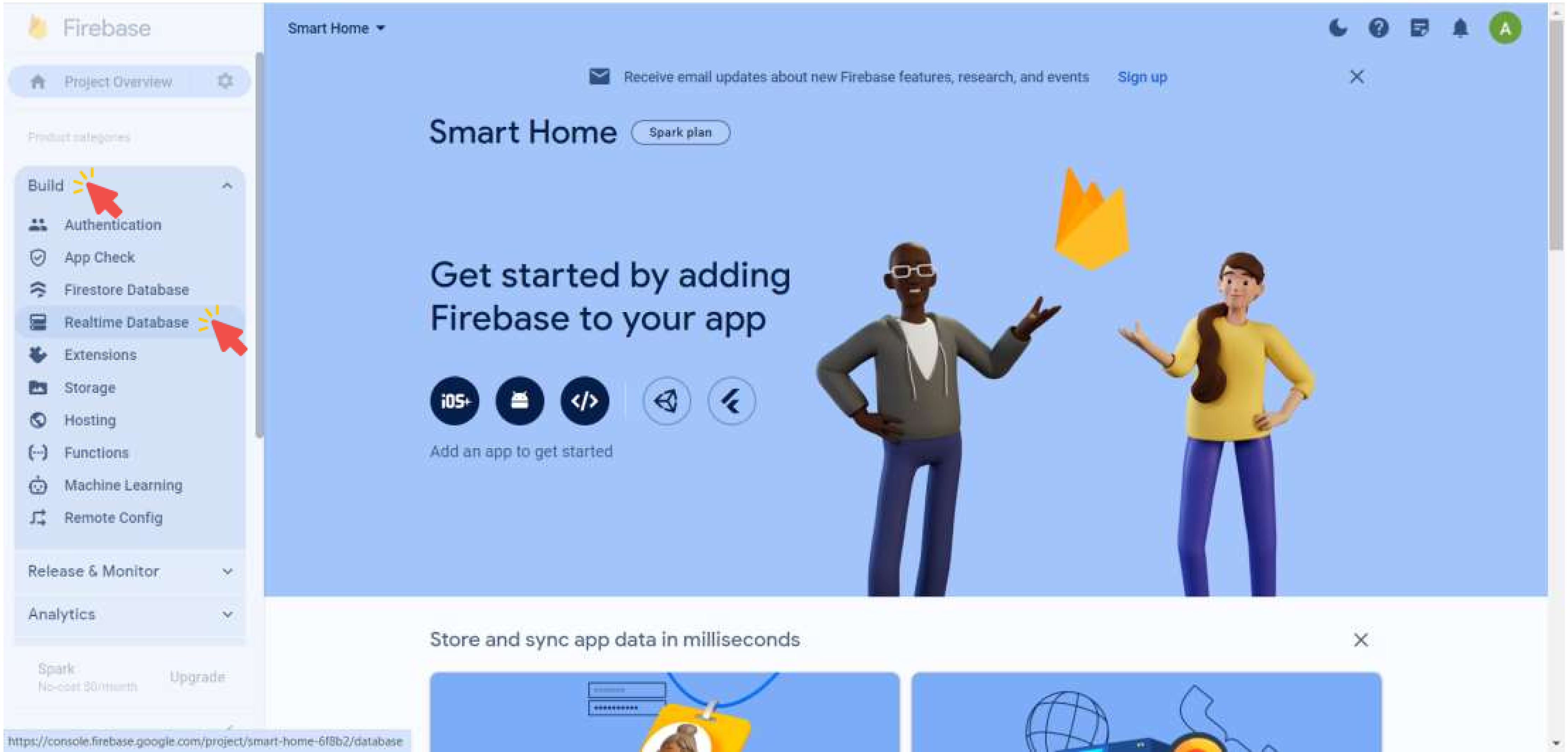
Get started by adding Firebase to your app

Add an app to get started

Store and sync app data in milliseconds

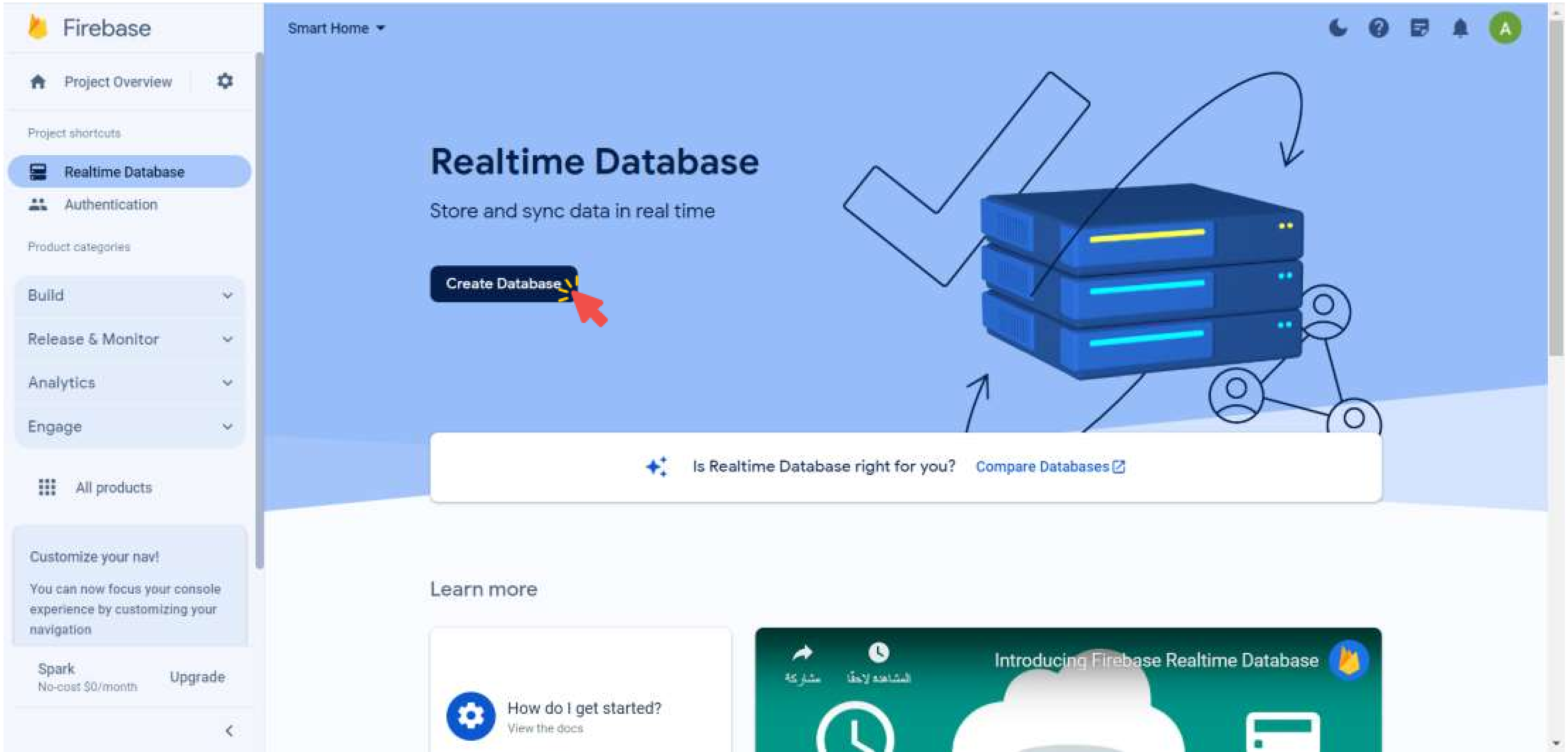
Firebase: Creating a Realtime Database

- Expand Build tap, and go to Realtime Database.



Firebase: Creating a Realtime Database

- Click on Create Database.



Firebase

Smart Home

Project Overview

Project shortcuts

Realtime Database

Authentication

Product categories

Build

Release & Monitor

Analytics

Engage

All products

Customize your nav!

You can now focus your console experience by customizing your navigation

Spark
No-cost $0/month

Upgrade

Realtime Database

Store and sync data in real time

Create Database

Is Realtime Database right for you? Compare Databases

Learn more

How do I get started?
View the docs

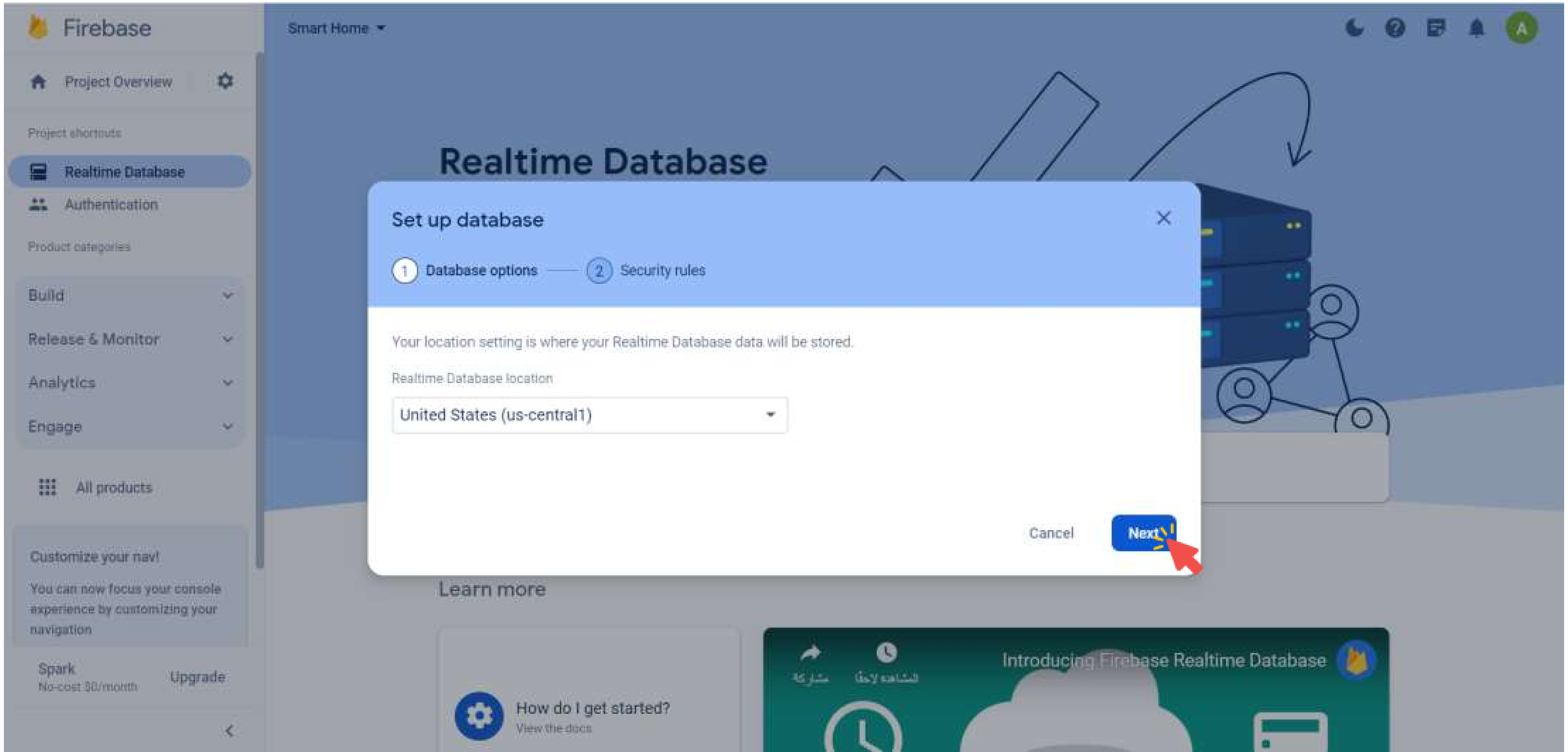
Introducing Firebase Realtime Database

مشاركة

المشاهدة لاحقًا

Firebase: Creating a Realtime Database

- Select your database location, and click Next.



Set up database

1 Database options — 2 Security rules

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

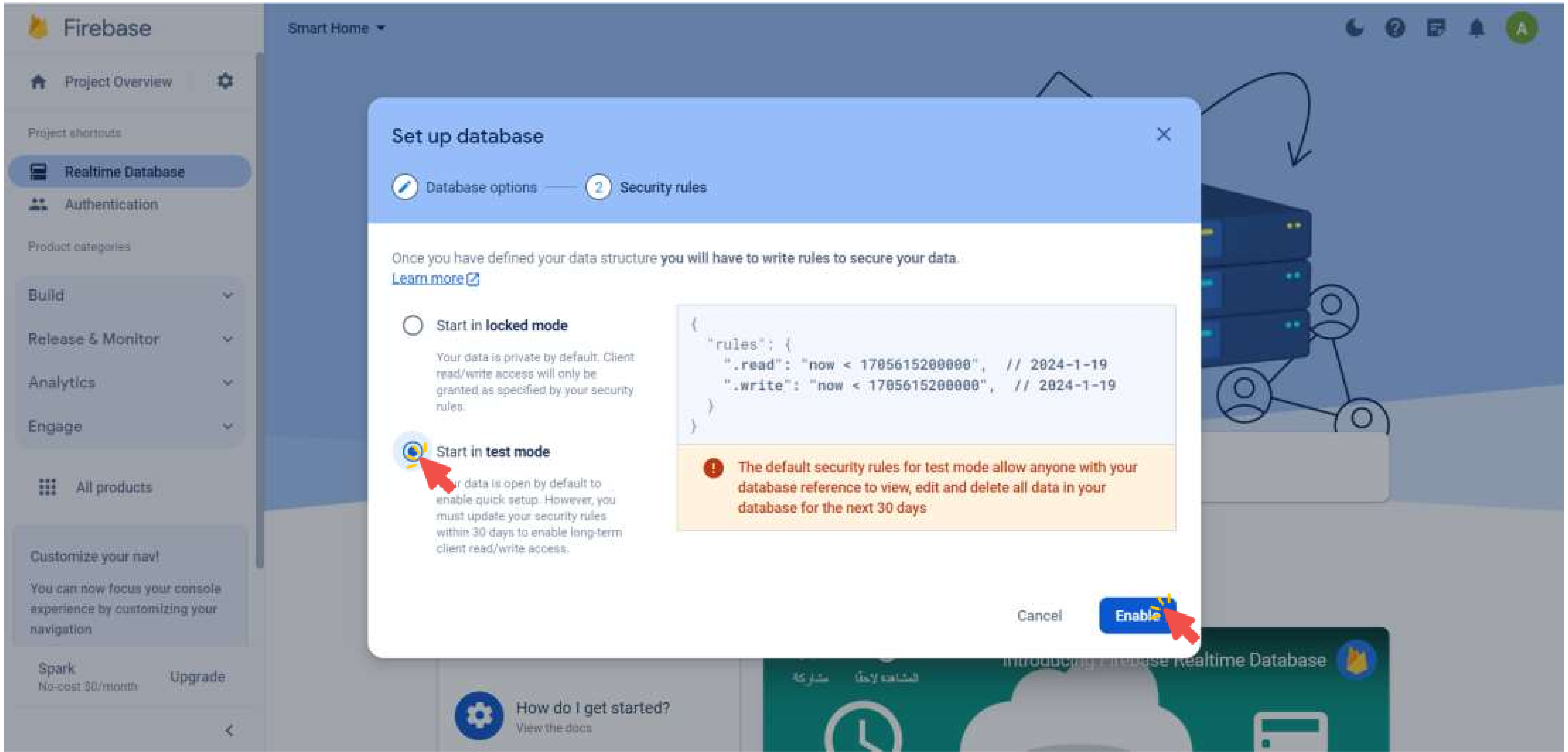
Realtime Database location

United States (us-central1)

Cancel Next

Firebase: Creating a Realtime Database

- For testing purposes, select Start in test mode, and click Enable.



Set up database

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.

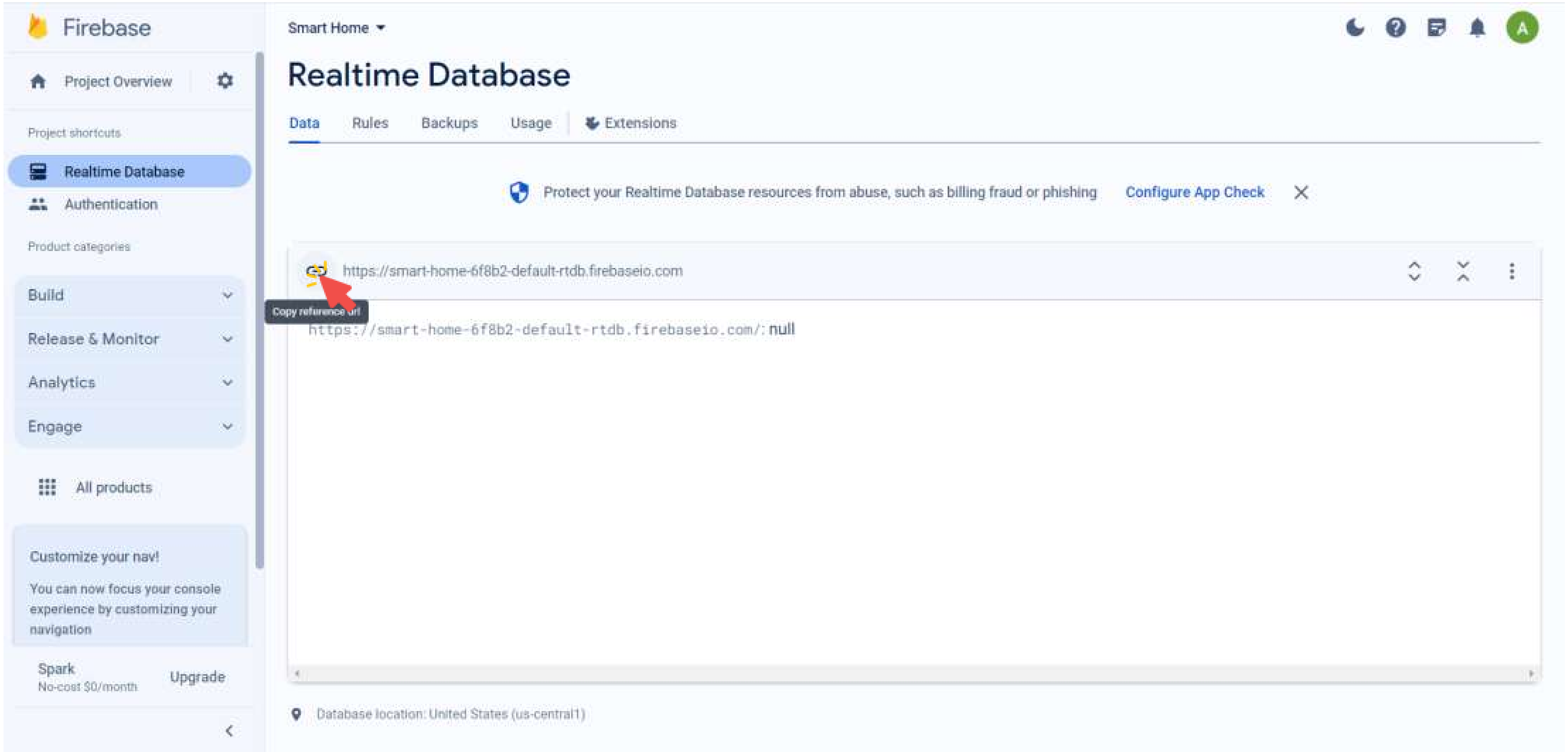
```
{
  "rules": {
    ".read": "now < 1705615200000",  // 2024-1-19
    ".write": "now < 1705615200000",  // 2024-1-19
  }
}
```

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 Enable

Firebase: Creating a Realtime Database

- Your database is now created. You need to copy the database URL.



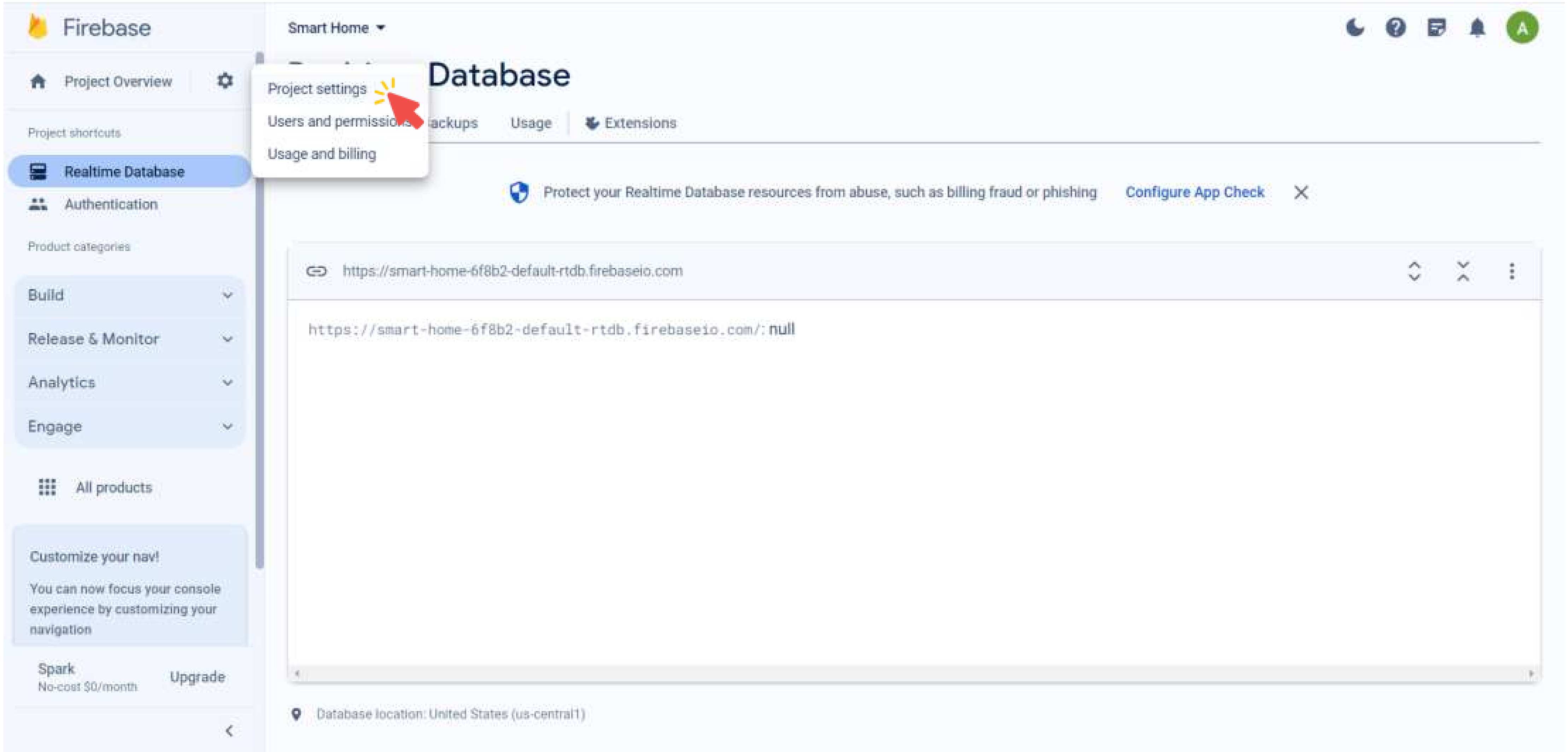
https://smart-home-6f8b2-default-rtdb.firebaseio.com

```
https://smart-home-6f8b2-default-rtdb.firebaseio.com/: null
```

Database location: United States (us-central1)

Firebase: Getting Database Secret

- Goto Project settings.



Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://smart-home-6f8b2-default-rtdb.firebaseio.com

```
https://smart-home-6f8b2-default-rtdb.firebaseio.com/: null
```

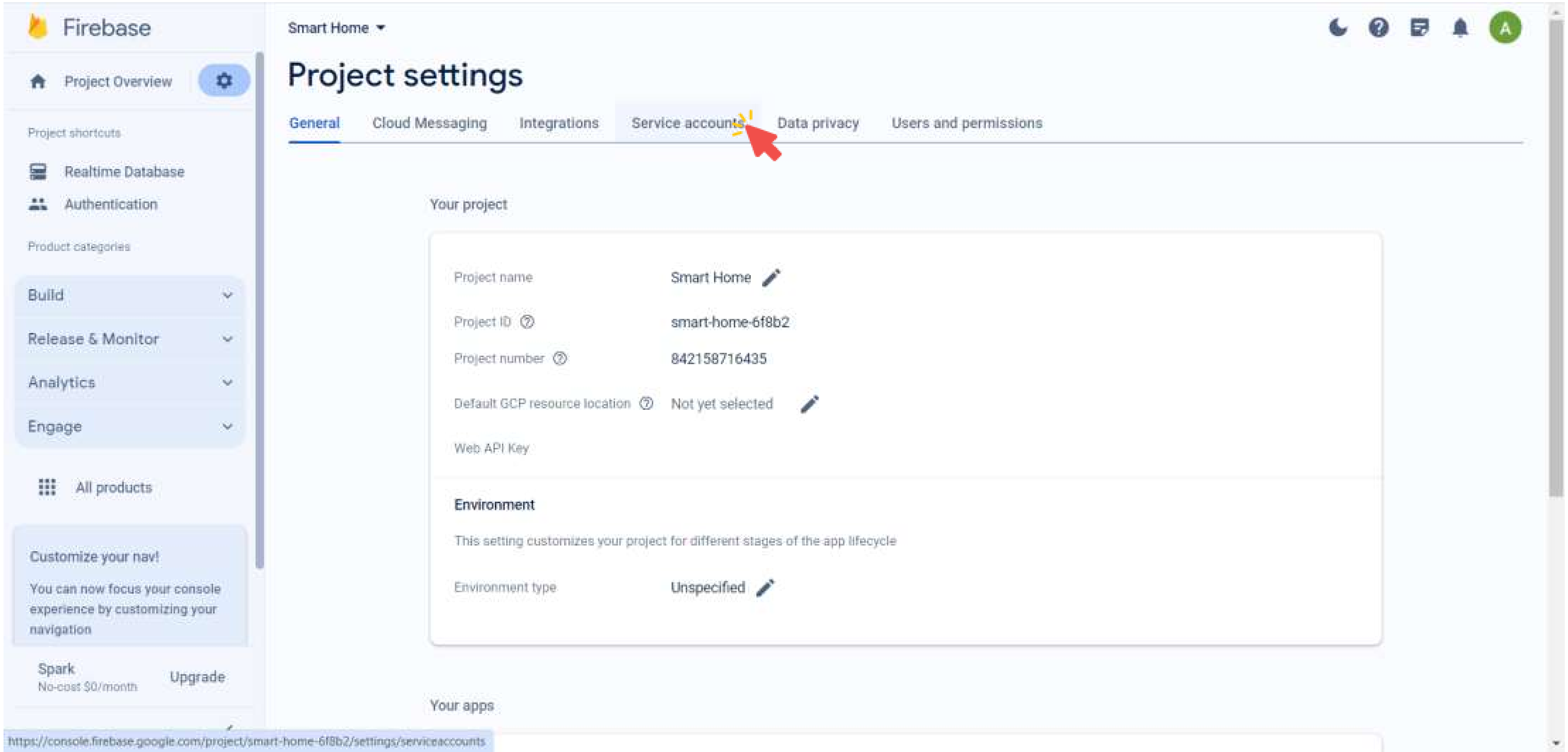
Customize your nav!

You can now focus your console experience by customizing your navigation

Database location: United States (us-central1)

Firebase: Getting Database Secret

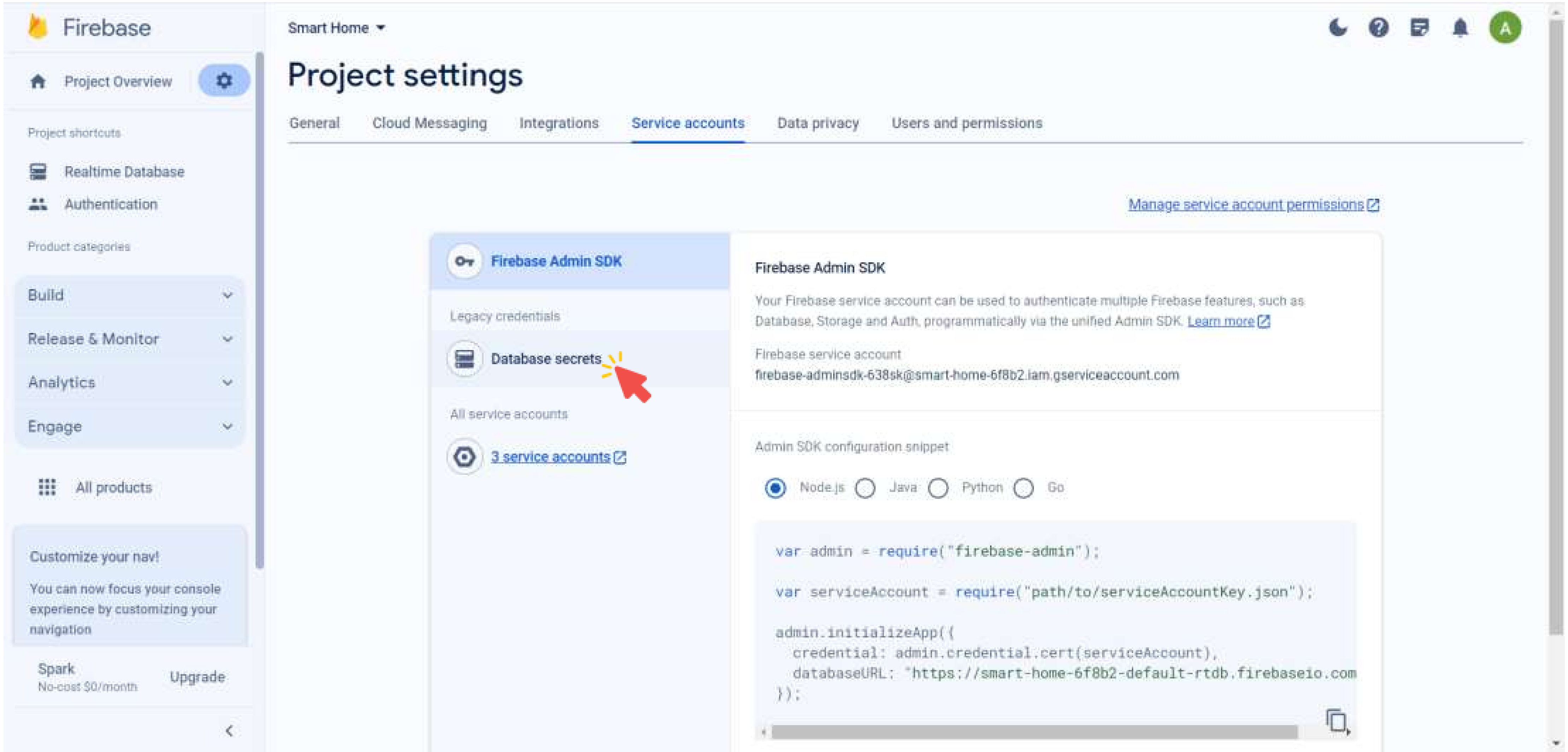
- Click on Service accounts.



Project name	Smart Home
Project ID	smart-home-6f8b2
Project number	842158716435
Default GCP resource location	Not yet selected
Web API Key	
Environment type	Unspecified

Firebase: Getting Database Secret

- Click on Database secrets.



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-638sk@smart-home-6f8b2.iam.gserviceaccount.com

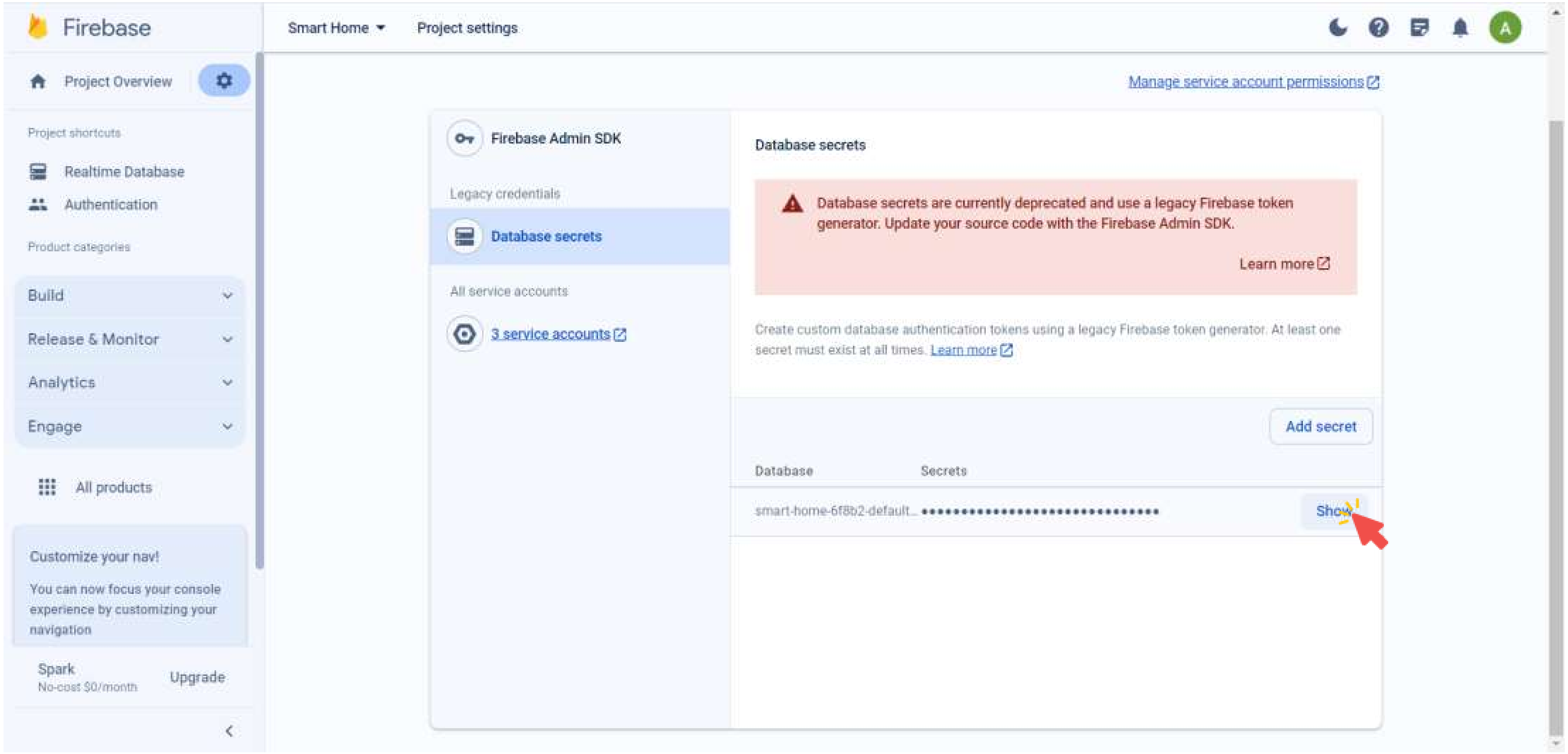
Admin SDK configuration snippet

[✓ 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://smart-home-6f8b2-default-rtdb.firebaseio.com
});
```

Firebase: Getting Database Secret

- Click on Show.



Database secrets

Database secrets are currently deprecated and use a legacy Firebase token generator. Update your source code with the Firebase Admin SDK.

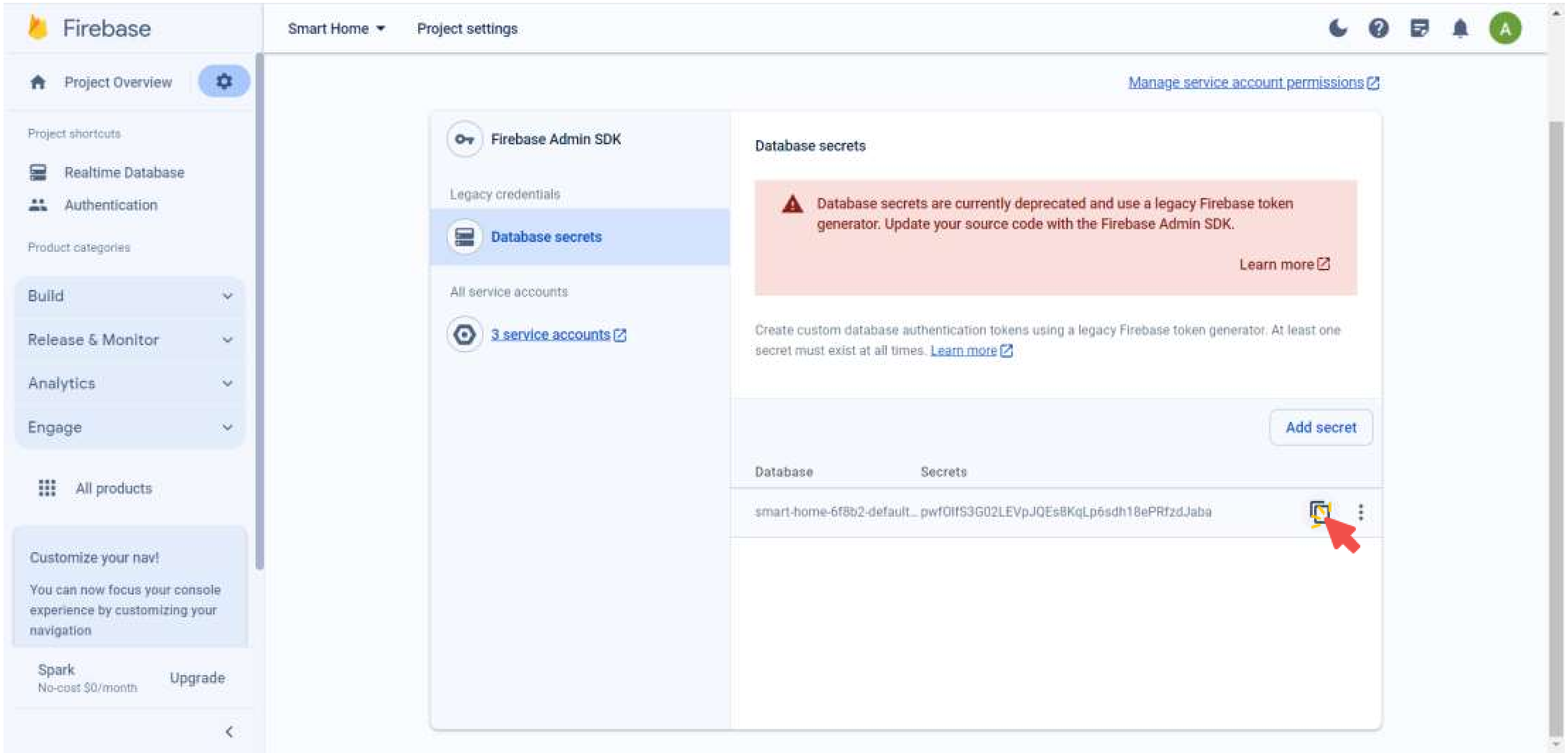
Learn more

Create custom database authentication tokens using a legacy Firebase token generator. At least one secret must exist at all times. Learn more

Database	Secrets	
smart-home-6f8b2-default...	••••••••••••••••••••••••••••••	Show

Firebase: Getting Database Secret

- Copy the Secret.



Database secrets

Database secrets are currently deprecated and use a legacy Firebase token generator. Update your source code with the Firebase Admin SDK.

Learn more

Create custom database authentication tokens using a legacy Firebase token generator. At least one secret must exist at all times. Learn more

Add secret

Database	Secrets
smart-home-6f8b2-default...	pwfOlfS3G02LEVpJQEs8KqLp6sdh18ePRfzdJaba

Firebase: Structure Your Database

- All Firebase Realtime Database data is stored as JSON objects.
- You can think of the database as a cloud-hosted JSON tree.
- Unlike a SQL database, there are no tables or records.
- When you add data to the JSON tree, it becomes a node in the existing JSON structure with an associated key.

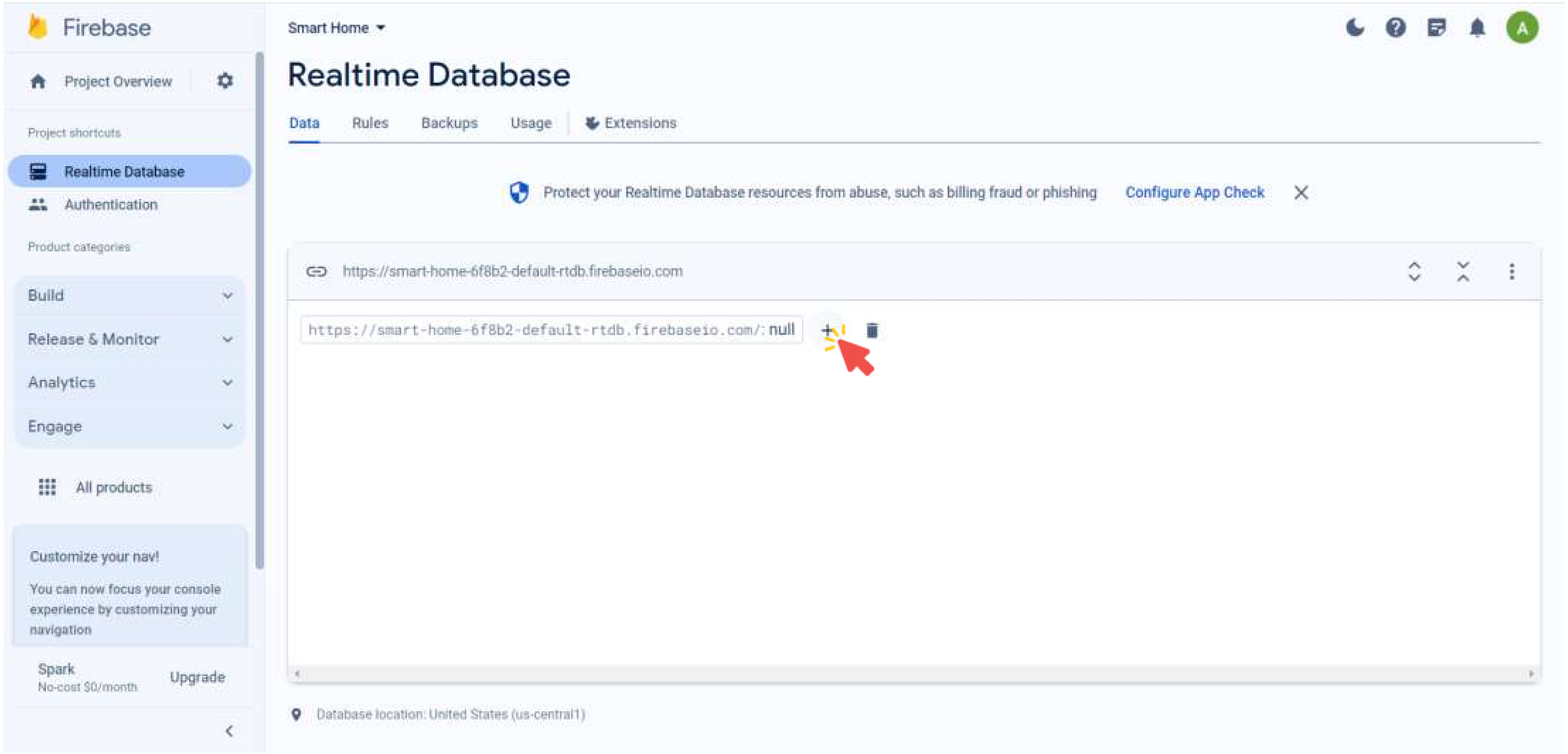
```
{
  "smartHome": {
    "kitchen": {
      "light": "off",
      "temperature": 20
    },
    "bedroom": {
      "light": "on",
      "temperature": 18
    }
  }
}
```

Firebase: Structure Your Database

```json
{
  "smartHome": {
    "devices": {
      "light": {
        "status": "on",
        "brightness": 75
      },
      "thermostat": {
        "temperature": 22,
        "mode": "auto"
      },
      "door": {
        "status": "closed"
      }
    },
    "security": {
      "alarm": {
        "status": "armed"
      },
      "camera": {
        "status": "active"
      }
    }
  }
}
```

Firebase: Adding Nodes to Your Database

- Add a new node.



Firebase

Project Overview

Project shortcuts

Realtime Database

Authentication

Product categories

Build

Release & Monitor

Analytics

Engage

All products

Customize your nav!

You can now focus your console experience by customizing your navigation

Spark
No-cost $0/month

Upgrade

Smart Home

Realtime Database

Data Rules Backups Usage Extensions

Protect your Realtime Database resources from abuse, such as billing fraud or phishing Configure App Check

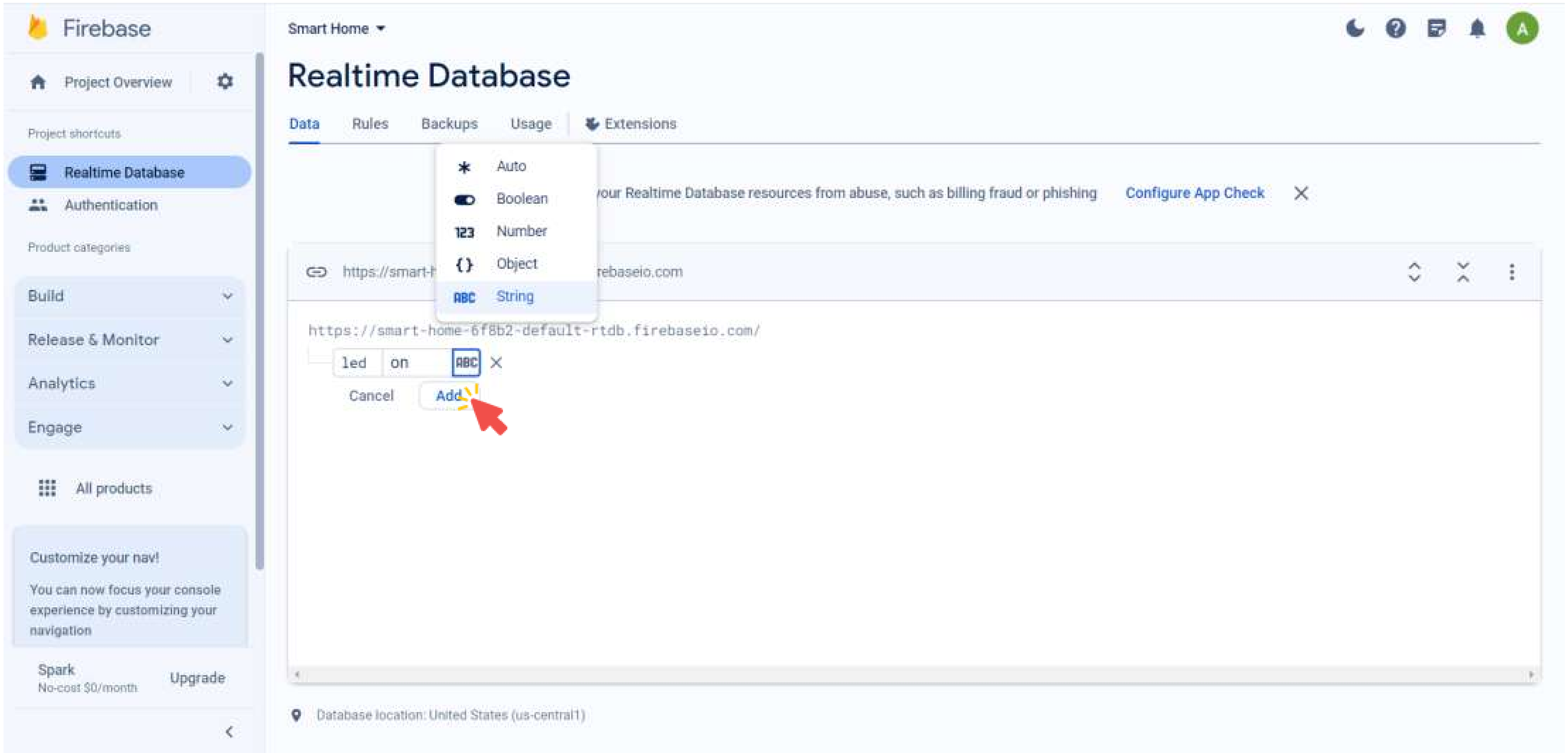
https://smart-home-6f8b2-default-rtdb.firebaseio.com

```
https://smart-home-6f8b2-default-rtdb.firebaseio.com/: null
```

Database location: United States (us-central1)

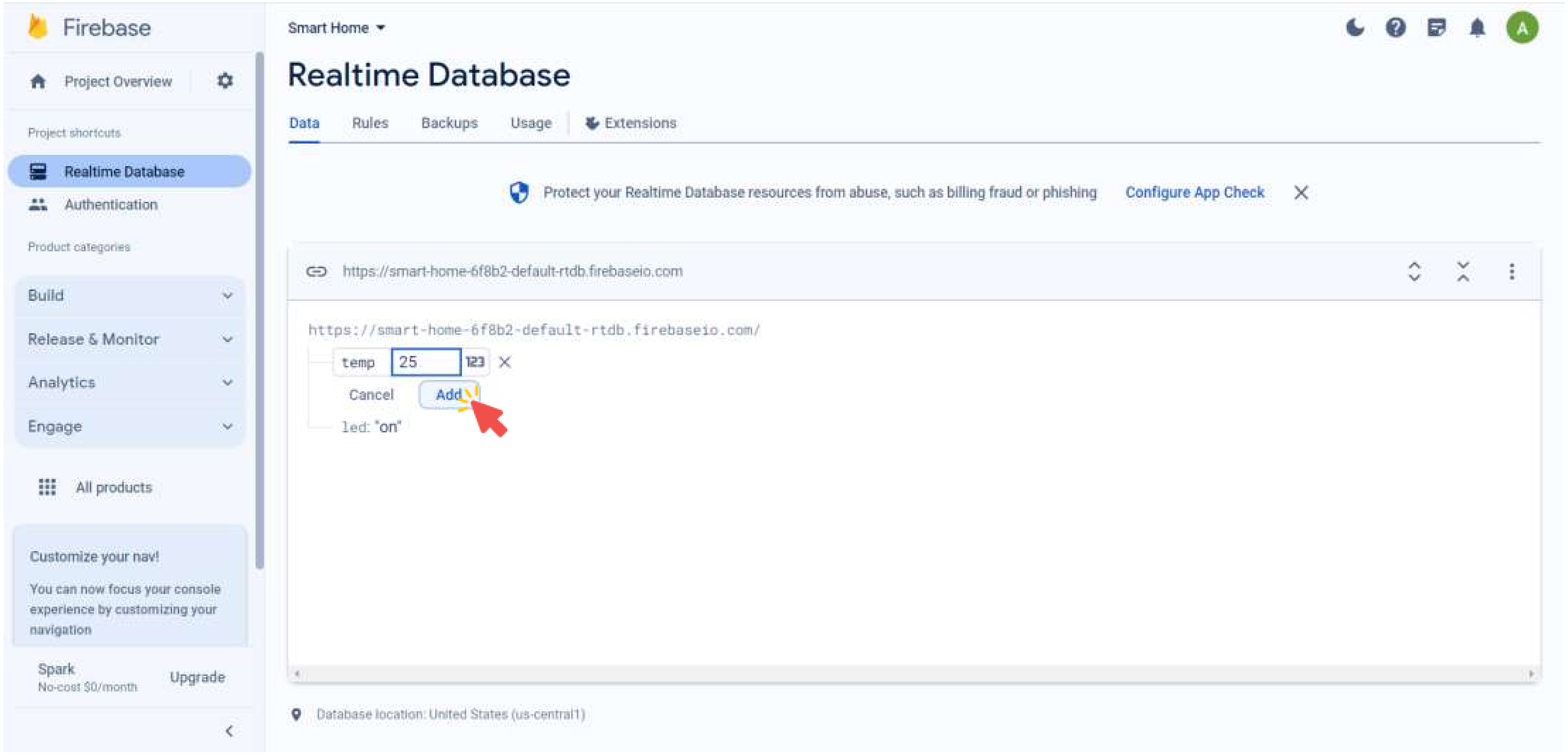
Firebase: Adding Nodes to Your Database

- Enter node key, value, choose its data type, and click Add.



Firebase: Adding Nodes to Your Database

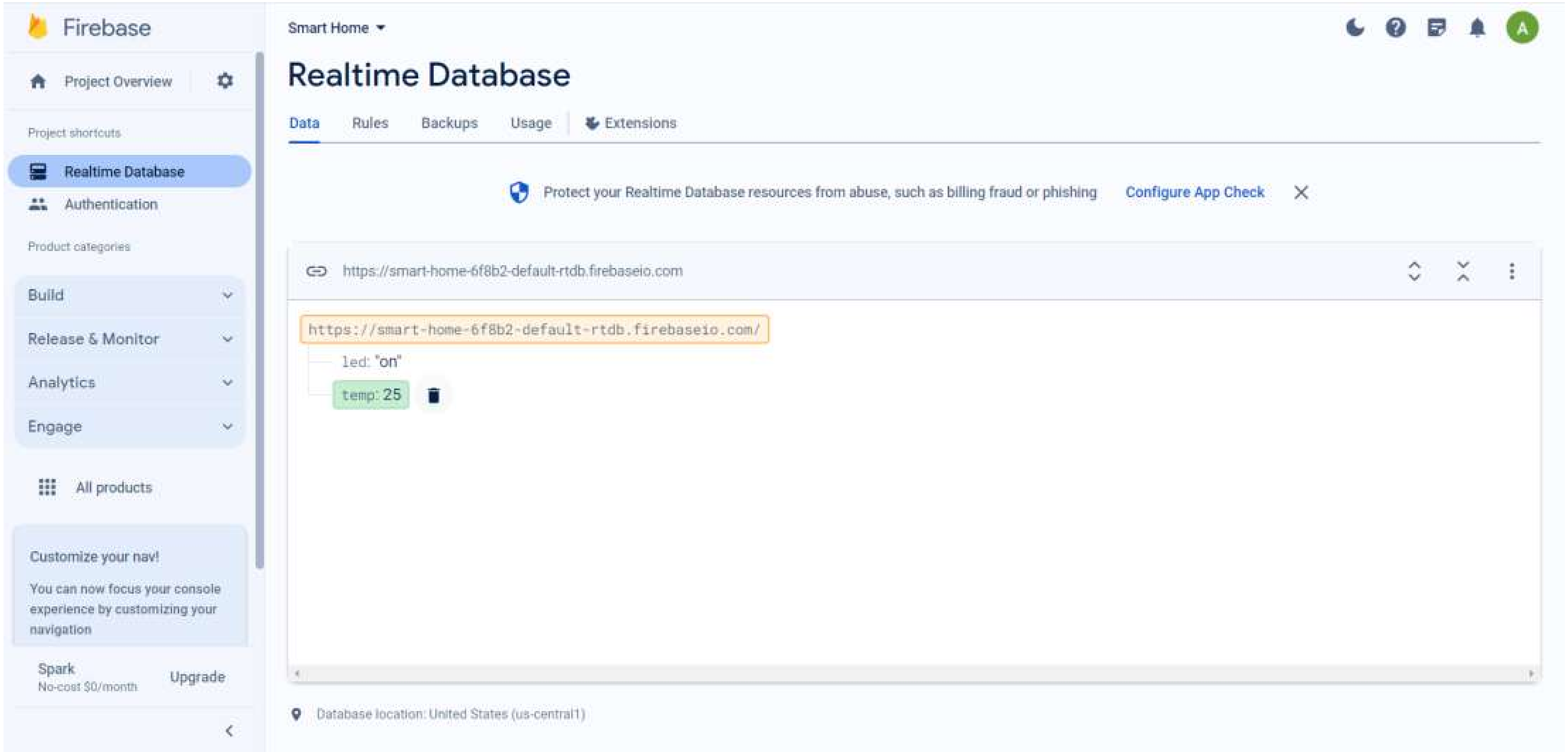
- Add a second node.



```
https://smart-home-6f8b2-default-rtdb.firebaseio.com/
  temp  25  123
  led: "on"
```

Firebase: Adding Nodes to Your Database

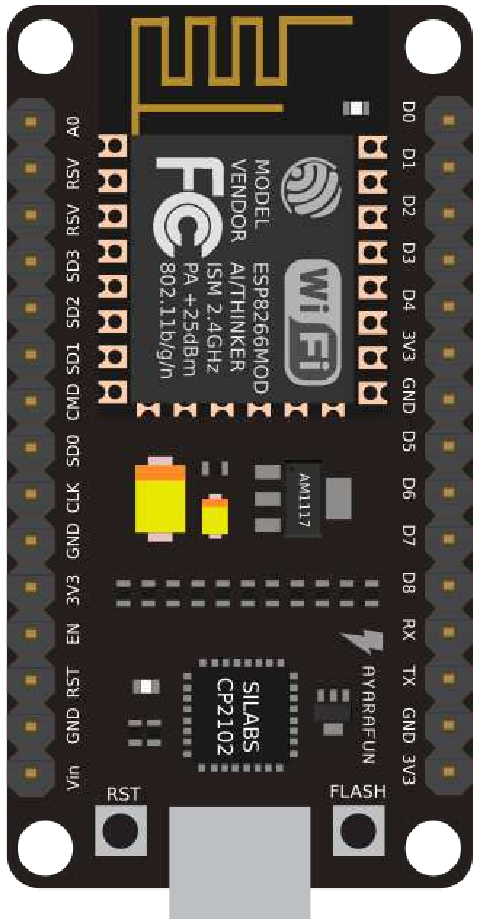
- Now, we have two nodes.



```
https://smart-home-6f8b2-default-rtdb.firebaseio.com/
    led: "on"
    temp: 25
```

FirebaseESP8266 Library

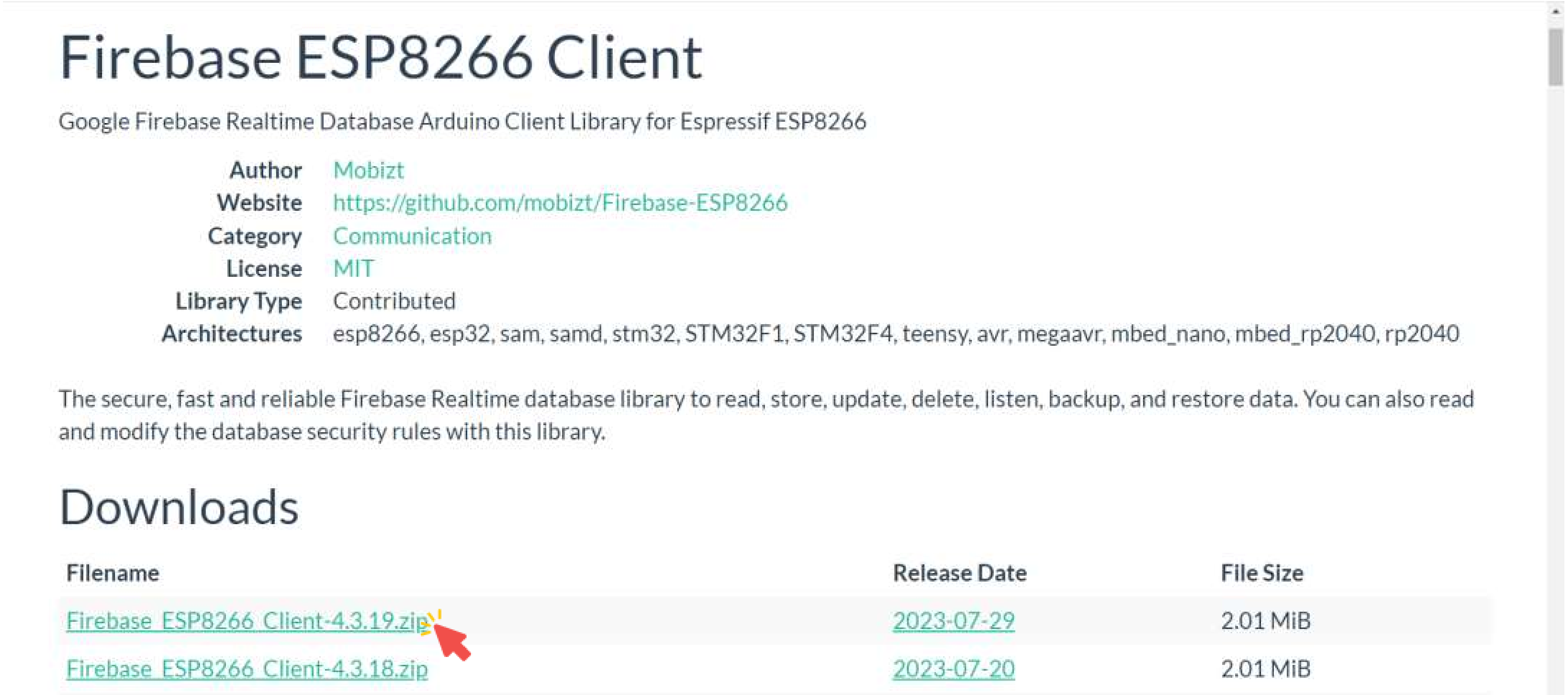
- The FirebaseESP8266 library provides Firebase Realtime database and Firebase Cloud Messaging functions and supports ESP8266 MCU.



Firebase

FirebaseESP8266 Library: Installation

- Download the ZIP file from he following link
  https://www.arduinolibraries.info/libraries/firebase-esp8266-client



Firebase ESP8266 Client

Google Firebase Realtime Database Arduino Client Library for Espressif ESP8266

Author	Mobizt
Website	https://github.com/mobizt/Firebase-ESP8266
Category	Communication
License	MIT
Library Type	Contributed
Architectures	esp8266, esp32, sam, samd, stm32, STM32F1, STM32F4, teensy, avr, megaavr, mbed_nano, mbed_rp2040, rp2040

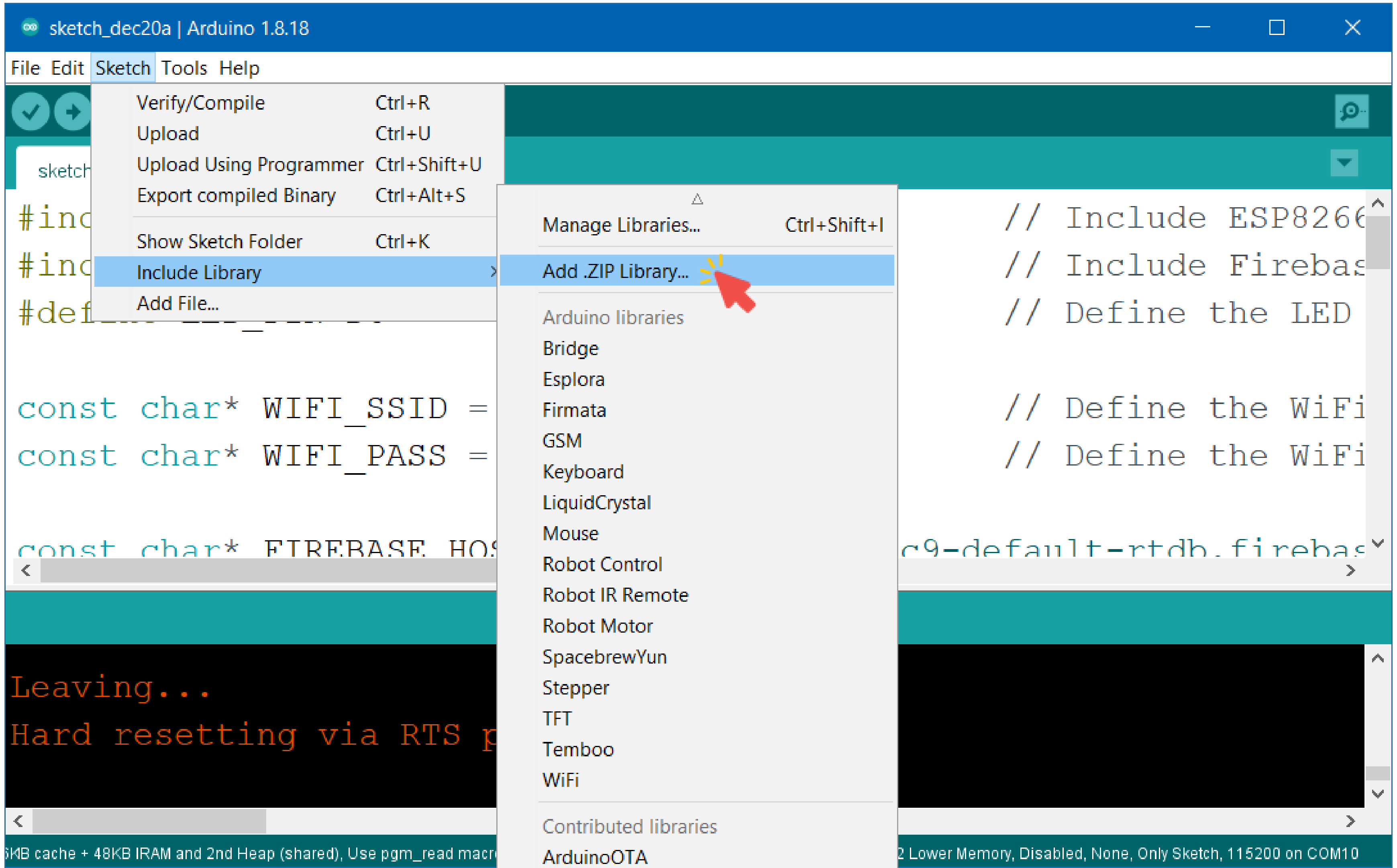
The secure, fast and reliable Firebase Realtime database library to read, store, update, delete, listen, backup, and restore data. You can also read and modify the database security rules with this library.

Downloads

Filename	Release Date	File Size
Firebase_ESP8266_Client-4.3.19.zip	2023-07-29	2.01 MiB
Firebase_ESP8266_Client-4.3.18.zip	2023-07-20	2.01 MiB

FirebaseESP8266 Library: Installation

- Select Sketch → Include Library → Add .ZIP Library...



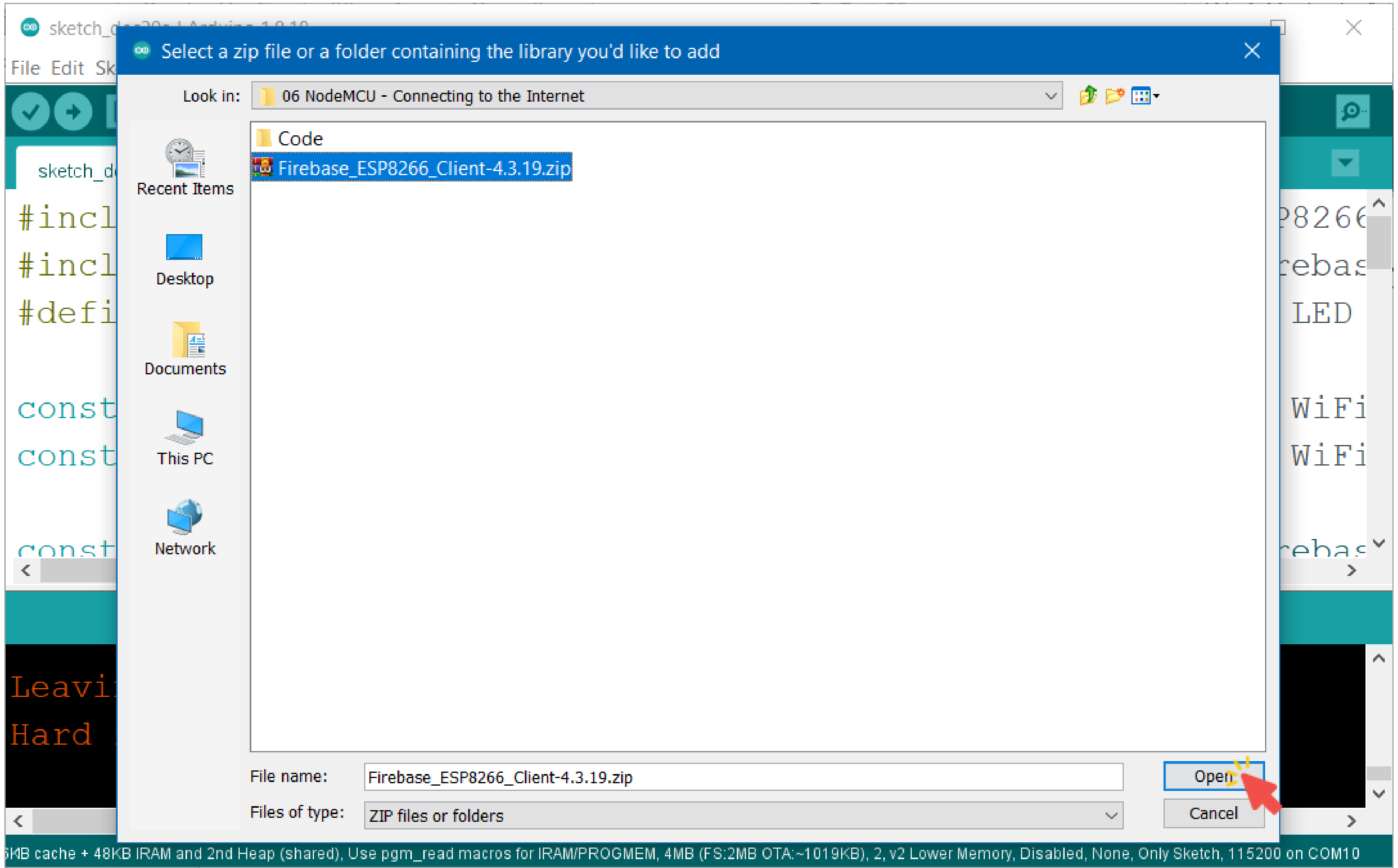
```
#inc                                    // Include ESP826
#inc                                    // Include Firebas
#def                                    // Define the LED

const char* WIFI_SSID =                 // Define the WiFi
const char* WIFI_PASS =                 // Define the WiFi

const char* FIREBASE_HOS                c9-default-rtdb.firebas
```

```
Leaving...
Hard resetting via RTS p
```

KB cache + 48KB IRAM and 2nd Heap (shared), Use pgm_read macr … 2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM10

FirebaseESP8266 Library: Installation

- Choose `Firebase_ESP8266_Client-4.3.19.zip` that previously downloaded.



Select a zip file or a folder containing the library you'd like to add

Look in: 06 NodeMCU - Connecting to the Internet

Code

Firebase_ESP8266_Client-4.3.19.zip

File name: Firebase_ESP8266_Client-4.3.19.zip

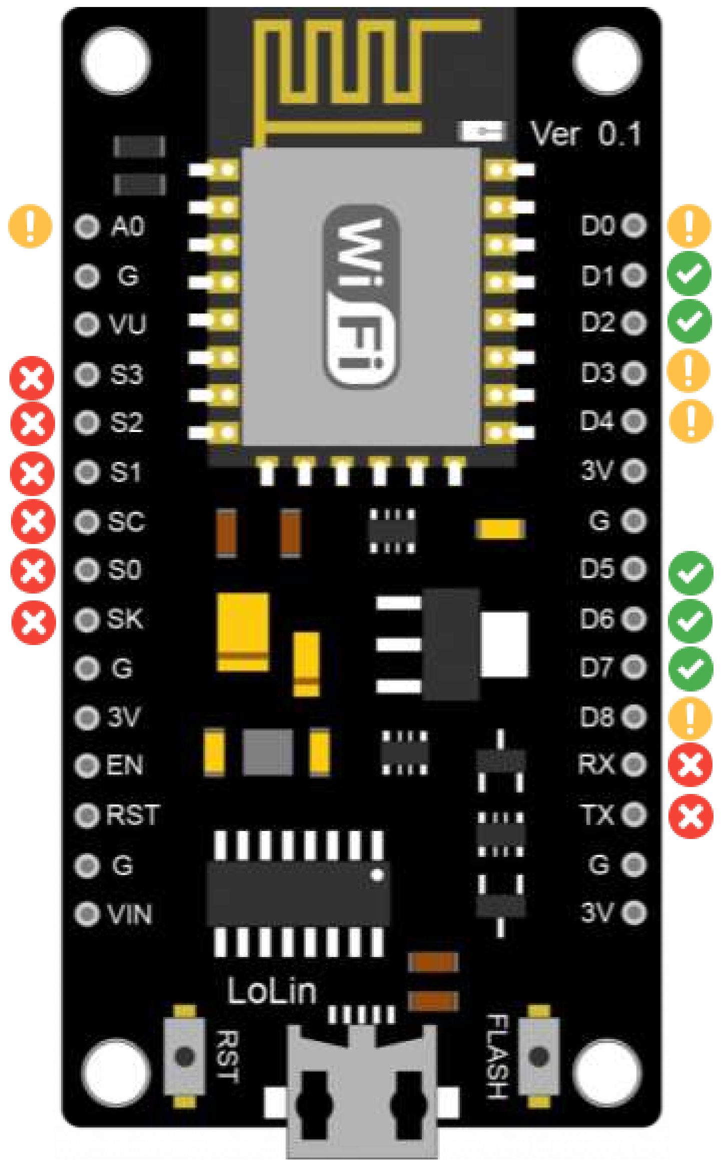
Files of type: ZIP files or folders

```
Leavi
Hard
```

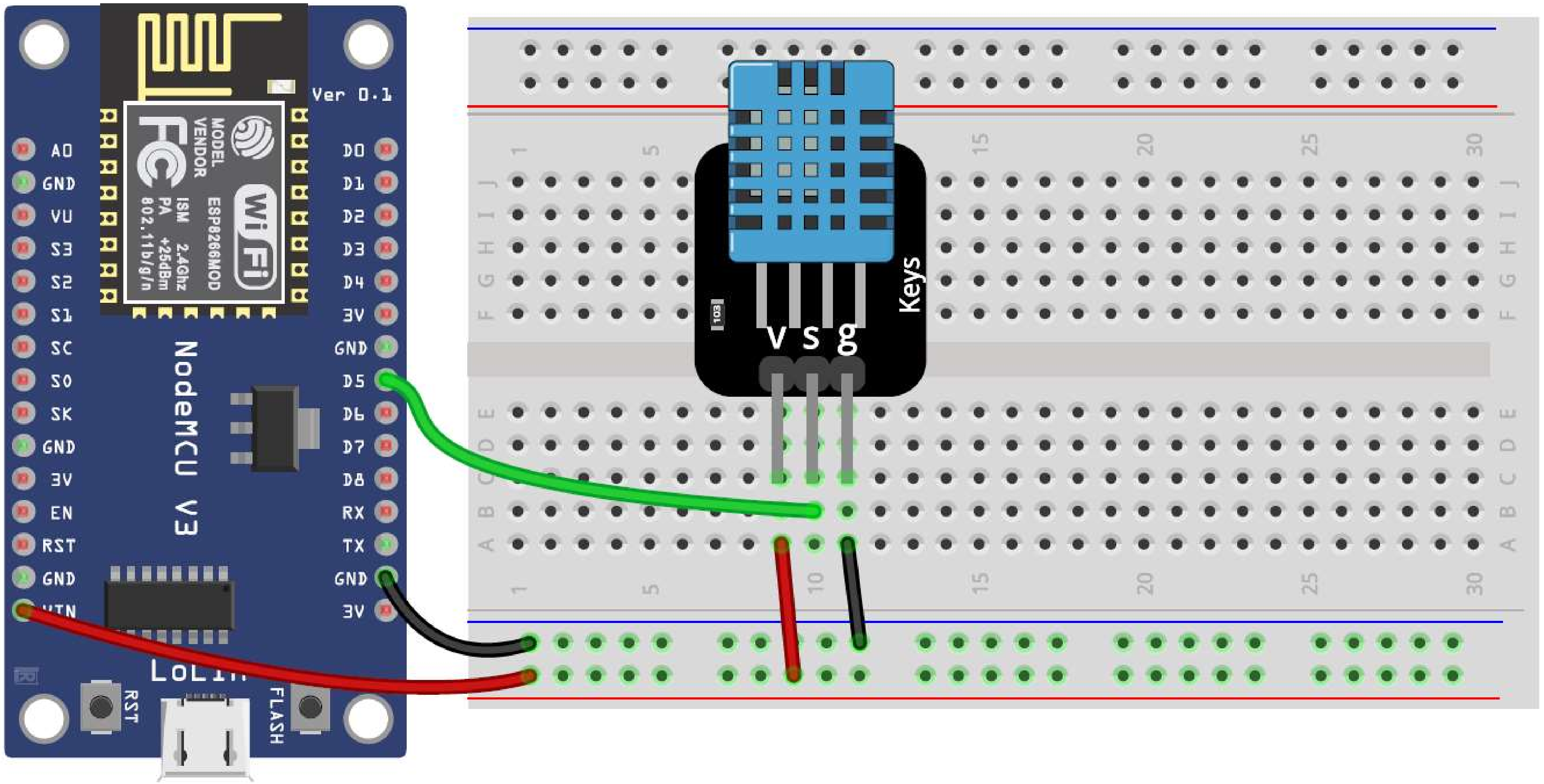
KB cache + 48KB IRAM and 2nd Heap (shared), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM10

Sending Data to Firebase: NodeMCU ESP8266 Pinout

PIN	GPIO	Why Not Safe?
D0	GPIO16	HIGH at boot Used to wake up from deep sleep
D1	GPIO5	-
D2	GPIO4	-
D3	GPIO0	Connected to FLASH button Boot fails if pulled LOW
D4	GPIO2	HIGH at boot Boot fails if pulled LOW
D5	GPIO14	-
D6	GPIO12	-
D7	GPIO13	-
D8	GPIO15	Required for boot Boot fails if pulled HIGH

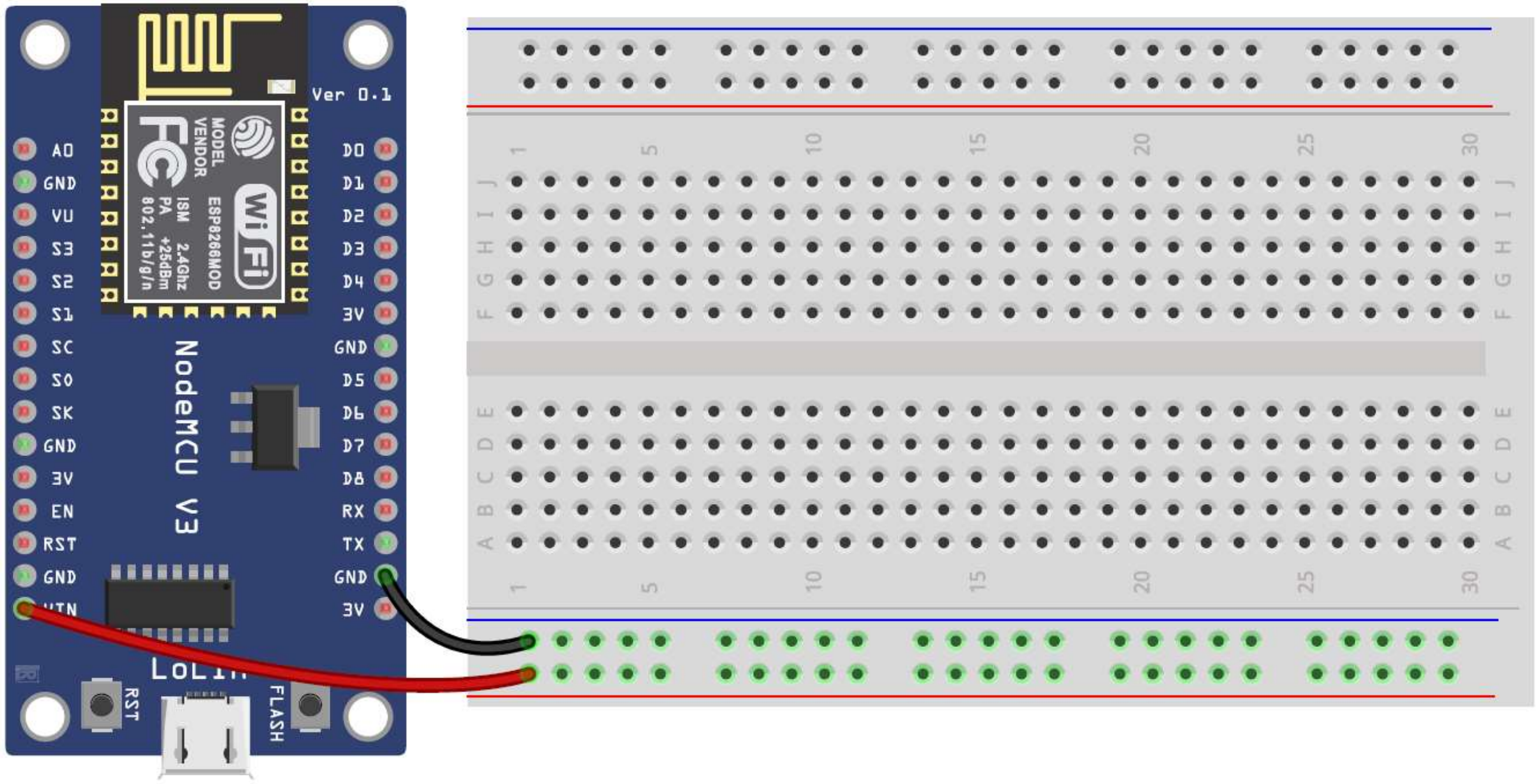


Sending Data to Firebase: DHT11 – Circuit



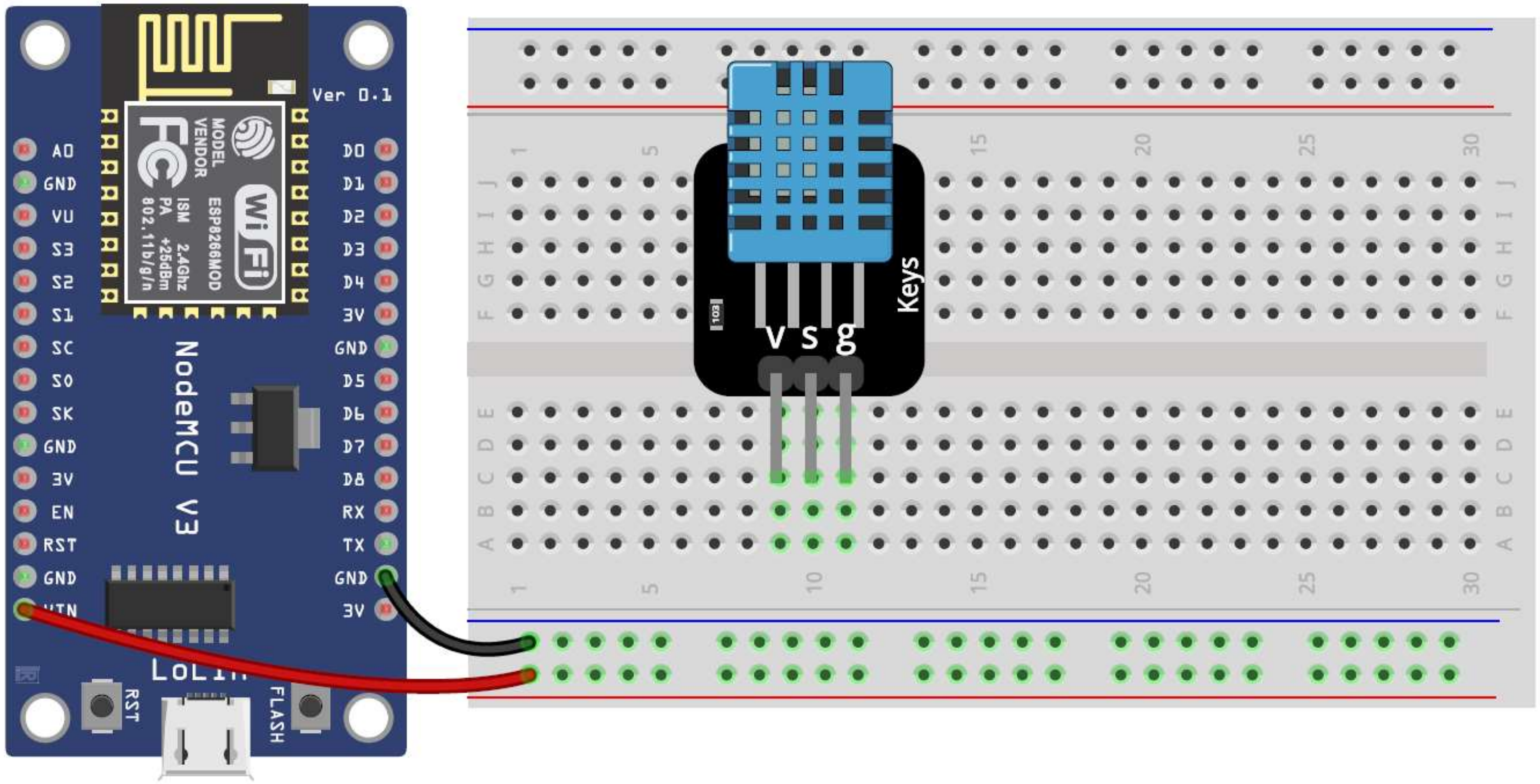
Sending Data to Firebase: DHT11 – Steps

1. Connect breadboard power (+) and ground (-) rails to NodeMCU VIN and ground (GND), respectively.



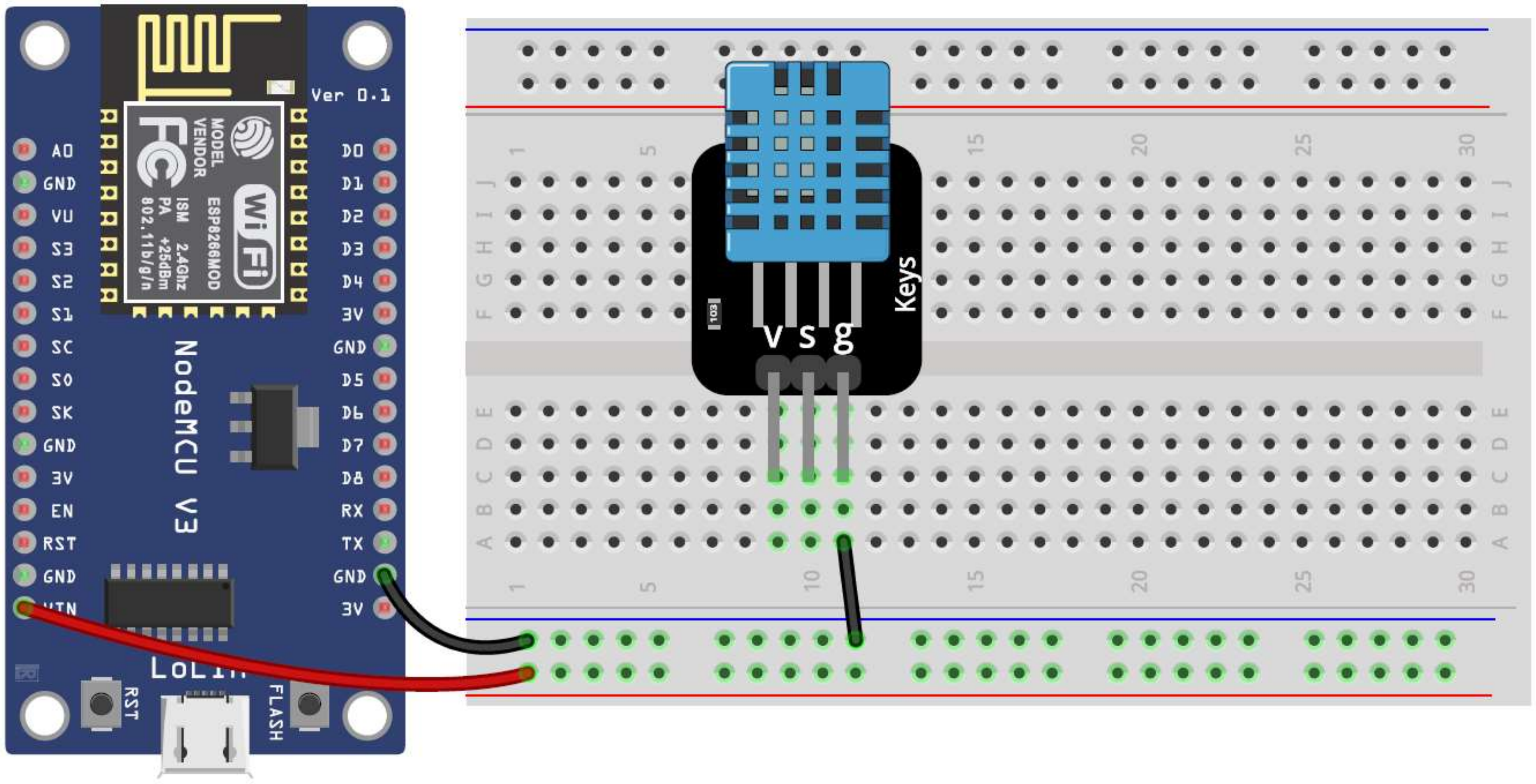
Sending Data to Firebase: DHT11 – Steps

2. Plug the DHT11 sensor into the breadboard.



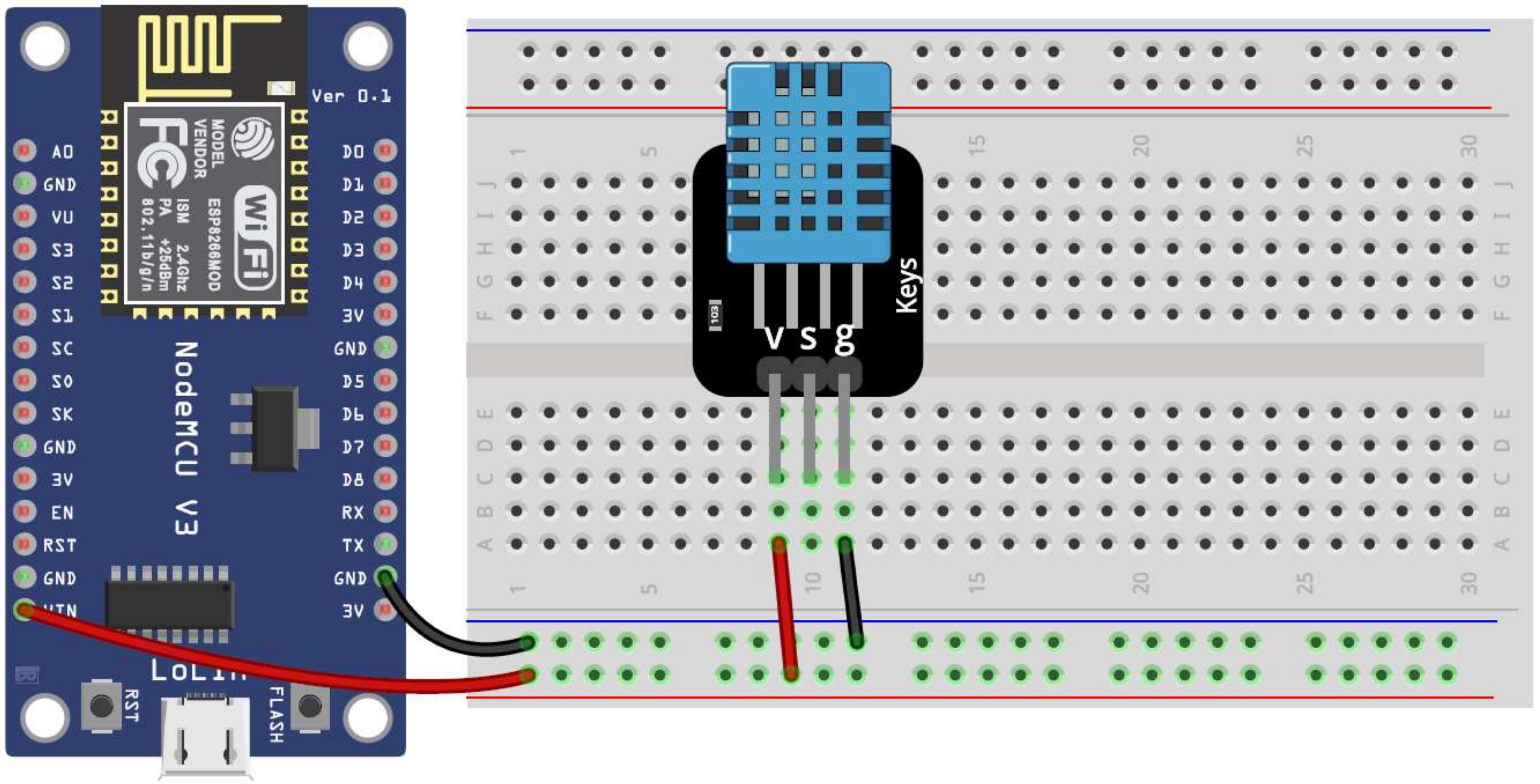
Sending Data to Firebase: DHT11 – Steps

3. The sensor GND pin connects to the ground on NodeMCU.



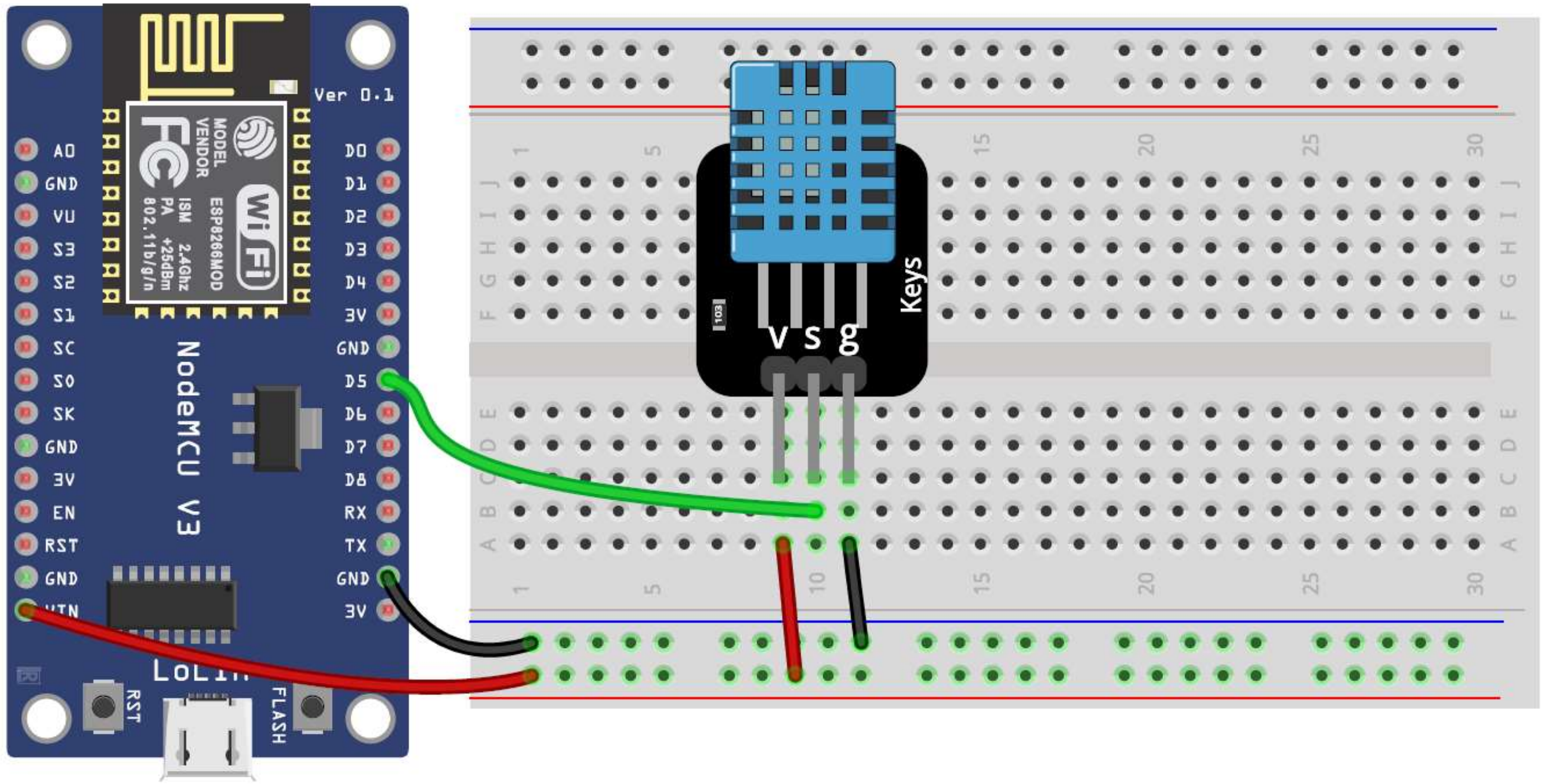
Sending Data to Firebase: DHT11 – Steps

4. The sensor Power pin connects to the VCC on NodeMCU.



Sending Data to Firebase: DHT11 – Steps

5. Wire up the sensor Data pin to the analog pin D5 on NodeMCU.



Sending Data to Firebase: Code

```
#include <ESP8266WiFi.h>                        // Include ESP8266WiFi library for WiFi features
#include <FirebaseESP8266.h>                    // Include FirebaseESP8266 library for Firebase integration
#include "DHT.h"                                // Include DHT sensor library
#define DHT_PIN D5                              // Define the digital pin connected to the DHT sensor
DHT dht(DHT_PIN, DHT11);                        // Initialize DHT sensor object with pin and sensor type

const char* WIFI_SSID = "iotlab";               // Define the WiFi network SSID
const char* WIFI_PASS = "hostiotlab";           // Define the WiFi network password

// Firebase Realtime Database URL and secret
const char* FIREBASE_HOST = "smart-home-6f8b2-default-rtdb.firebaseio.com";
const char* FIREBASE_AUTH = "pwfOlfS3G02LEVpJQEs8KqLp6sdh18ePRfzdJaba";

FirebaseData fbdo;                              // Define Firebase Data object
```

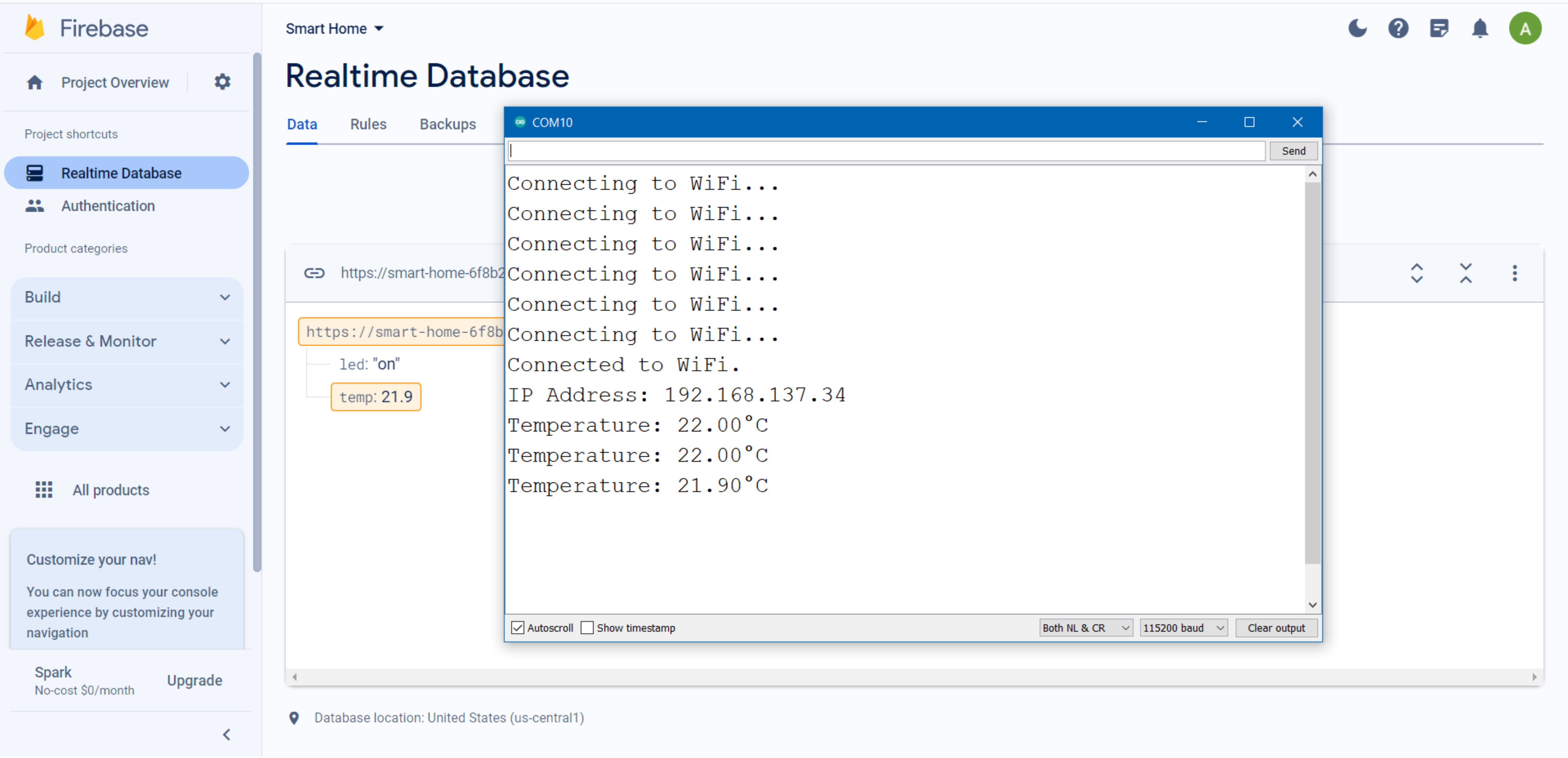
Sending Data to Firebase: Code

```
void setup() {
  Serial.begin(115200);                          // Start serial communication at 115200 baudrate
  WiFi.begin(WIFI_SSID, WIFI_PASS);              // Begin WiFi connection using SSID and password

  while(WiFi.status() != WL_CONNECTED){          // Check if the WiFi status is not connected
    delay(1000);                                 // Wait 1 second between WiFi connection checks
    Serial.println("Connecting to WiFi...");     // Print message indicating an attempt to connect to WiFi
  }

  Serial.println("Connected to WiFi.");          // Print message when WiFi connection is successful
  Serial.print("IP Address: ");                  // Print the label for the IP address
  Serial.println(WiFi.localIP());                // Print the assigned IP address

  dht.begin();                                   // Start DHT sensor
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // Initialize Firebase connection
  Firebase.reconnectWiFi(true);                  // Automatic reconnection to WiFi if connection is lost
}
```

Sending Data to Firebase: Code

```
void loop() {
  delay(2000);                            // Wait 2 seconds between measurements
  float temp = dht.readTemperature();     // Read temperature in Celsius

  if(Firebase.setFloat(fbdo, "/temp", temp)){  // Set temperature value in the Firebase under the "/temp" path
    Serial.print("Temperature: ");        // Print the label "Temperature: "
    Serial.print(temp);                   // Print the temperature value
    Serial.println("°C ");                // Print the unit "°C"
  }
  else                                    // If Firebase operation fails,
    Serial.println(fbdo.errorReason());   // Print the error reason
}
```

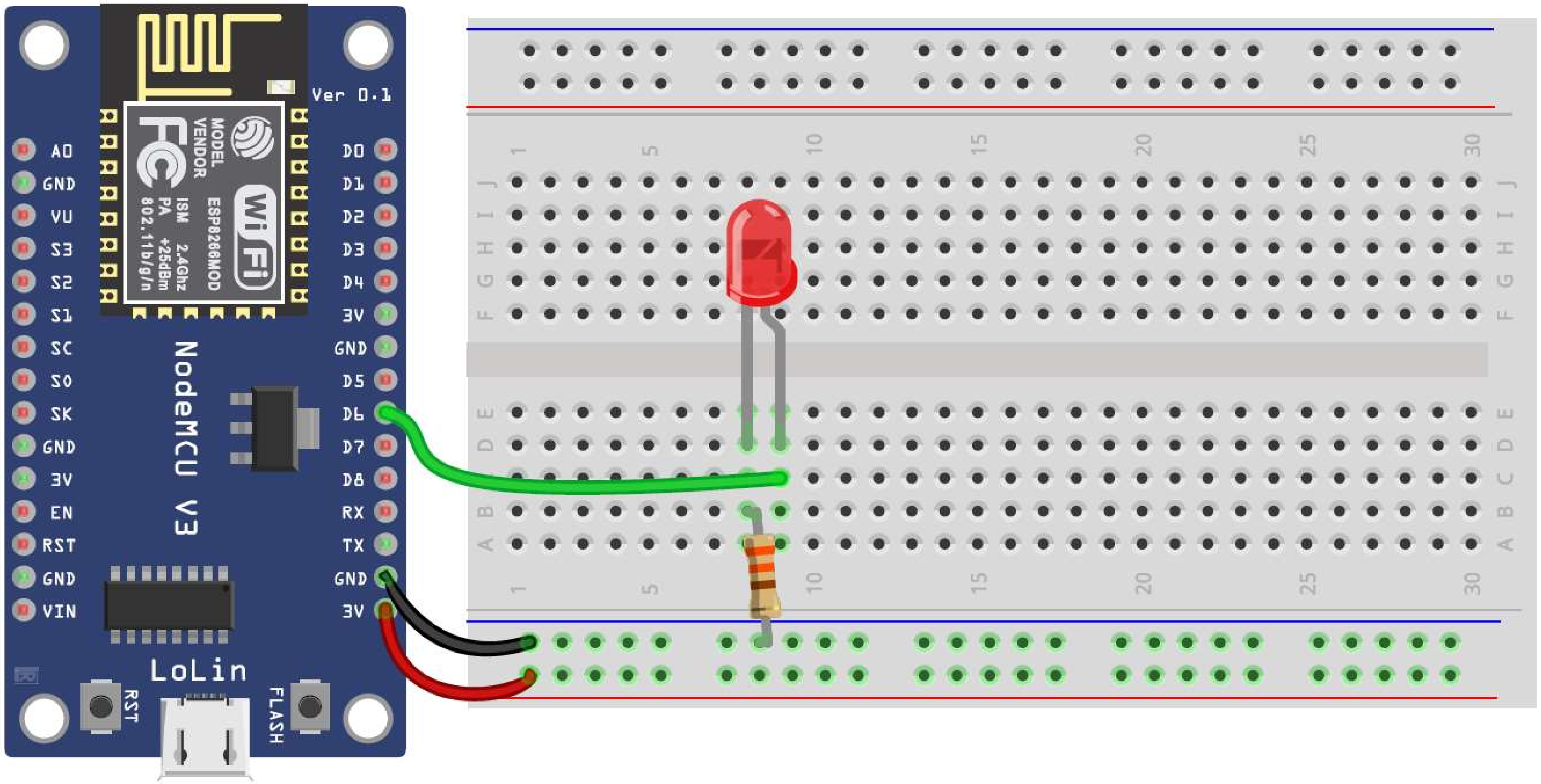
Sending Data to Firebase: Output



```
https://smart-home-6f8b
    led: "on"
    temp: 21.9
```

```
Connecting to WiFi...
Connecting to WiFi...
Connecting to WiFi...
Connecting to WiFi...
Connecting to WiFi...
Connecting to WiFi...
Connected to WiFi.
IP Address: 192.168.137.34
Temperature: 22.00°C
Temperature: 22.00°C
Temperature: 21.90°C
```

Reading Data from Firebase: Controlling an LED from Anywhere



Reading Data from Firebase: Code

```
#include <ESP8266WiFi.h>                        // Include ESP8266WiFi library for WiFi features
#include <FirebaseESP8266.h>                    // Include FirebaseESP8266 library for Firebase integration
#define LED_PIN D6                              // Define the LED pin

const char* WIFI_SSID = "iotlab";               // Define the WiFi network SSID
const char* WIFI_PASS = "hostiotlab";           // Define the WiFi network password

// Firebase Realtime Database URL and secret
const char* FIREBASE_HOST = "smart-home-6f8b2-default-rtdb.firebaseio.com";
const char* FIREBASE_AUTH = "pwfOlfS3G02LEVpJQEs8KqLp6sdh18ePRfzdJaba";

FirebaseData fbdo;                              // Define Firebase Data object
String led_status;                              // A variable to store LED status
```

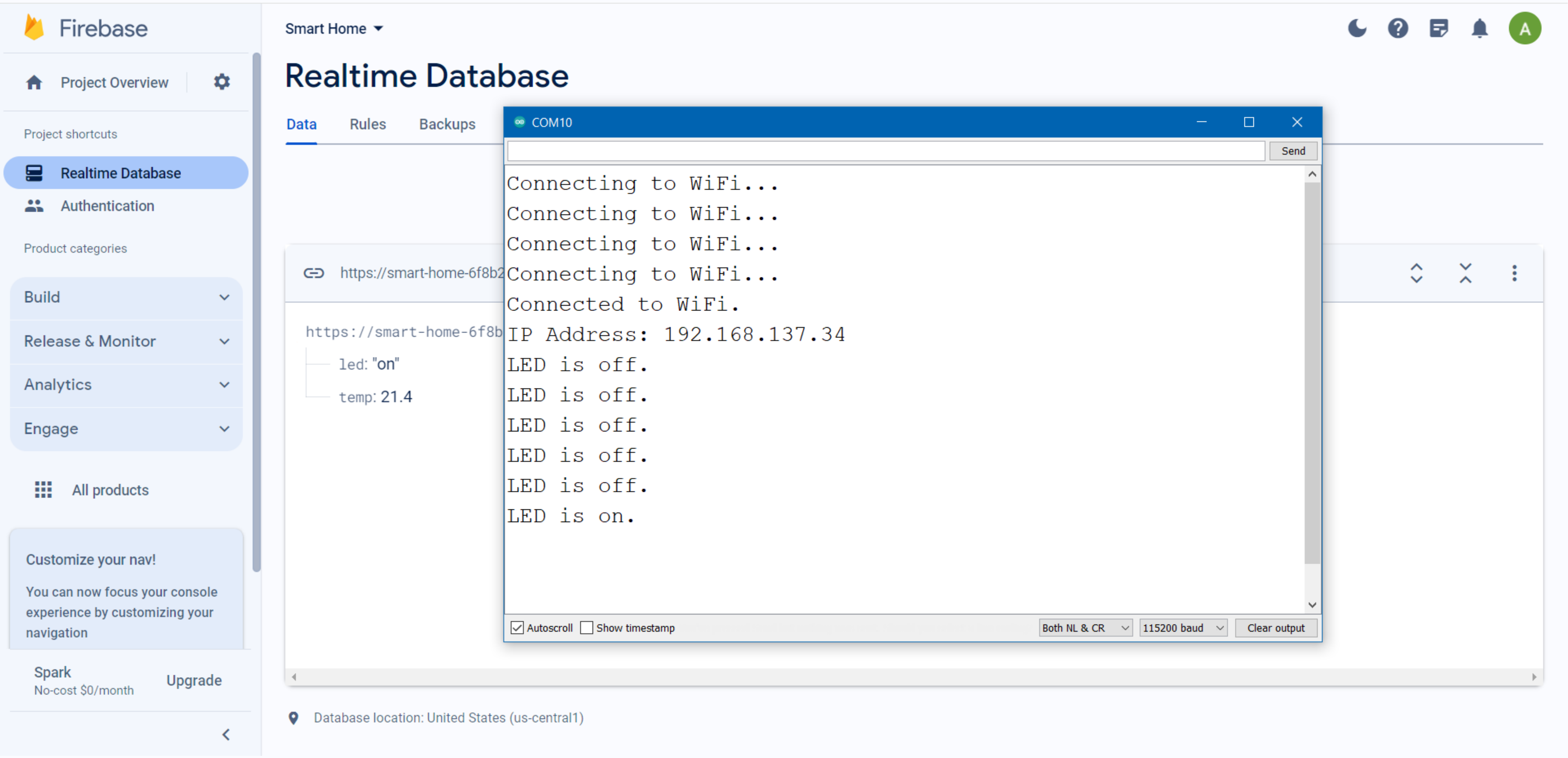
Reading Data from Firebase: Code

```
void setup() {
  Serial.begin(115200);                         // Start serial communication at 115200 baudrate
  WiFi.begin(WIFI_SSID, WIFI_PASS);             // Begin WiFi connection using SSID and password
  pinMode(LED_PIN, OUTPUT);                     // Initialize the pin D6 as an output

  while(WiFi.status() != WL_CONNECTED){         // Check if the WiFi status is not connected
    delay(1000);                                // Wait 1 second between WiFi connection checks
    Serial.println("Connecting to WiFi...");    // Print message indicating an attempt to connect to WiFi
  }

  Serial.println("Connected to WiFi.");         // Print message when WiFi connection is successful
  Serial.print("IP Address: ");                 // Print the label for the IP address
  Serial.println(WiFi.localIP());               // Print the assigned IP address

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // Initialize Firebase connection
  Firebase.reconnectWiFi(true);                 // Automatic reconnection to WiFi if connection is lost
}
```

Reading Data from Firebase: Code

```
void loop() {
  if(Firebase.getString(fbdo, "/led")){          // Try to get the LED status from the Firebase
    led_status = fbdo.to<String>();               // Get the LED status from the Firebase

    if(led_status == "on"){                       // If LED status is "on",
      digitalWrite(LED_PIN, HIGH);                // Turn on LED
      Serial.println("LED is on.");               // Print LED status
    }
    else if(led_status == "off"){                 // If LED status is "off",
      digitalWrite(LED_PIN, LOW);                 // Turn off LED
      Serial.println("LED is off.");              // Print LED status
    }
  }
  else                                            // If Firebase operation fails,
    Serial.println(fbdo.errorReason());           // Print the error reason

  delay(1000);                                    // Check Firebase for LED status every 1 second
}
```

Reading Data from Firebase: Output



```
https://smart-home-6f8b
  led: "on"
  temp: 21.4
```

```
Connecting to WiFi...
Connecting to WiFi...
Connecting to WiFi...
Connecting to WiFi...
Connected to WiFi.
IP Address: 192.168.137.34
LED is off.
LED is off.
LED is off.
LED is off.
LED is off.
LED is on.
```

Firebase ESP Client Library

- The Firebase ESP8266 library provides Firebase Realtime database and Firebase Cloud Messaging functions and supports only ESP8266 MCU.
  Firebase ESP8266 - GitHub
  Firebase ESP32 - GitHub
- If you use other Arduino devices or want to use more Firebase services included Firestore database, Firebase Storage, Google Cloud Storage and Cloud Functions for Firebase, use Firebase ESP Client library instead.
  Firebase ESP Client - GitHub

References and Tutorials

- ESP32 WiFi Tutorial & Library Examples (Arduino IDE)
- ESP8266 NodeMCU Access Point (AP) for Web Server
- ESP8266 NodeMCU: Getting Started with Firebase
- Firebase Realtime Database Arduino Library for ESP8266
- Firebase ESP8266 Client
- Firebase - Structure Your Database
- Simple Example to Store and Read Data from the Firebase