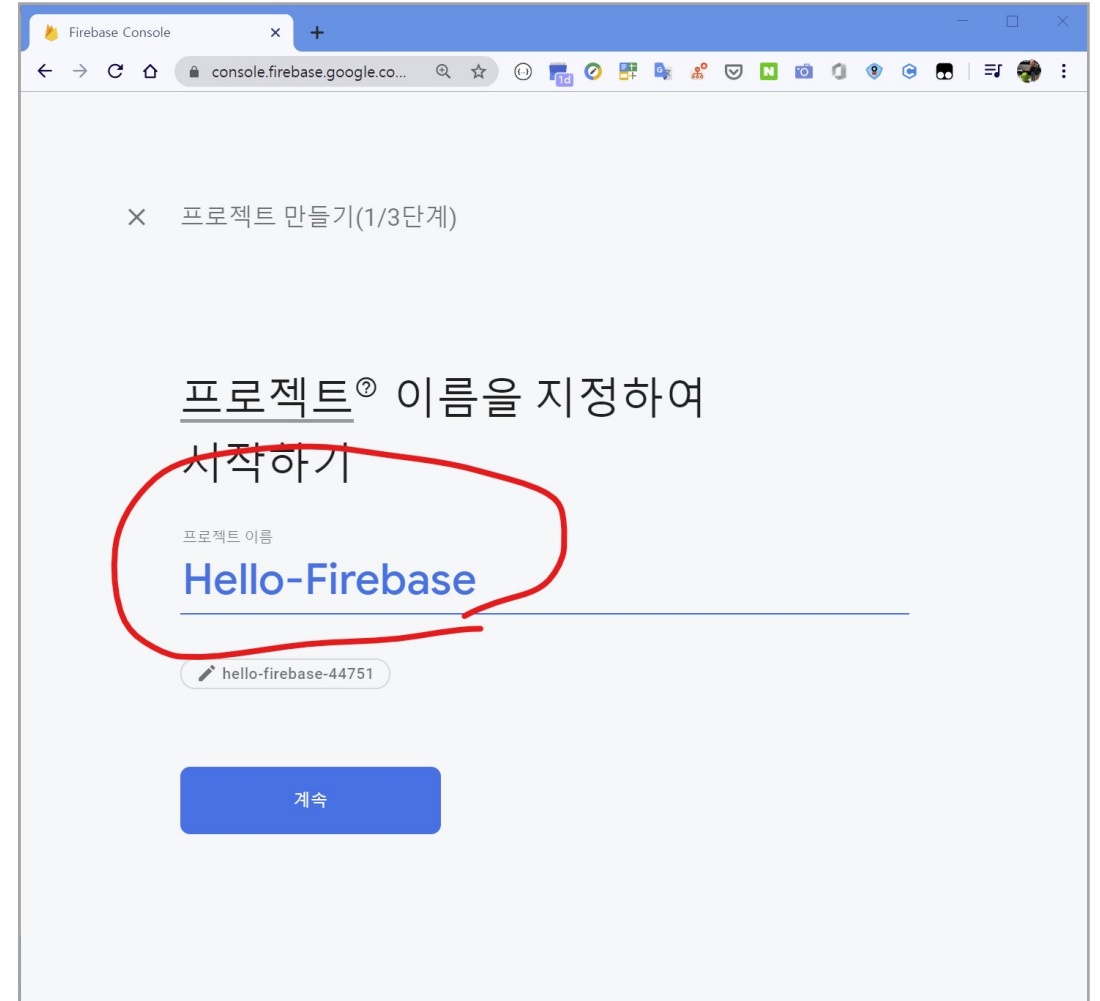
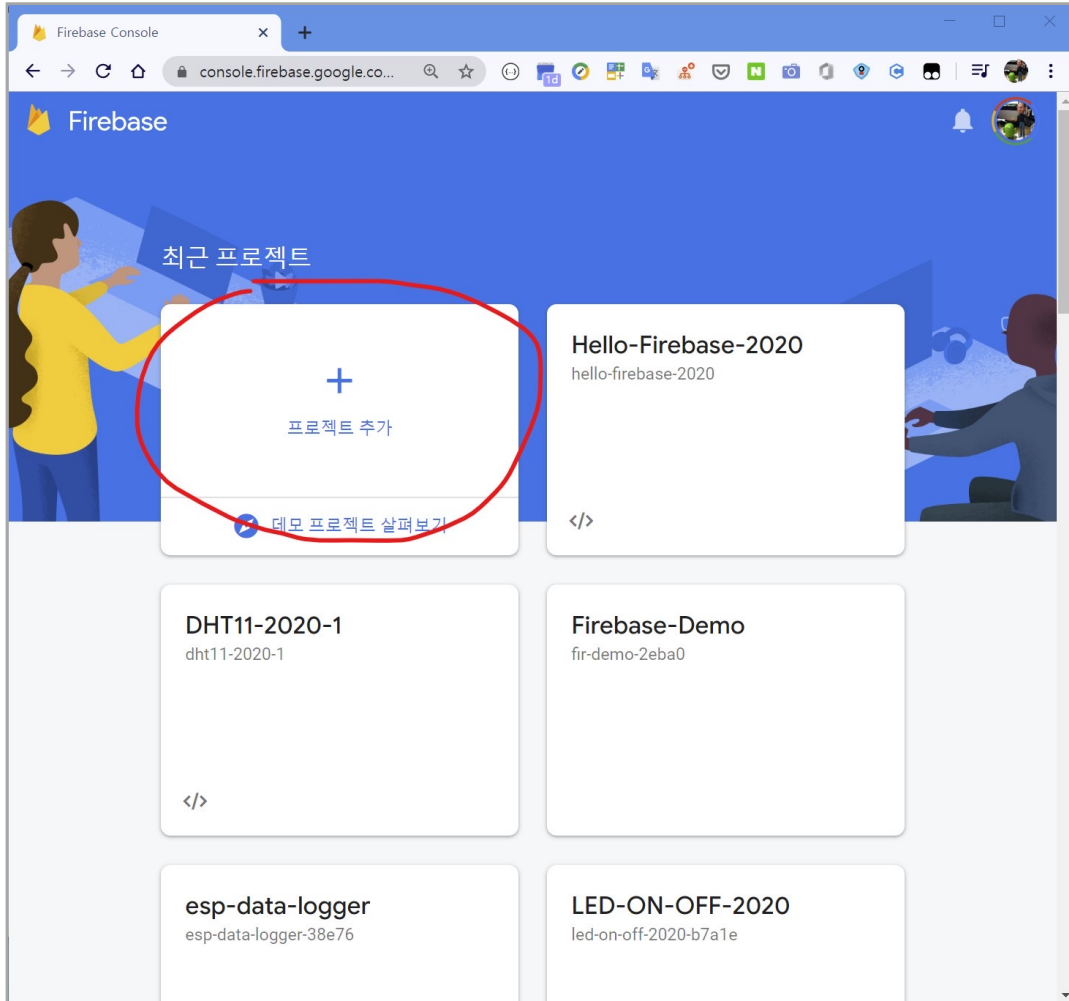


Hello Firebase

JavaScript 웹 앱 만들기

Week11

프로젝트 만들기



프로젝트 만들기

× 프로젝트 만들기(2/3단계)

Firebase 프로젝트를 위한 Google 애널리틱스

무제한 무료 분석 솔루션인 Google 애널리틱스를 사용하면 Firebase Crashlytics, 클라우드 메시징, 인앱 메시징, 원격 구성, A/B 테스트, 예측, Cloud Functions에서 타겟팅, 보고 등을 이용할 수 있습니다.

Google 애널리틱스를 통해 다음 기능을 이용할 수 있습니다.

- A/B 테스트
- Firebase 제품 전반에서 사용자 세분화 및 타겟팅
- 사용자 행동 예측
- 장애가 발생하지 않은 사용자
- 이벤트 기반 Cloud Functions 트리거
- 제한 없는 무료 보고

☒ 이 프로젝트에서 Google 애널리틱스 사용 설정 권장사항

이전

계속

× 프로젝트 만들기(3/3단계)

Google 애널리틱스 구성

Google 애널리틱스 계정 선택 또는 만들기

Default Account for Firebase


이 계정에서 자동으로 새 측정 만들기

프로젝트를 만들면 선택한 Google 애널리틱스 계정에 새 Google 애널리틱스 속성이 생성되고 Firebase 프로젝트에 연결됩니다. 이 연결을 통해 제품 간에 데이터 흐름이 활성화됩니다. Google 애널리틱스 속성에서 Firebase로 보내는 데이터에는 Firebase 서비스 약관이 적용되지만 Google 애널리틱스로 가져온 Firebase 데이터에는 Google 애널리틱스 서비스 약관이 적용됩니다. [자세히 알아보기](#)

이전

프로젝트 만들기

× 프로젝트 만들기(완료)

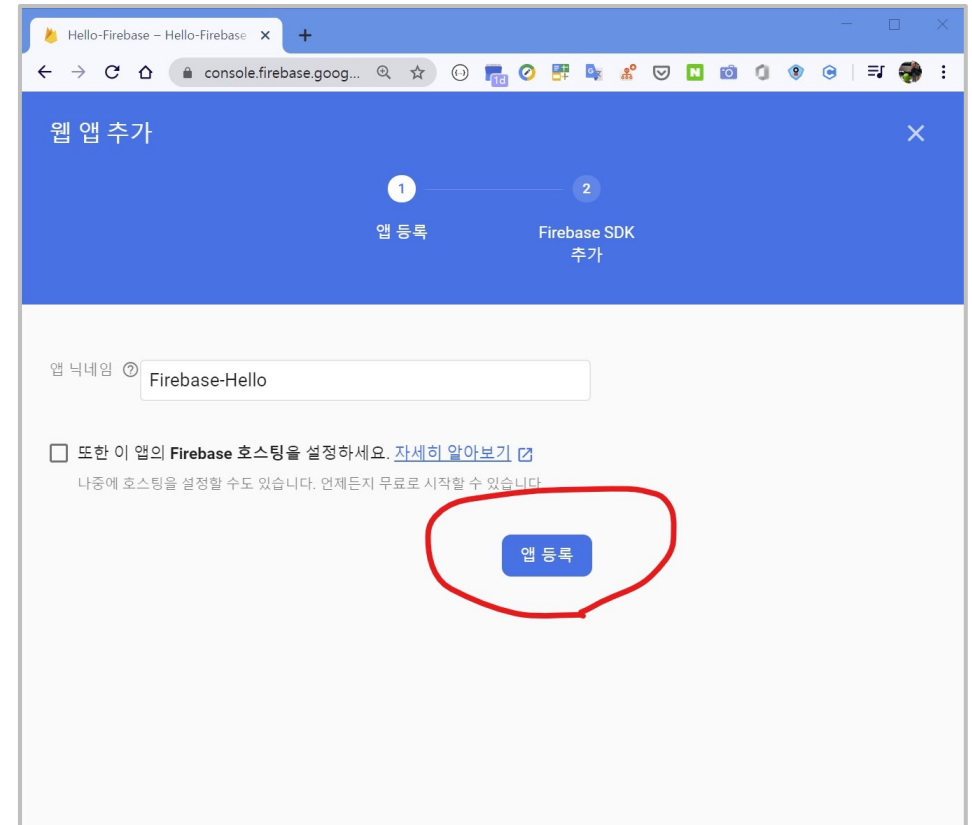
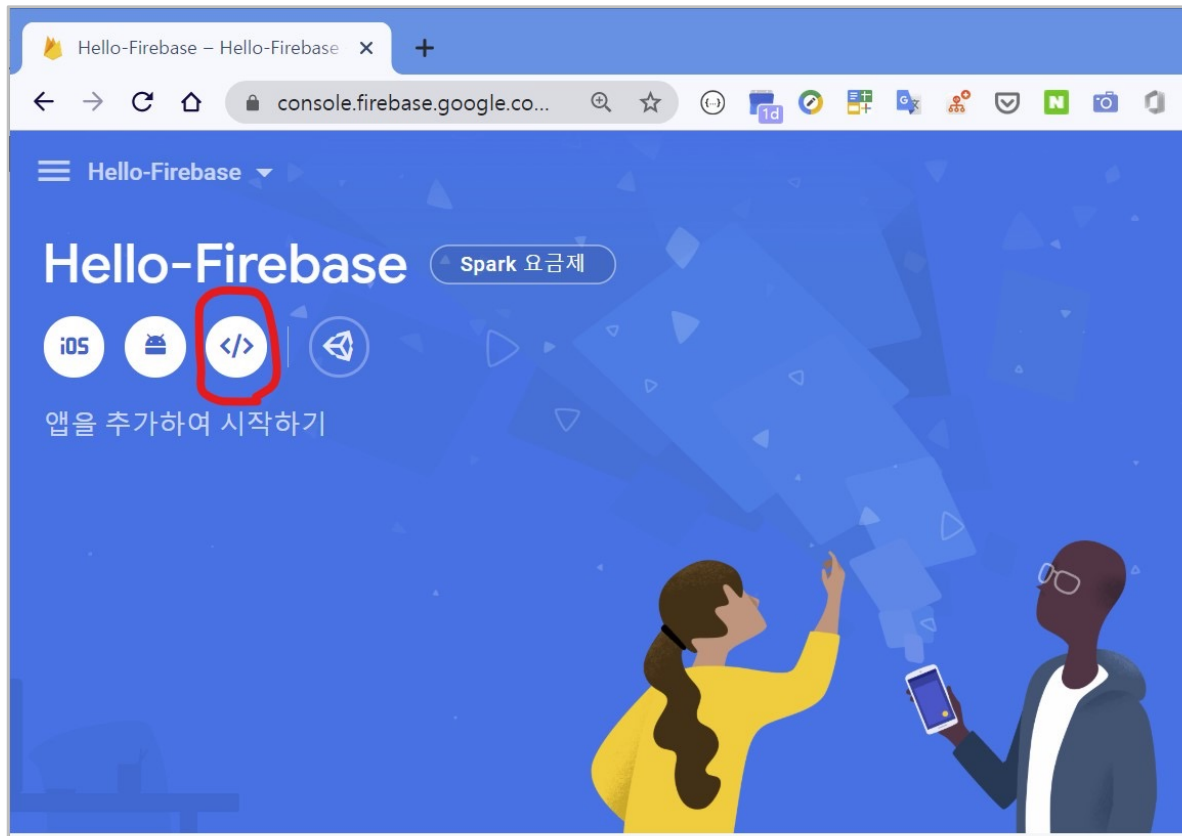


Hello-Firebase

✓ 새 프로젝트가 준비되었습니다.

계속

웹 앱 추가



Index.html 파일 만들기

```
<html>
<body>
  <p>Getting started with Firebase</p>
  <h1 id="bigOne"></h1>
  <script>
    // your firebase JavaScript code here
    bigOne.innerText = "Hello Firebase";
  </script>

</body>
</html>
```

웹 앱에 Firebase 설정 추가

웹 앱 추가

✓

2

앱 등록

Firebase SDK
추가

스크립트를 복사하여 <body> 태그 하단에 붙여넣으세요. Firebase 서비스를 사용하기 전에 진행해야 합니다.

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/7.15.1/firebase-app.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
      https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/7.15.1/firebase-analytics.js"></script>

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyCRk8YKM88p__NXa1ZjFqDxDxFYmEb7104",
    authDomain: "hello-firebase-44751.firebaseio.com",
    databaseURL: "https://hello-firebase-44751.firebaseio.com",
    projectId: "hello-firebase-44751",
    storageBucket: "hello-firebase-44751.appspot.com",
    messagingSenderId: "385587168776",
    appId: "1:385587168776:web:e59a5212426608ddecf3bd",
    measurementId: "G-NBEGY26JHW"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
</script>
```

Index.html에 Firebase 설정 추가

```
<body>
```

```
  <p>Getting started with Firebase</p>
```

```
  <h1 id="bigOne"></h1>
```

```
<script type="module">
```

```
  import { initializeApp } from "https://www.gstatic.com/firebasejs/9.5.0/firebase-app.js";
  import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.5.0/firebase-analytics.js";
  import { getDatabase, set, ref, onValue } from "https://www.gstatic.com/firebasejs/9.5.0/firebase-database.js";
```

Firebase SDK 추가

```
  const firebaseConfig = {
    apiKey: "AIzaSyAqD5Ng6TZfpx5CRmqLby5PWp13JSbhcho",
    authDomain: "hello-firebase-126d0.firebaseio.com",
    databaseURL: "https://hello-firebase-126d0-default-rtdb.firebaseio.com",
    projectId: "hello-firebase-126d0",
    storageBucket: "hello-firebase-126d0.appspot.com",
    messagingSenderId: "698300406698",
    appId: "1:698300406698:web:41ddfaa396efc9e05da9d3",
    measurementId: "G-CHYYQRW238"
  };
```

Firebase
Configuration
추가

☞ Firebase DB 연결, 이벤트 처리, 데이터 읽기 등~~~~

```
</script>
```

```
</body>
```

데이터베이스 연결, 데이터 쓰기/ 이벤트 처리/ 웹 페이지 출력

```
<script type="module">

  .....

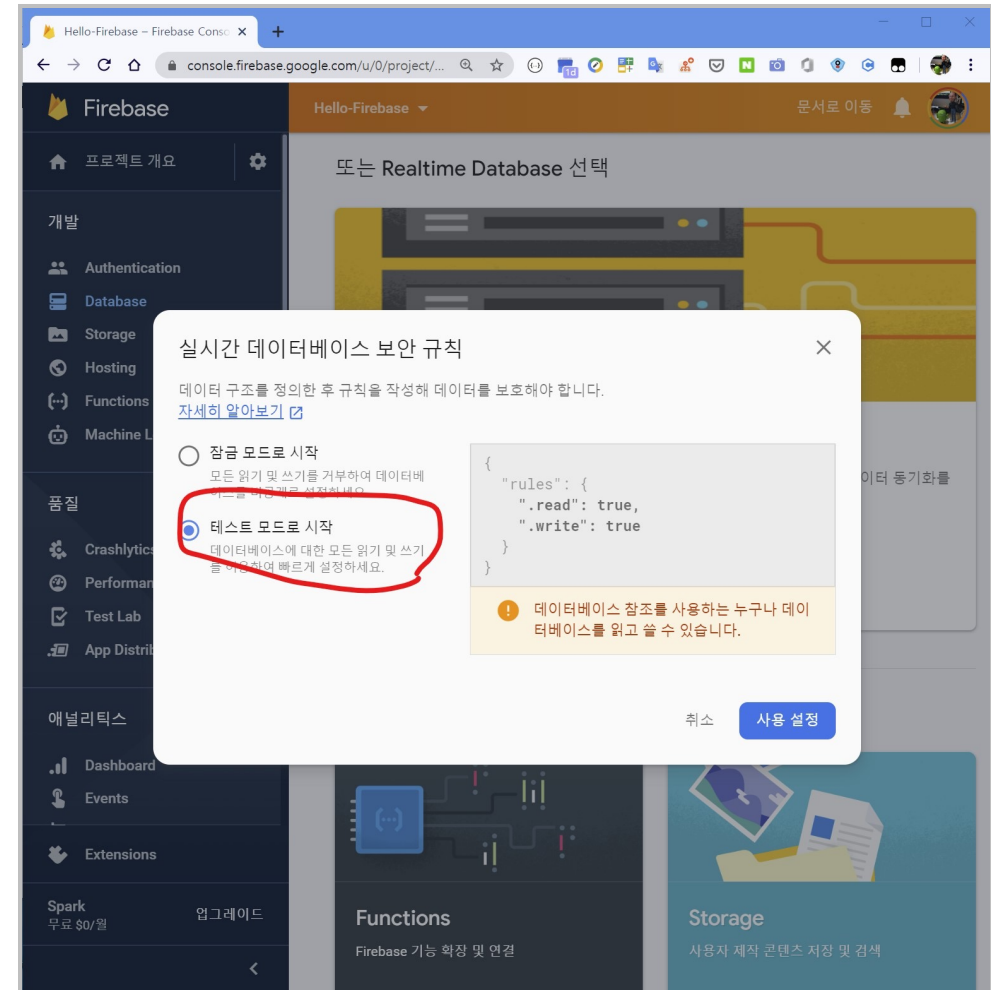
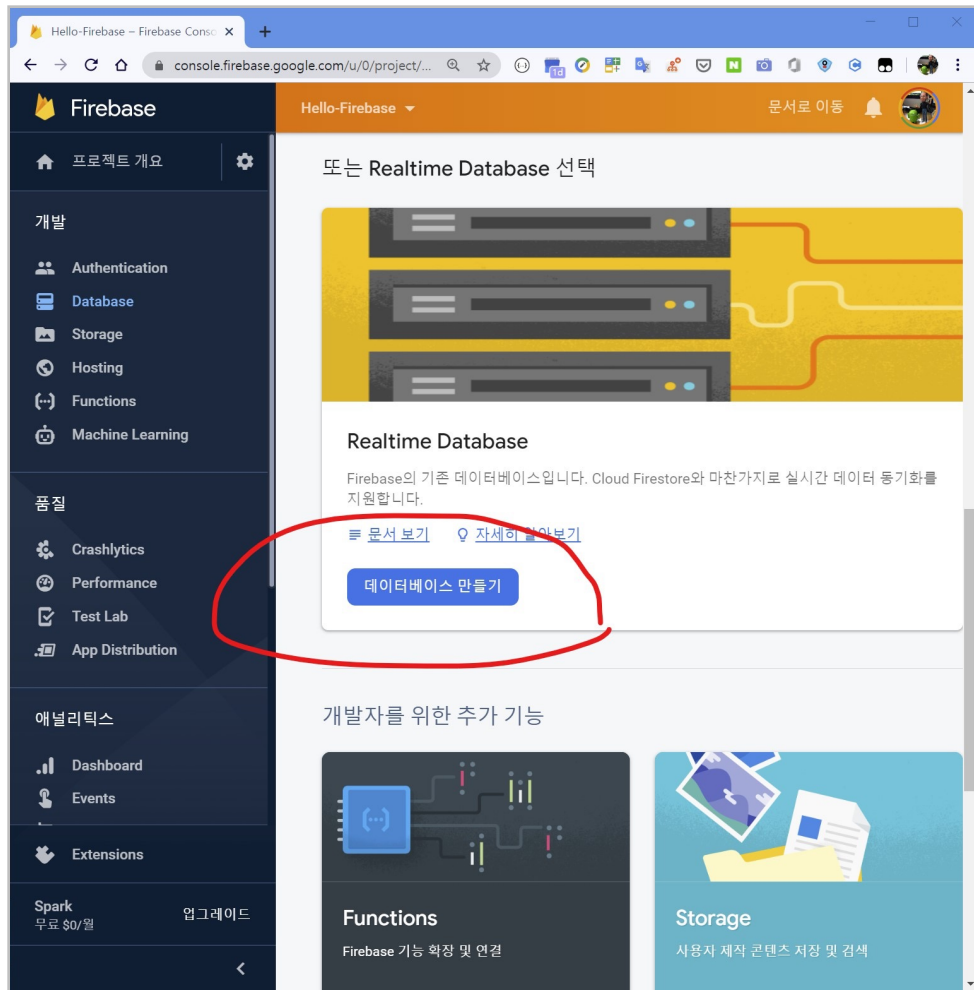
  const app = initializeApp(firebaseConfig);
  const analytics = getAnalytics(app);
  const db = getDatabase(app);

  // db 객체의 key: value 값 쓰기
  const dbRef = ref(db, 'Text');
  set(dbRef, 'Hello Firebase!!!');

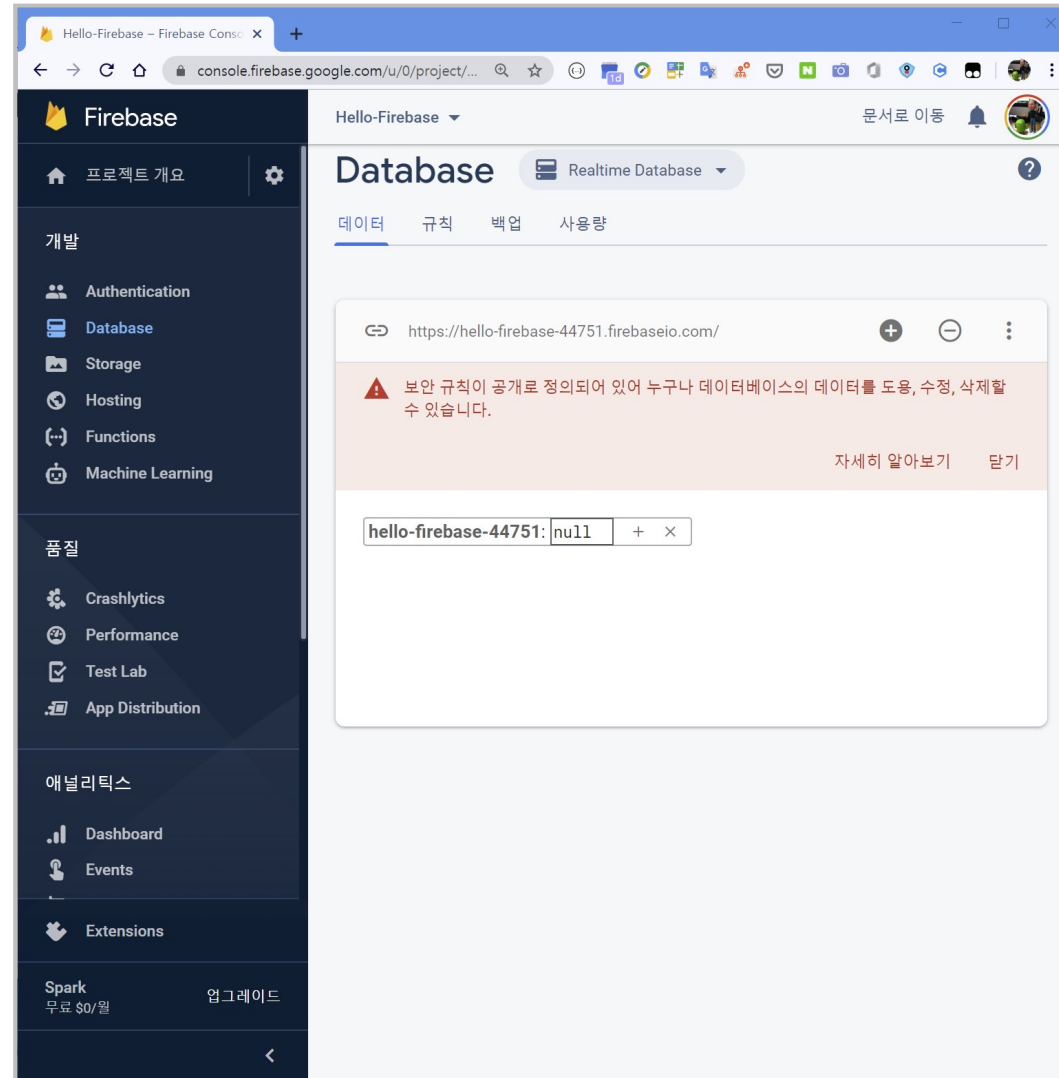
  // 쓰기 이벤트가 발생하면 db key에 대한 값을 웹페이지에 출력
  onValue(dbRef, (snapshot) => {
    console.log(snapshot.val());
    bigOne.innerText = snapshot.val();
  })
</script>
```

- 전체 소스 : https://github.com/DIT-IoT-Cloud-2021-2/Source/blob/main/Firebase_Hello_WebApp/index.htm

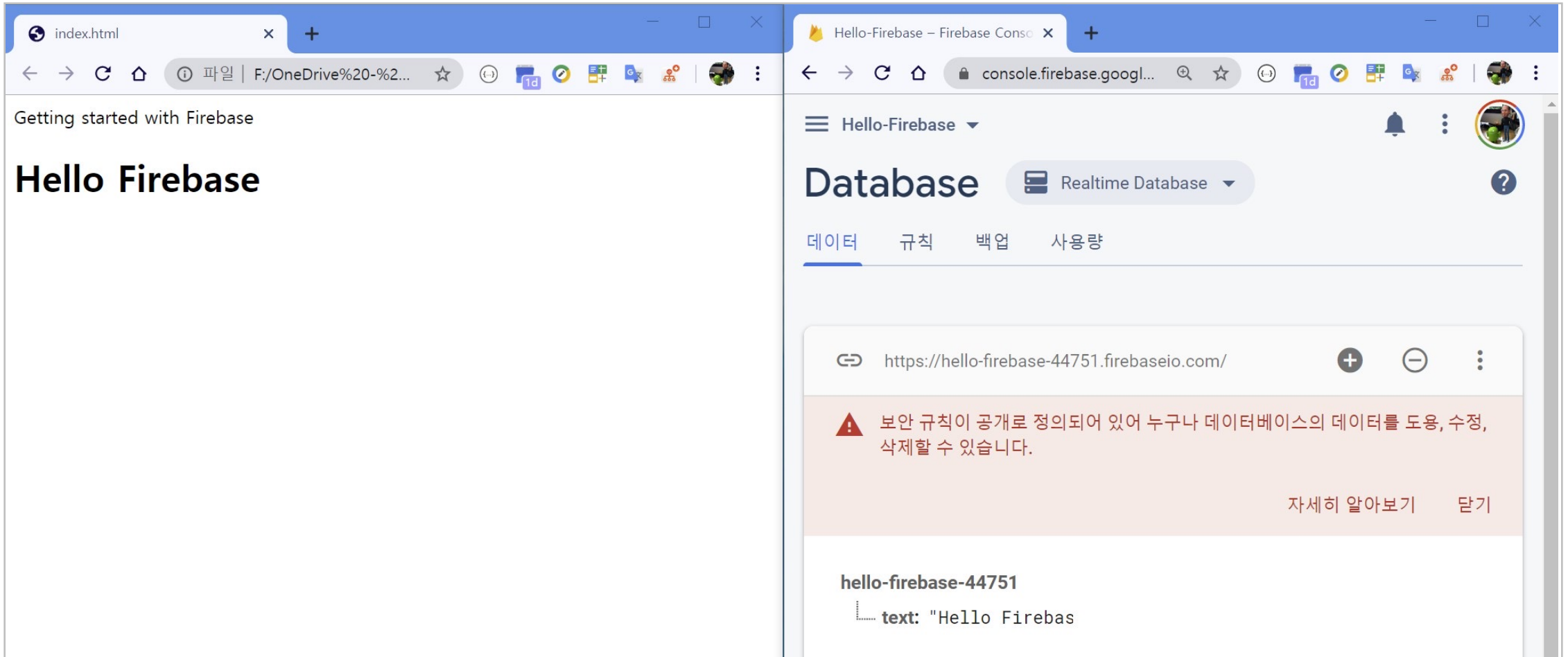
데이터베이스 만들기



데이터베이스 만들기



웹 앱(index.html) 실행



데이터베이스 값 변경 : update

The image shows a side-by-side comparison of a local web page and the Firebase console. On the left, a browser window displays 'index.html' with the text 'Getting started with Firebase' and 'Good Firebase'. The text 'Good Firebase' is circled in red. On the right, the Firebase console for 'Hello-Firebase' is shown. The 'Database' tab is active, displaying a warning about security rules and a data entry for 'hello-firebase-44751' with the value 'Good Firebas'. The text 'hello-firebase-44751' and 'text: "Good Firebas"' are circled in red.

index.html

Getting started with Firebase

Good Firebase

Hello-Firebase

Database

Realtime Database

데이터 규칙 백업 사용량

<https://hello-firebase-44751.firebaseio.com/>

⚠ 보안 규칙이 공개로 정의되어 있어 누구나 데이터베이스의 데이터를 도용, 수정, 삭제할 수 있습니다.

[자세히 알아보기](#) [닫기](#)

hello-firebase-44751

text: "Good Firebas"

과제 03-01

- Firebase 프로젝트 Hello-Firebase를 생성하여, 웹앱에 {"text", "hello Firebase"}를 출력하게 하시오.
- update 함수를 사용하여 데이터베이스에 저장된 {"text", "hello Firebase"}가 {"text", "Good Morning Firebase"}로 수정되도록 하시오.

Firestore 기반 IoT 온습도 모니터링 웹앱 만들기



Temperature

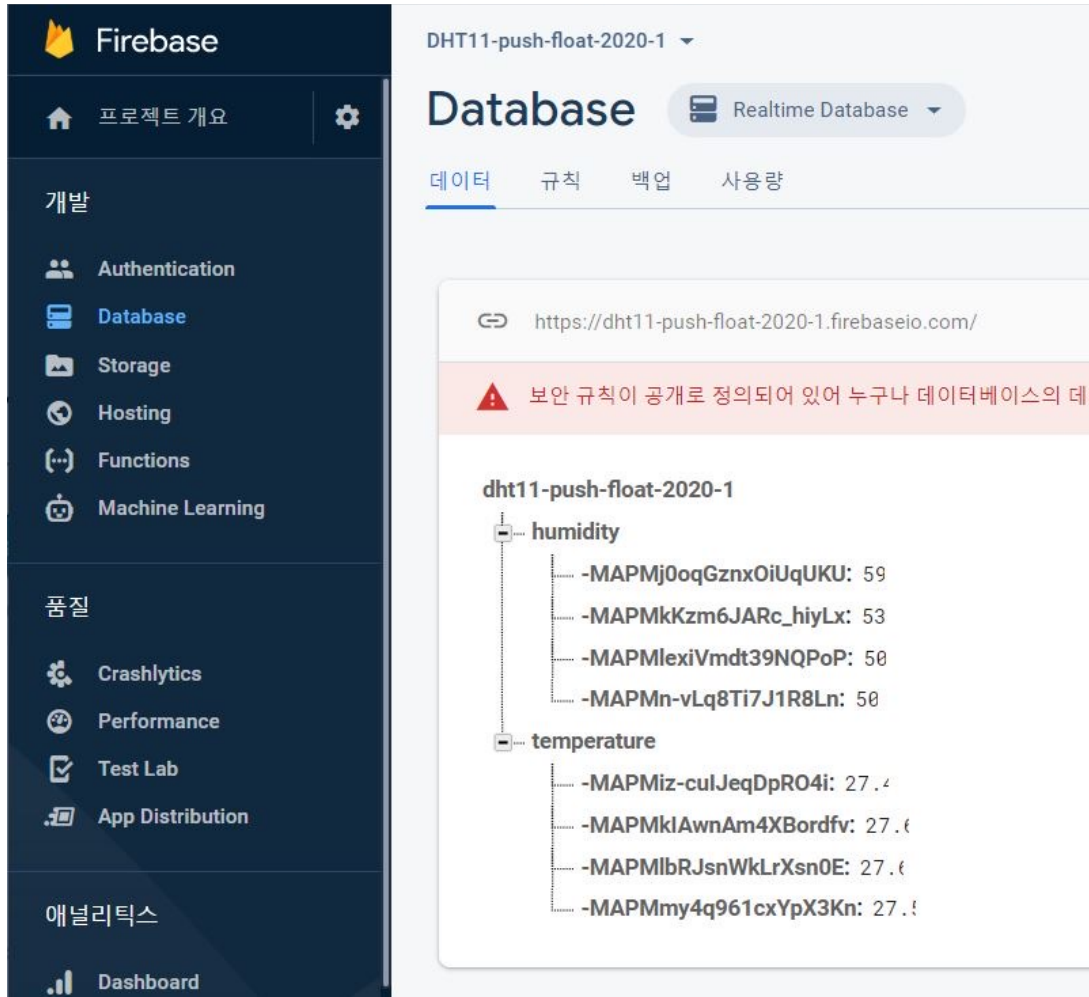


Humidity



두개의 객체 사용(pushFloat())

- Firebase – Read Data




// Ardunio Code

```
void loop() {  
    float temp = dht.readTemperature();  
    float humi = dht.readHumidity();
```

```
    Firebase.pushFloat("temperature", temp);  
    Firebase.pushFloat("humidity", humi);
```

```
    if (Firebase.failed()) {  
        Serial.print("pushing /logs failed:");  
        Serial.println(Firebase.error());  
        return;  
    }  
    Serial.print("pushed: /logDHT/");  
    delay(5000);  
}
```

- on() 메소드는 value 이벤트가 발생하면 데이터베이스의 스냅샷(snapshot)을 가져 온다.
- val() 메소드로 스냅샷의 값을 뽑아 웹페이지(Javascript)에 데이터를 출력한다.

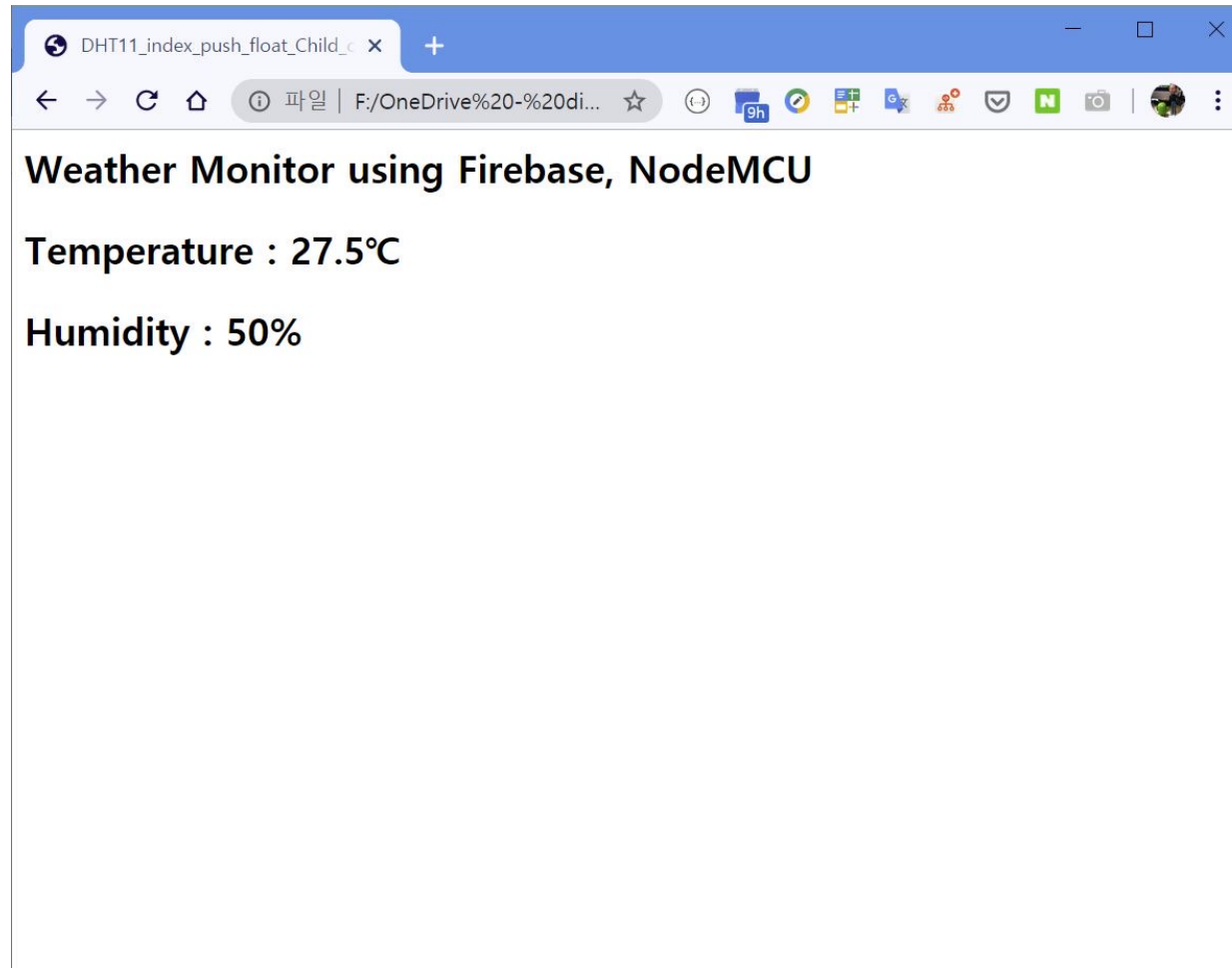
```
// index.html
<html>
<body>
  <h4> Weather Monitor using Firebase, NodeMCU
  <h4 id="temp"> </h4>
  <h4 id="humi"> </h4>
   insert web app's Firebase configuration
  <script>
    var dbRefHumi = firebase.database().ref().child('humidity');
    var dbRefTemp = firebase.database().ref().child('temperature');

    dbRefHumi.on("child_added", function(snapshot) {
      var newHumi = snapshot.val();
      humi.innerText = "Humidity : " + newHumi + "%";
    })

    dbRefTemp.on("child_added", function(snapshot) {
      var newTemp = snapshot.val();
      temp.innerText = "Temperature : " + newTemp + "°C";
    })
  </script>
</body>
</html>
```

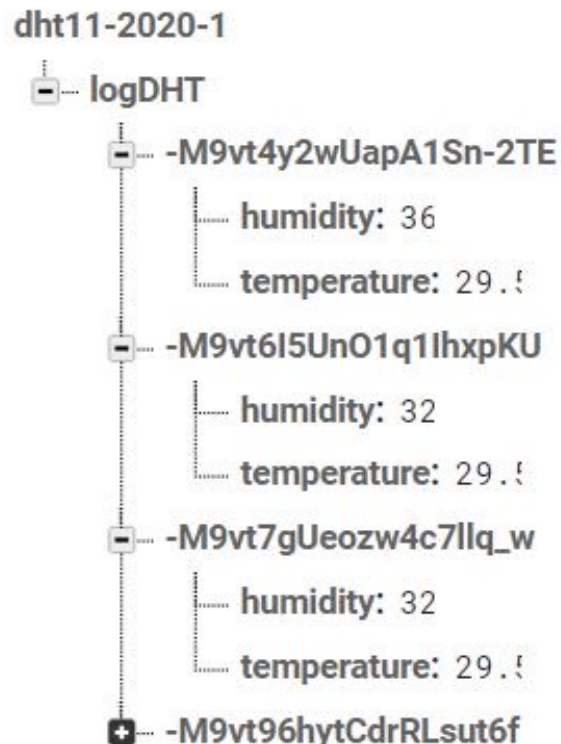
- 전체 소스 코드 : <https://github.com/loT-Lab-02/DHT11-Firebase-Web-PushFloat>

웹 앱 결과 : temperature, Humidity / pushFloat()



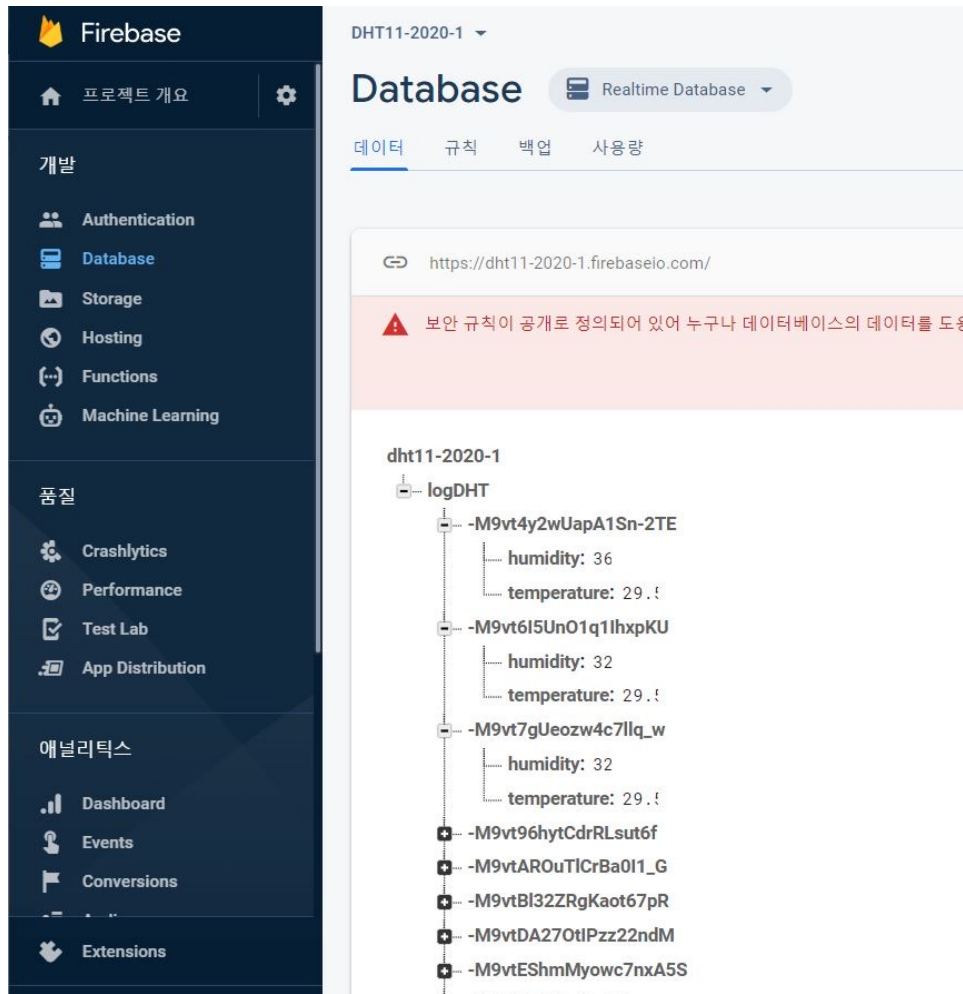
Firestore-Event Types

- value
 - 데이터베이스의 데이터가 변화가 있을 때 이벤트 발생
 - 자식(children)을 포함하는 모든 데이터를 읽어온다.
- child_added
 - 데이터베이스에 새로운 하위 객체(child)가 추가될 때 이벤트 발생
 - 새로 추가된 객체를 읽어 온다.



하나의 객체 사용 : StaticJsonBuffer/ push()

- Firebase – Read Data



// Ardunio Code

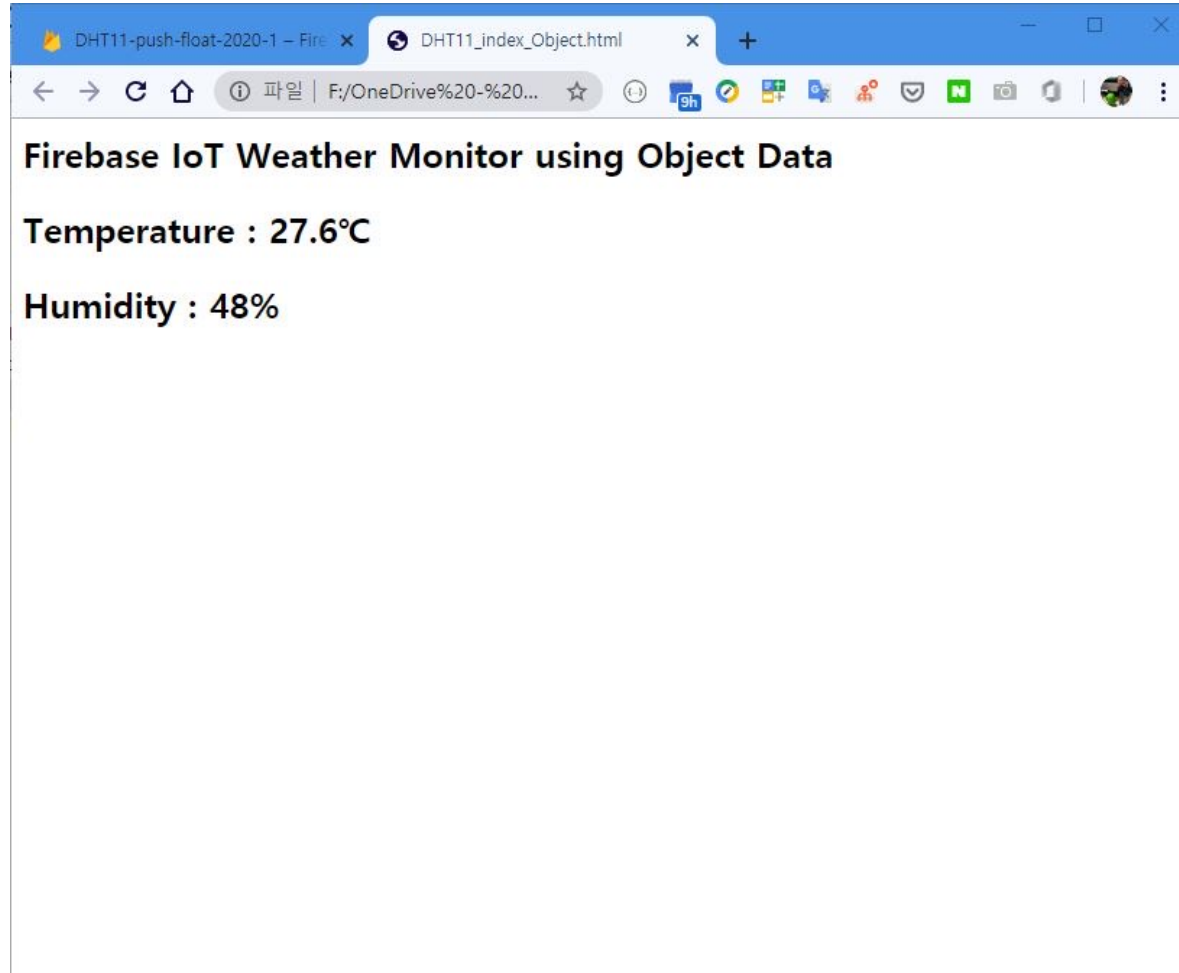
```
void loop() {  
    float temp = dht.readTemperature();  
    float humi = dht.readHumidity();  
  
    StaticJsonBuffer<200> jsonbuffer;  
    JsonObject& root = jsonbuffer.createObject();  
    root["temperature"] = temp;  
    root["humidity"] = humi;  
  
    String name = Firebase.push("logDHT", root);  
  
    if (Firebase.failed()) {  
        Serial.print("pushing /logs failed:");  
        Serial.println(Firebase.error());  
        return;  
    }  
    Serial.print("pushed: /logDHT/");  
    delay(5000);  
}
```

- on() 메소드는 value 이벤트가 발생하면 데이터베이스의 스냅샷(snapshot)을 가져 온다.
- val() 메소드로 스냅샷의 값을 뽑아 웹페이지(JavaScript)에 데이터를 출력한다.

```
// index.html
<html>
<body>
  <h4> Weather Monitor using Firebase, NodeMCU
  <h4 id="temp"> </h4>
  <h4 id="humi"> </h4>
  ✎ insert web app's Firebase configuration
  <script>
var objRef = firebase.database().ref().child('logDHT/');
objRef.on("child_added", function(snapshot) {
  var newHumi = snapshot.val().humidity;
  var newTemp = snapshot.val().temperature;
  humi.innerHTML = "Humidity : " + newHumi + "%";
  temp.innerHTML = "Temperature : " + newTemp + "°C";
</script>
</body>
</html>
```

- 전체 소스 코드 : <https://github.com/loT-Lab-02/DHT11-Firebase-Web-PushObject>

웹 앱 결과 : Object/ push()



Firestore Tutorial

- https://github.com/loT-Lab-02/Class-Resource/blob/master/firebase_tutorial.pdf



- 과제 03-02-01

- 두개의 객체(temperature, humidity/ pushFloat())를 사용하여 온습도 모니터링 웹앱을 만드시오.

- 과제 03-02-02

- 한 개의 객체(Object/ push())를 사용하여 온습도 모니터링 웹앱을 만드시오.

과제 03-03

- CSS, 이미지 등을 사용하여 웹앱의 UI를 보기 좋게 만드시오.
- Bootstrap 활용 참고 :

https://www.w3schools.com/bootstrap/bootstrap_ver.asp

Temperature
27.2 °C

Humidity
72 %

```
pushed: /logDHT/-M9kxtY3IJExghg0y72t
Temperature = 27.10
Humidity = 72.00
pushed: /logDHT/-M9kxv4CkygVSkNht1ml
Temperature = 27.10
Humidity = 72.00
pushing /logs failed:
Temperature = 27.10
```

```
DHT11_Firebase_DB_NodeMCU
1 #include <ESP8266WiFi.h>
2 #include <FirebaseArduino.h>
3
4 #define DHTPIN D4
5 #define DHTTYPE DHT11
6
7 #include <DHT.h>
8 DHT dht(DHTPIN, DHTTYPE);
9
10 // Set these to run example
11 #define FIREBASE_HOST "dht1
12 #define FIREBASE_AUTH "f6cH
13 #define WIFI_SSID "Amadeus"
14 #define WIFI_PASSWORD "deit
15
16 void setup() {
17   Serial.begin(9600);
18   // connect to wifi.
19   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
20   dht.begin(9600);
21
22   Serial.print("connecting");
23   while (WiFi.status() != WL_CONNECTED) {
24     Serial.print(".");
25     delay(500);
```