

# Anbindung HLF-Messwagen an Cloud-Dienste als Echtzeit-Datenquelle mit einem ESP32

IT-Bildungsoffensive TP1 Handlungsfeld «Lernfabrik»

## Änderungsnachweis

**Autor:** Leonie Zellweger  
**Erstellt am:** 29.02.2024

Version	Änderungsgrund	Kurz-Z.	Datum
0	Text	Kürzel	TT.MM.JJ
0	Text	Kürzel	TT.MM.JJ

## Kurzbeschreibung und Bezug zur IT-Bildungsoffensive

Die Aufgabe dieser aufbauenden Lerneinheit besteht darin, die Daten eines bereits selbstentwickelten Temperatursensors in eine Cloud zu übertragen. Zusätzlich sollen die Temperaturdaten in Echtzeit in einer Webanwendung dargestellt bzw. abgefragt werden können. Dazu wird ein bestehendes MicroPython Programm ergänzt und ein Clouddienst zur Datenspeicherung kennengelernt und eingerichtet. Das Blockschaltbild ist auf Abbildung 1 ersichtlich.

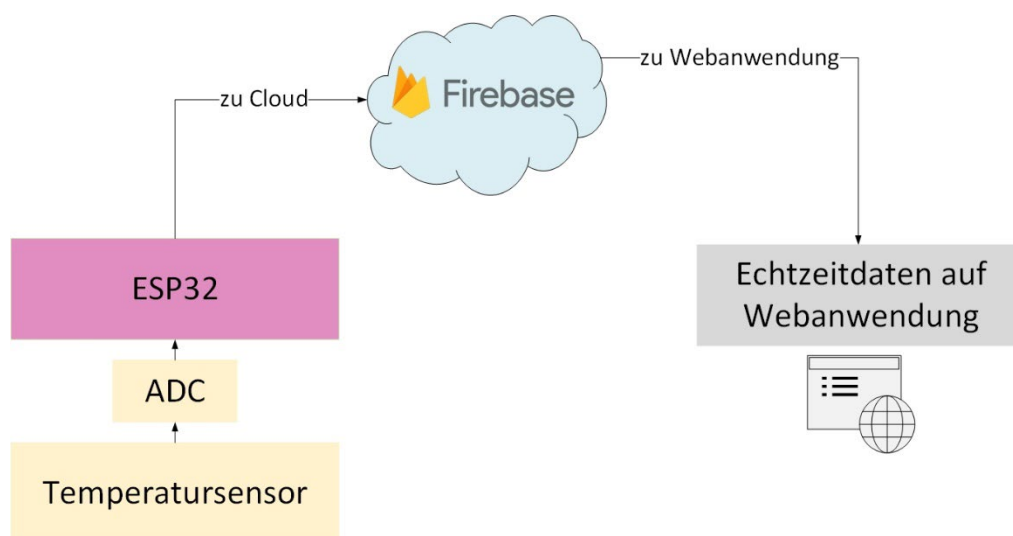


Abbildung 1: Blockschaltbild

## Inhaltsverzeichnis

Kurzbeschreibung und Bezug zur IT-Bildungsoffensive.....	1
Rahmenbedingungen .....	3
1. Was ist Firebase .....	5
1.1 Projekt in Firebase erstellen .....	5
1.2 Projekteinstellungen anpassen.....	8
1.3 Realtime Database erstellen .....	11
2. index.html anpassen .....	12
3. MicroPython Skript anpassen.....	15
4. Daten senden/empfangen und in Cloud speichern .....	16
5. Firebase Projekt löschen .....	16

## Abbildungsverzeichnis

Abbildung 1: Blockschaltbild.....	1
Abbildung 2: Projekt erstellen.....	6
Abbildung 3: Projektnamen eingeben .....	6
Abbildung 4: Google Analytics aktivieren .....	7
Abbildung 5: Projekt wird erstellt .....	7
Abbildung 6: Google Firebase Oberfläche .....	8
Abbildung 7: Authentication.....	8
Abbildung 8: Authentication mit E-Mail und Passwort.....	9
Abbildung 9: Anmeldemethode aktivieren.....	9
Abbildung 10: Nutzer hinzufügen .....	10
Abbildung 11: E-Mail/Passwort-Nutzer hinzufügen.....	10
Abbildung 12: Realtime Database erstellen .....	11
Abbildung 13: Realtime Database erstellen - Schritt eins .....	11
Abbildung 14: Realtime Database erstellen – Schritt zwei.....	11
Abbildung 15: App hinzufügen .....	12
Abbildung 16: Web-App erstellen.....	12
Abbildung 17: Konfiguration Web-App .....	13
Abbildung 18: Konfiguration in index.html .....	14
Abbildung 19: Benutzer definieren .....	14
Abbildung 20: Konfiguration Python-Skript.....	15
Abbildung 21: Projekt löschen .....	16

## Rahmenbedingungen

- Studiengänge: alle Studiengänge, die die Grundlagen der Elektronik und Informatik unterrichten
- Semester: 3. oder 4. Semester
- Art der Durchführung: Projektarbeit mit Einführung, Selbststudium, Teamarbeit (2-4 Studierende)
- Umfang:
  - 4 Lektionen à 45min
  - 4 Stunden Selbststudium
  - Leistungsnachweis: bewertete Präsentation
- Sprache der Materialien: Deutsch oder Englisch
- Voraussetzungen:
  - Grundlagen der Elektrotechnik/Elektronik/Informatik
  - Modul 12 im Fach HLF-B abgeschlossen
- Benötigte Vorarbeiten:
  - Schaltungsentwicklung Temperaturmessung und Bestückung
  - Einarbeitung in MicroPython-Programm und Erweiterung ADC-Programm
- Benötigte Ressourcen:
  - HLF
  - ESP32
  - Versuchssprint
  - Kleinmaterial Elektronik
- Anforderungen an Verfügbarkeit von Anlagen und Support:
  - Hybride Lernfabrik Buchs: 4 Lektionen à 45min, Präsenz

## Lernziele

- Die Studierenden können ein MicroPython Programm erweitern und debuggen.
- Die Studierenden können einen Clouddienst einrichten und diesen anwenden.
- Die Studierenden kennen den Übertragungspfad vom ADC des Messwagens – Carrier bis zur Webanwendung.

## Detaillierte Beschreibung von Inhalt und Ablauf

### Vorarbeit (Modul 12, HLF-B):

Die Studierenden werden sich im Selbststudium mit der Schaltungsentwicklung und Bestückung einer Temperaturmesselektronik befassen. Zusätzlich wird von ihnen verlangt, dass sie sich in das MicroPython-Programm einarbeiten und die Erweiterung für den ADC-Teil vornehmen oder sich zumindest eine Lösung dafür überlegen. Als weiterer Arbeitsschritt wird von ihnen erwartet, dass sie sich auf [Firebase](#) mit einem Google-Account registrieren und anmelden.

### Lektion 1 (Modul 12, HLF-B):

Während der ersten Lektion sollen die Studierenden ihr komplettes Messsystem zusammenbauen und in Betrieb nehmen. Dazu werden sie schrittweise vorgehen, angefangen mit dem Akku und übergehend dann zum ESP32, sowie der Temperaturmesselektronik.

### Lektion 2+3 (Modul 13, HLF-B):

In diesen beiden Lektionen werden sich die Studierenden mit dem Clouddienst Firebase beschäftigen und diesen einrichten. Des Weiteren soll das MicroPython Programm erweitert werden, sodass die Temperaturdaten in die Cloud gesendet werden können.

### Lektion 4 (Modul 12, HLF-B):

In der letzten Lektion sollen mit dem Messsystem auf dem Förderband Messungen durchgeführt werden. Ziel ist die Analyse und Dokumentation der Messdaten.

## Lernmaterialien des Moduls

### Vorlesungsmaterialien

- Beschreibung als PDF

### Übungsmaterialien

- Sourcecode zum Ergänzen

### Prüfungsmaterialien

- Keine
- Bewertete Präsentation

# 1. Was ist Firebase

Firebase ist eine der führenden Entwicklungs-Plattformen im Bereich der App-Entwicklung für mobile und Webanwendungen und gehört zum Google Konzern.

Es stellt verschiedene Funktionen zur Verfügung, wie beispielsweise Datenbanken, Authentifizierung, Analysefunktionen und vieles mehr.

In diesem Kurs wird eine Firebase Realtime Database instanziiert, die dazu genutzt wird, Temperaturwerte von einem ESP32 entgegenzunehmen und sie zu speichern. Firebase Realtime Database ist eine Cloud-gehostete Datenbank und ermöglicht Datenspeicherung und Echtzeit-Datensynchronisierung.

Nachfolgend wird beschrieben, wie ein neues Projekt erzeugt wird und welche Einstellungen für die Übertragung der Daten gemacht werden müssen.

Es kann sein, dass die Illustrationen in diesem Bericht von denen in der neuesten Version von Firebase abweichen, da die Anwendung kontinuierlich angepasst und weiterentwickelt wird (Ivanov, 2022).

## 1.1 Projekt in Firebase erstellen

Um Firebase nutzen zu können, wird als erstes ein Google-Account benötigt. Falls Sie keinen besitzen, können Sie diesen in wenigen Minuten selbst erstellen. Wenn Sie bereits einen eigenen Google-Account besitzen, verwenden Sie diesen. Die Realtime Database Anwendung ist für unsere Anwendung kostenlos. Sie kann aber je nach Nutzung und Datenmenge Kosten verursachen.

1. Suchen Sie im Internet nach «*Google Firebase*» und gehen Sie dann auf die Startseite.
2. Wenn Sie auf «*Get started*» klicken, werden Sie zum folgenden Fenster auf Abbildung 2 weitergeleitet.
3. Jetzt wählen Sie «*Projekt erstellen*» aus (rot umrandet).

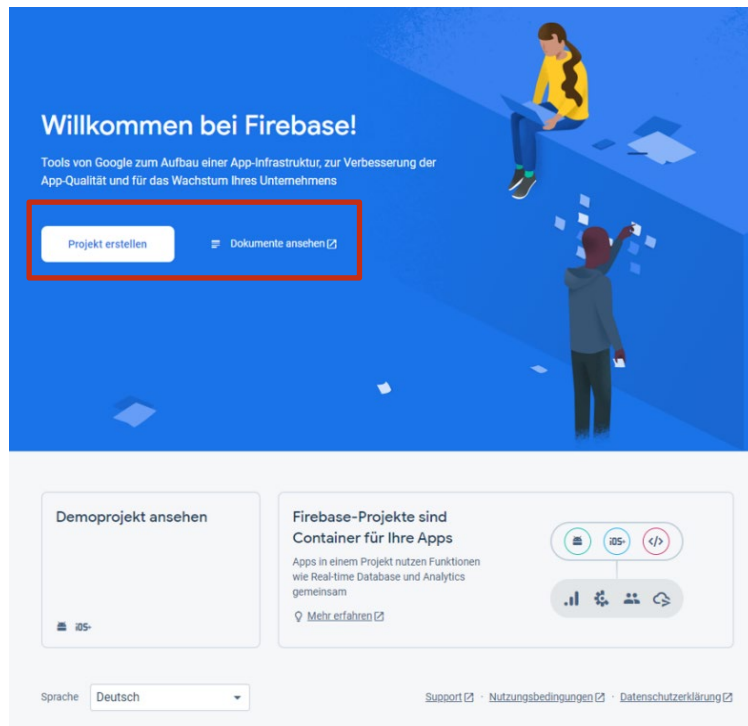


Abbildung 2: Projekt erstellen

4. Geben Sie einen passenden Projektnamen ein (4) (In diesem Beispiel wurde «ITBO-HLF» gewählt).
5. Setzen Sie zweimal einen Haken (5), bevor Sie mit «Weiter» weiterfahren können.

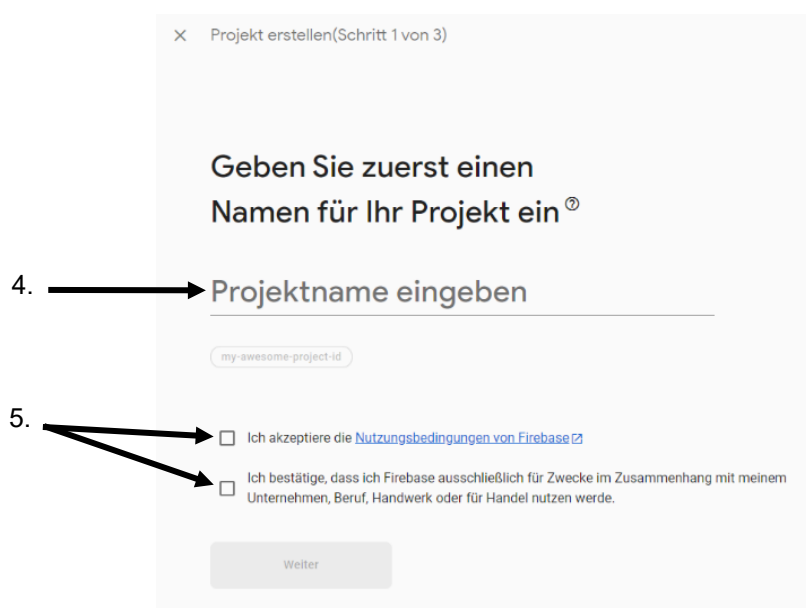


Abbildung 3: Projektnamen eingeben

6. Deaktivieren Sie «*Google Analytics*» für dieses Projekt und drücken Sie auf «*Projekt erstellen*».

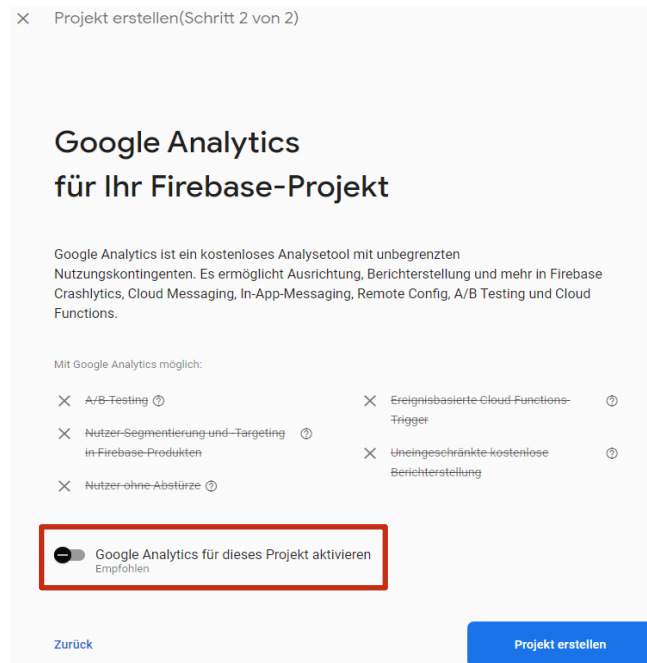


Abbildung 4: Google Analytics aktivieren

7. Nun vergehen ein paar Sekunden, bis das Projekt erstellt wurde. Sobald es bereitsteht, wählen Sie «*weiter*».



Abbildung 5: Projekt wird erstellt

8. Gratulation, Sie haben nun erfolgreich ein Firebase-Projekt erstellt! In einem weiteren Schritt geht es darum, die Projekteinstellungen so anzupassen, dass es für Ihre Anwendung mit dem ESP32 kompatibel ist.

## 1.2 Projekteinstellungen anpassen

9. Die Oberfläche von Google Firebase sieht folgendermassen aus:
- Auf der linken Seite sind die Projekteinstellungen, sowie die verschiedenen Produktkategorien zu finden.
  - In diesem Kurs werden «*Authentication*» und «*Realtime Database*» benötigt, die sich im Subordner von «*Entwickeln*» befinden.

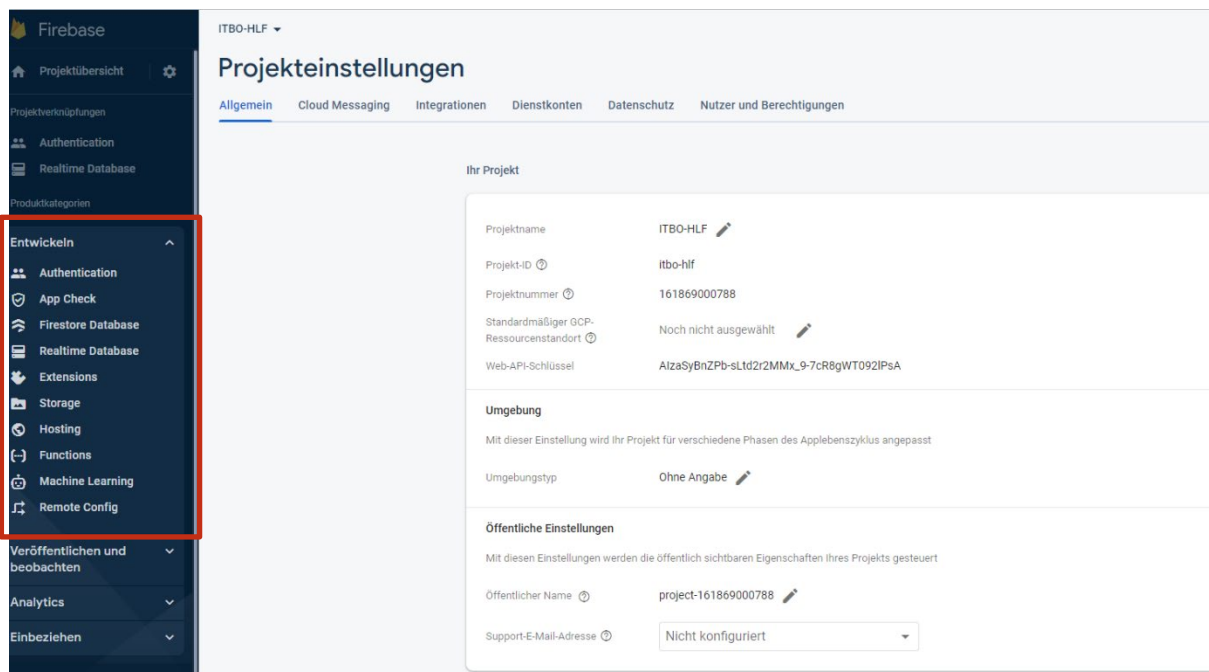


Abbildung 6: Google Firebase Oberfläche

10. Um nun die Daten des Temperatursensors zu empfangen, müssen ein paar weitere Einstellungen vorgenommen werden.
11. Wählen Sie > *Entwickeln* > *Authentication*. Eventuell müssen Sie noch auf «*Los gehts*» klicken (Abbildung 7).

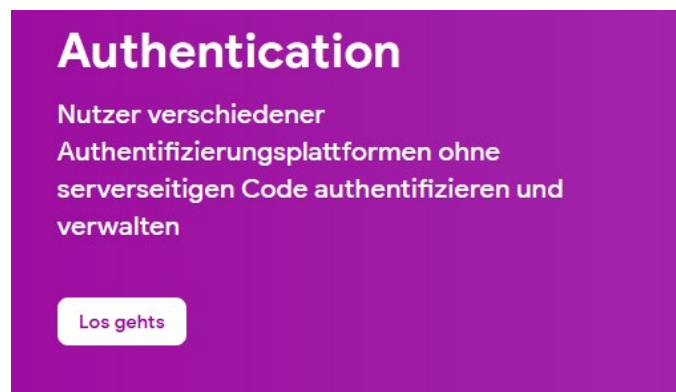


Abbildung 7: Authentication

12. Es öffnet sich ein Fenster unter dem ersten Reiter «*Sign-in-method*». Wählen Sie unter Native Anbieter «*E-Mail-Adresse/Passwort aus*».

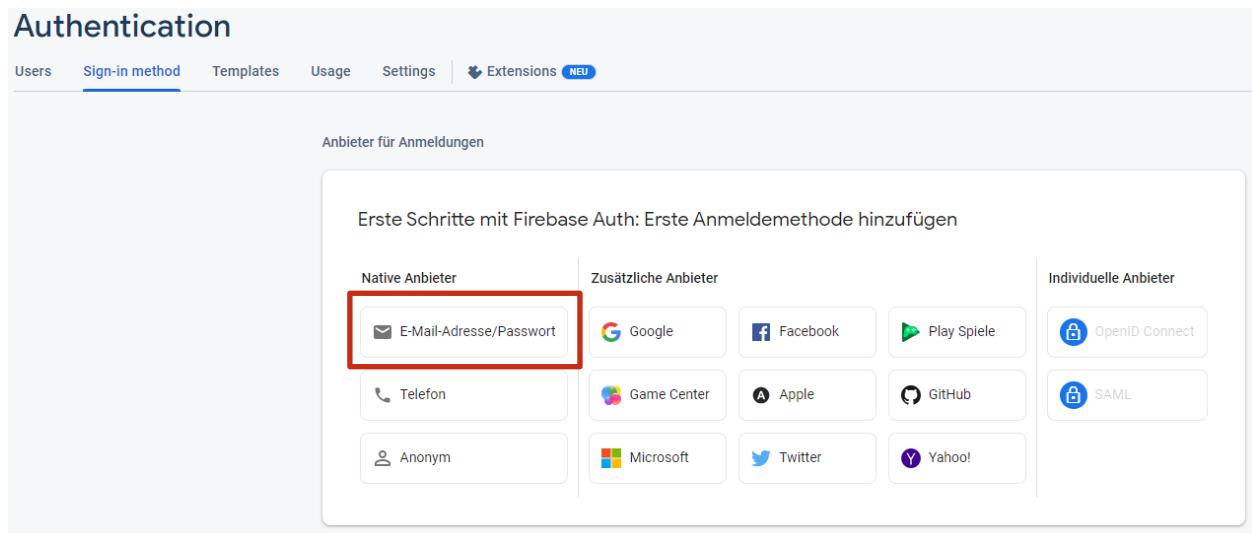


Abbildung 8: Authentication mit E-Mail und Passwort

13. Aktivieren Sie die Anmeldemethode und klicken Sie auf «*speichern*» (Abbildung 9).

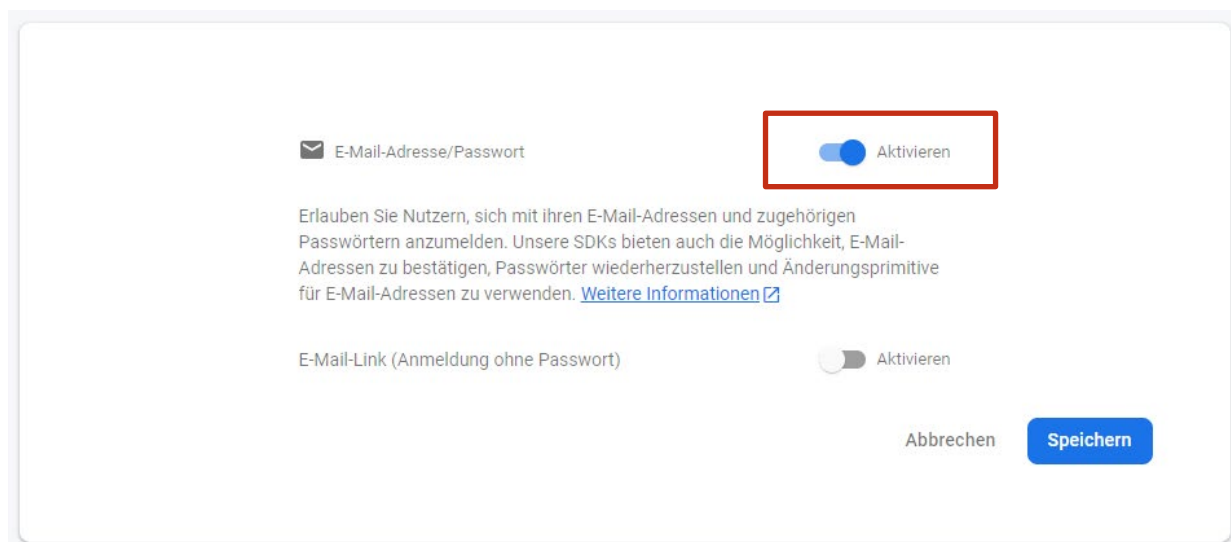


Abbildung 9: Anmeldemethode aktivieren

14. Gehen Sie nun zum Reiter «Users» und wählen Sie «Nutzer hinzufügen» (Abbildung 10).

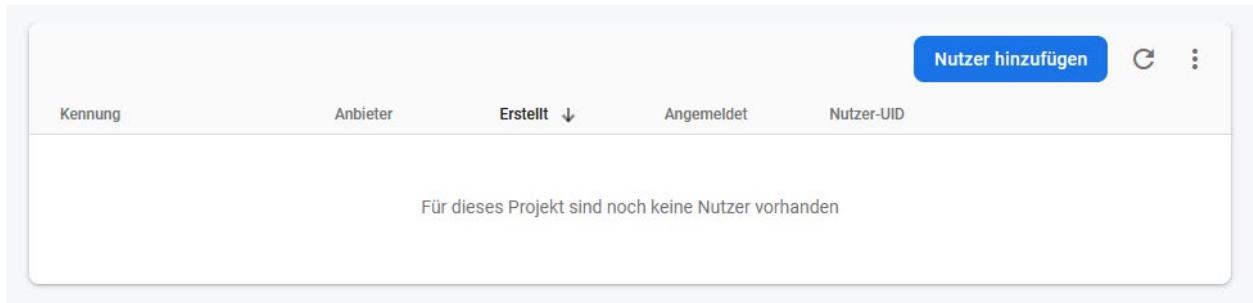


Abbildung 10: Nutzer hinzufügen

15. Es erscheint folgendes Fenster, wo Sie beispielsweise die nachfolgende E-Mail-Adresse und Passwort eingeben können. Sie dürfen die E-Mail und das Passwort auch selbst wählen. Am Schluss klicken Sie auf «Nutzer hinzufügen».

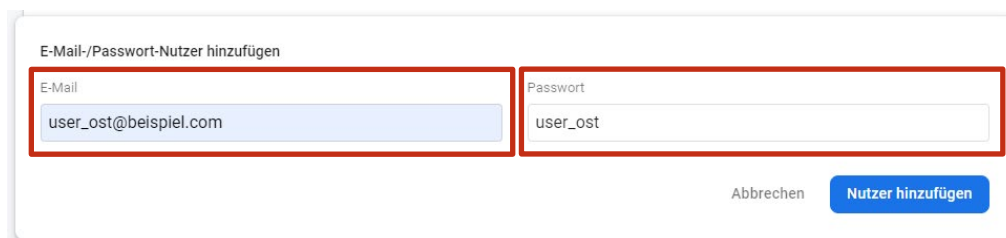


Abbildung 11: E-Mail/Passwort-Nutzer hinzufügen

### 1.3 Realtime Database erstellen

16. Um Ihre eigene Database zu erstellen, können Sie links bei den Produktkategorien «*Realtime Database*» auswählen und dann auf «*Datenbank erstellen*» drücken (Abbildung 12).

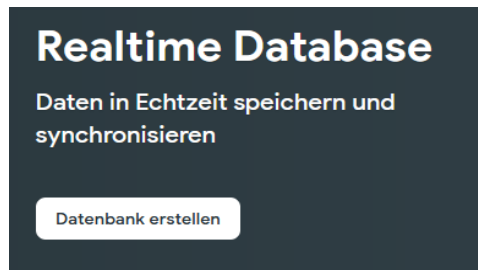


Abbildung 12: Realtime Database erstellen

17. Es erscheint ein neues Fenster mit der Auswahl des Standortes für die Realtime-Daten. Wählen Sie «*Belgien (europe-west1)*» aus (Abbildung 13).

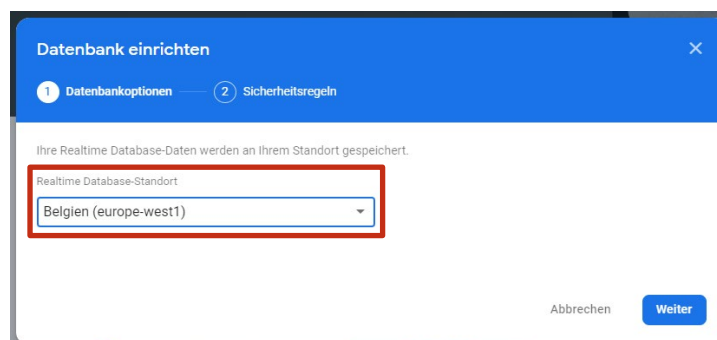


Abbildung 13: Realtime Database erstellen - Schritt eins

18. In einem zweiten Schritt legen Sie die Sicherheitsregeln fest. Für unsere Art von Anwendung genügt es, wenn «*Im Testmodus starten*» ausgewählt und mit «*aktivieren*» bestätigt wird.

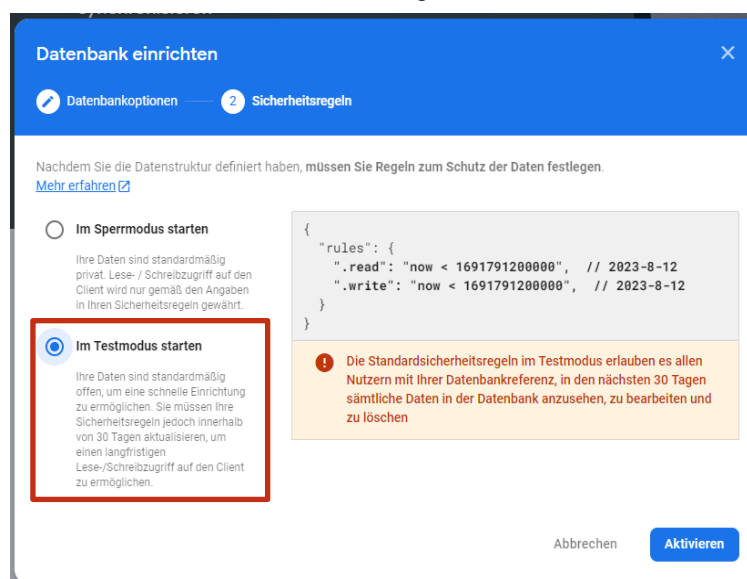


Abbildung 14: Realtime Database erstellen – Schritt zwei

## 2. index.html anpassen

19. Um die Datenübertragung vom Temperatursensor zur Datenbank zu ermöglichen, müssen Sie im HTML-Code die Konfiguration anpassen. Dies machen Sie am besten, indem Sie zuerst eine «App» erstellen. Gehen Sie dazu in die Projekteinstellungen und scrollen Sie unter dem Reiter «Allgemein» ganz nach unten. Dort ist folgende Auswahl wie auf Abbildung 15 zu finden. Wählen Sie das Symbol des Web-Apps (rot umrandet) aus.

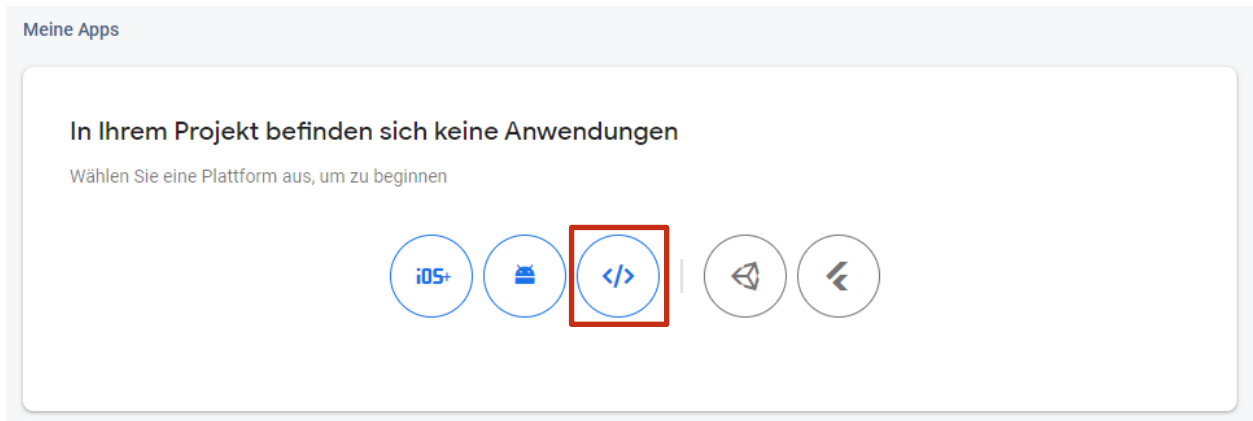


Abbildung 15: App hinzufügen

20. Wählen Sie wieder einen Namen für die Web-App und klicken Sie auf «App registrieren».



Abbildung 16: Web-App erstellen

21. In einem zweiten Schritt wird die Firebase SDK hinzugefügt. Sie können die Auswahl so belassen und auf «Weiter zur Konsole» klicken (Abbildung 17).

✓ App registrieren

**2** Firebase SDK hinzufügen

npm verwenden     <script>-Tag verwenden

Wenn Sie schon [npm](#) und einen Module Bundler wie [Webpack](#) oder [Rollup](#) verwenden, können Sie mit dem folgenden Befehl das neueste SDK installieren ([weitere Informationen](#)):

```
$ npm install firebase
```

Initialisieren Sie anschließend Firebase und verwenden Sie die SDKs für die gewünschten Produkte.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// 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: "AIzaSyBnZPb-sLtd2r2MMx_9-7cR8gWT0921PsA",
  authDomain: "itbo-hlf.firebaseio.com",
  databaseURL: "https://itbo-hlf-default-rtdb.europe-west1.firebaseio.com",
  projectId: "itbo-hlf",
  storageBucket: "itbo-hlf.appspot.com",
  messagingSenderId: "161869000788",
  appId: "1:161869000788:web:0bee3106cc79d56f7f7dc1",
  measurementId: "G-VRKDHVJCB6"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

**Hinweis:** Bei dieser Option wird das [modulare JavaScript SDK](#) verwendet, da dieses SDK kleiner ist.

Hier finden Sie weitere Informationen zu Firebase für das Web: [Startleitfaden](#), [Web SDK API-Referenz](#), [Beispiele](#)

**Weiter zur Konsole**

Abbildung 17: Konfiguration Web-App

22. Mit dieser Einstellung können Sie jetzt die HTML-Datei «*index.html*» öffnen und den Abschnitt:

```
// Hier Konfiguration aus der Firebase Konsole einfügen
```

mit Ihren Daten überschreiben. Dazu können Sie in der Konsole bei den Projekteinstellungen von Firebase beim Reiter «Allgemein» ganz unten die Daten kopieren und im «*index.html*» einfügen.

23. Am Schluss sieht das dann so aus, wie auf Abbildung 18.

```
// -----  
// Hier Konfiguration aus der Firebase Konsole einfügen  
const firebaseConfig = {  
  apiKey: "AIzaSyBnZPb-sLtd2r2MMx_9-7cR8gWT092lPsA",  
  authDomain: "itbo-hlf.firebaseio.com",  
  databaseURL: "https://itbo-hlf-default-rtdb.europe-west1.firebaseio.com",  
  projectId: "itbo-hlf",  
  storageBucket: "itbo-hlf.appspot.com",  
  messagingSenderId: "161869000788",  
  appId: "1:161869000788:web:0bee3106cc79d56f7f7dc1",  
};
```

Abbildung 18: Konfiguration in index.html

24. Des Weiteren müssen Sie im Code Ihre E-Mail-Adresse und das Passwort anlegen, welches Sie bereits in Ihrem Firebase-Projekt vergeben haben (Abbildung 19).

```
// -----  
// Hier Email / Passwort des angelegten Accounts in der Firebase Konsole einfügen  
const email = "user_ost@beispiel.com";  
const password = "user_ost";  
const app = initializeApp(firebaseConfig);  
const auth = getAuth(app);  
const database = getDatabase(app);
```

Abbildung 19: Benutzer definieren

### 3. MicroPython Skript anpassen

Auch im MicroPython Skript auf Ihrem ESP32 müssen Sie im Code ein paar Ergänzungen machen. Versorgen Sie dazu das ESP32 mit Spannung und laden und öffnen Sie die App "Thonny" ([Thonny Python IDE for beginners](#)) und dann das MicroPython Skript "main.py". Suchen Sie die Stelle mit der Konfiguration und bearbeiten Sie diese gleich wie im vorigen Kapitel. Auf Abbildung 20 sind die Anpassungen ersichtlich, die Sie machen müssen.

Schauen Sie sich bitte das Skript an und versuchen Sie zu verstehen, wie der Code aufgebaut ist und was damit gemacht wird.

```
# Firebase
API_KEY = "AIzaSyBnZPb-sLtd2r2MMx_9-7cR8gWT092lPsA"
DATABASE_URL = "https://itbo-hlf-default-rtdb.europe-west1.firebaseio.com"
EMAIL = "user_ost@beispiel.com"
PASSWORD = "user_ost"
```

Abbildung 20: Konfiguration Python-Skript

## 4. Daten senden/empfangen und in Cloud speichern

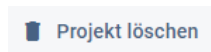
Jetzt ist alles eingerichtet, um die Daten des Temperatursensors zu empfangen und in der Realtime-Database zu speichern.

Führen Sie nun folgende Schritte durch:

- Öffnen Sie die angepasste Version von «*index.html*» im Browser. Sobald die Daten gesendet werden, sollten Sie den Datenplot und die Tabelle mit den Daten sehen.
- Öffnen Sie in ihrem Firebase-Projekt die Realtime Database und bleiben Sie im Reiter «*Daten*». Hier sollten Sie dann die ankommenden Daten sehen können.
- Wenn Sie jetzt im MicroPython Programm auf «*Run*» klicken, dann sollten Sie sehen können, wie die Daten in die Cloud und in die Webansicht geschrieben und gespeichert werden.
- Falls alles funktioniert, dann gratulieren wir Ihnen herzlich zum Erfolg! Sehen Sie sich die Daten an und überprüfen Sie diese auch auf Richtigkeit. Danach können Sie Tests durchführen und die Daten speichern und anzeigen. Experimentieren Sie mit der Anwendung.
- Falls etwas noch nicht funktionieren sollte, dann überprüfen Sie nochmals alle Einstellungen und auch Ihre modifizierten Skripte und Dateien. Vielleicht hat sich ein Schreib-, oder Kopierfehler eingeschlichen.
- Sollten Sie den Fehler nicht finden, dann bitten Sie den Kursleiter um Hilfe.

## 5. Firebase Projekt löschen

Nach Beendigung der Aufgabe kann das Projekt wieder gelöscht werden. Gehen Sie dazu in die Projekteinstellungen und scrollen Sie ganz nach unten. In der Mitte sollte «*Projekt löschen*» zur Auswahl stehen. Wählen Sie es aus.



Danach öffnet sich ein Fenster, bei dem Sie allen Bedingungen mit einem Haken zustimmen müssen, bevor Sie das Projekt endgültig mit «*Projekt löschen*» löschen können (Abbildung 21).

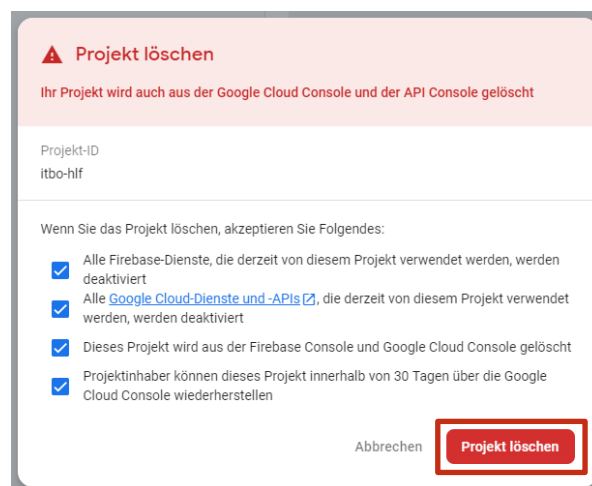


Abbildung 21: Projekt löschen