

IoTLab2 ▾


Realtime Database

Data

Rules


Backups

Usage

 Extensions



Protect your Realtime Database resource

 <https://iotlab2-ca830-default-rtdb.firebaseio.com>

<https://iotlab2-ca830-default-rtdb.firebaseio.com/>

humidity: 31.872451782226562

▼ light_info

light_b: 255

light_col: 1

light_g: 0

light_r: 0

light_row: 0

temperature: 30.31153678894043

update_light: false

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
○ chase@raspberrypi:~/lab2 $ node app.js
Settings file RTIMULib.ini loaded
Using fusion algorithm RTQF
min/max compass calibration not in use
Ellipsoid compass calibration not in use
Accel calibration not in use
LSM9DS1 init complete
Temperature: 27.77717399597168°C, Humidity: 36.78681182861328%
Temperature: 27.941268920898438°C, Humidity: 34.96751022338867%
Temperature: 28.032434463500977°C, Humidity: 35.13697052001953%
Temperature: 28.068899154663086°C, Humidity: 34.96430969238281%
Temperature: 28.21476173400879°C, Humidity: 36.16652297973633%
Temperature: 28.269460678100586°C, Humidity: 35.44071960449219%
Temperature: 28.232994079589844°C, Humidity: 34.18415069580078%
Temperature: 28.397090911865234°C, Humidity: 34.87798309326172%
Temperature: 28.542953491210938°C, Humidity: 34.49429702758789%
Temperature: 28.561185836791992°C, Humidity: 34.52627182006836%
Light updated: [0, 1] to [0, 0, 255]
Temperature: 28.670581817626953°C, Humidity: 35.11138916015625%
Temperature: 28.72528076171875°C, Humidity: 34.48470687866211%
```

Code:

```
var firebase = require('firebase/app');
require('firebase/analytics');
const { getDatabase, ref, onValue, set, update, get } =
require('firebase/database');
var nodeimu = require( '@trbll/nodeimu' );
var IMU = new nodeimu.IMU( );
var sense = require( '@trbll/sense-hat-led' );

// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyC2JWGJOylB21xTbAfTgzpA0gC2ceIG8fY",
  authDomain: "iotlab2-ca830.firebaseio.com",
  projectId: "iotlab2-ca830",
  storageBucket: "iotlab2-ca830.appspot.com",
  messagingSenderId: "868635210220",
  appId: "1:868635210220:web:11b13892607ff5ebfb50c9",
```

```

    measurementId: "G-NB06Q0QTH2"
  };

  // Initialize Firebase
  const app = firebase.initializeApp(firebaseConfig);
  const database = getDatabase();

  // Function to initialize default values in the database
  function initializeDefaultValues() {
    const defaultValues = {
      temperature: 0,
      humidity: 0,
      update_light: false,
      light_info: {
        light_r: 0,
        light_g: 0,
        light_b: 0,
        light_row: 0,
        light_col: 0
      }
    };

    get(ref(database)).then((snapshot) => {
      if (!snapshot.exists()) {
        set(ref(database), defaultValues).then(() => {
          console.log("Default values set in Firebase database.");
        }).catch((error) => {
          console.error("Error setting default values in Firebase
database:", error);
        });
      }
    });
  }

  // Call initializeDefaultValues at the start of your application
  initializeDefaultValues();

  // Function to update temperature and humidity
  function updateSensorData() {
    IMU.getValue((error, data) => {

```

```

    if (error) {
        console.log(error);
        return;
    }

    const { temperature, humidity } = data;
    console.log(`Temperature: ${temperature}°C, Humidity: ${humidity}%`);

    const updates = {};
    updates['/temperature'] = temperature;
    updates['/humidity'] = humidity;

    update(ref(database), updates);
  });
}

// Listen for updates
const lightUpdateRef = ref(database, 'update_light');
onValue(lightUpdateRef, (snapshot) => {
    const updateLight = snapshot.val();
    if (updateLight) {
        get(ref(database, 'light_info')).then((snapshot) => {
            if (snapshot.exists()) {
                const { light_r, light_g, light_b, light_row, light_col } =
snapshot.val();
                sense.setPixel(parseInt(light_row), parseInt(light_col),
parseInt(light_r), parseInt(light_g), parseInt(light_b));
                console.log(`Light updated: [${light_row}, ${light_col}] to
[${light_r}, ${light_g}, ${light_b}]`);

                set(ref(database, 'update_light'), false);
            }
        });
    }
});

// Periodically update sensor data
setInterval(updateSensorData, 5000); // Every 5 seconds

```