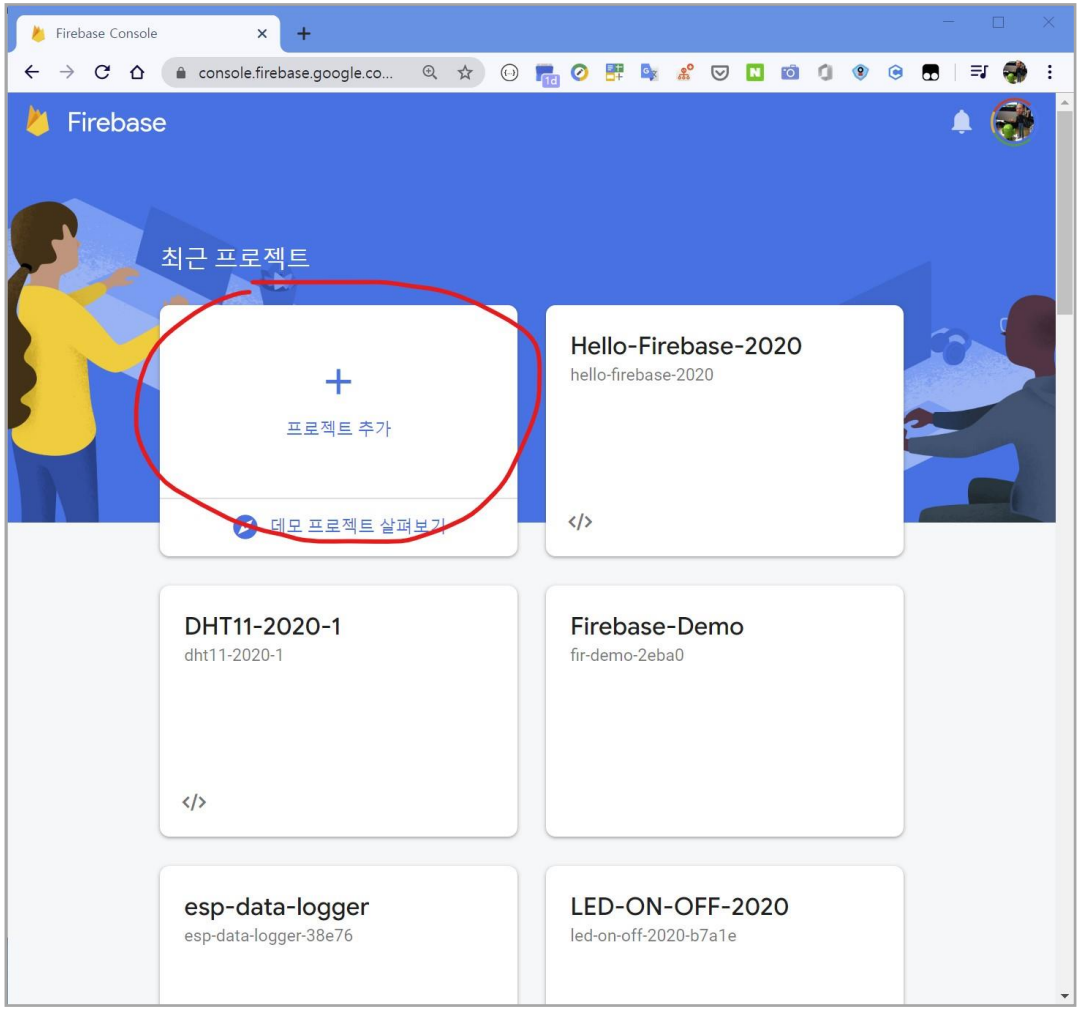
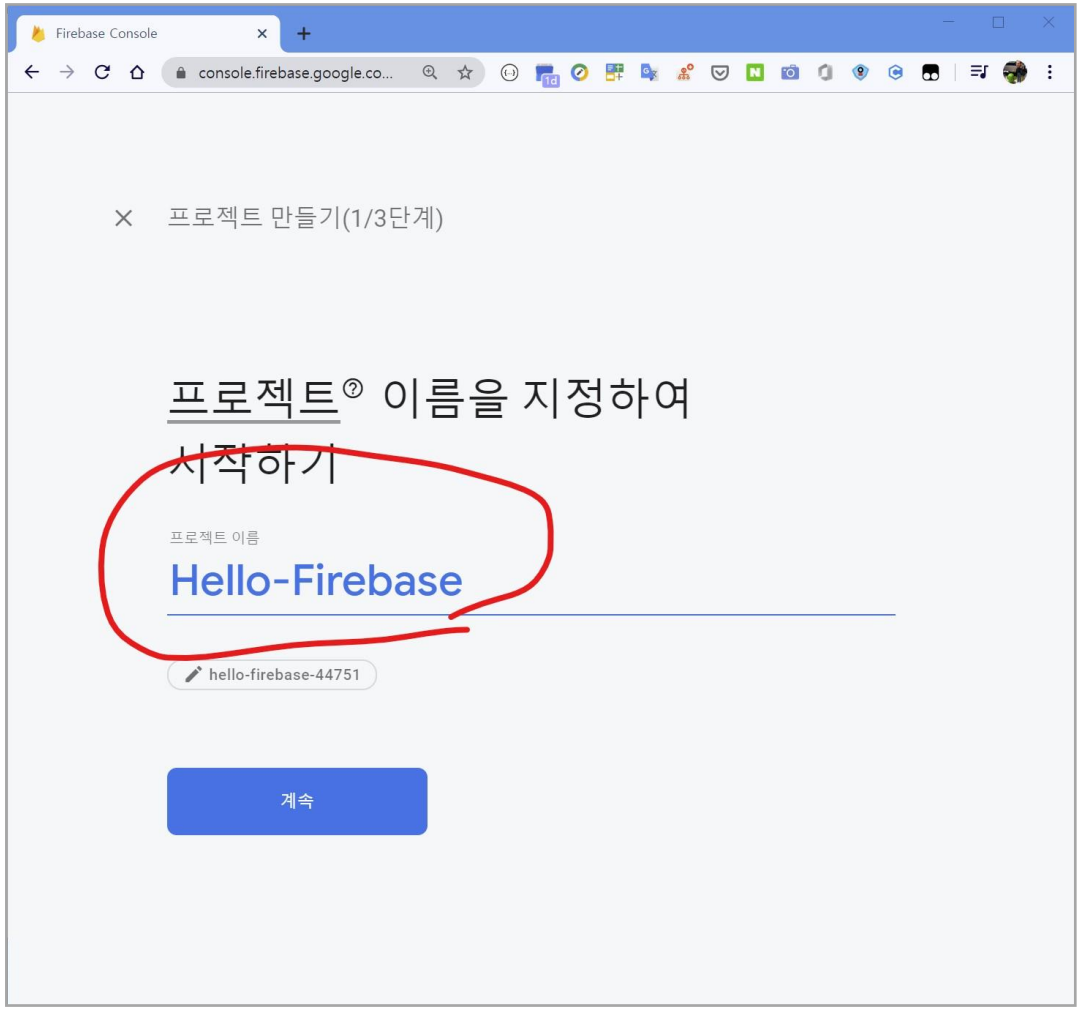
Hello Firebase
JavaScript 웹앱 만들기

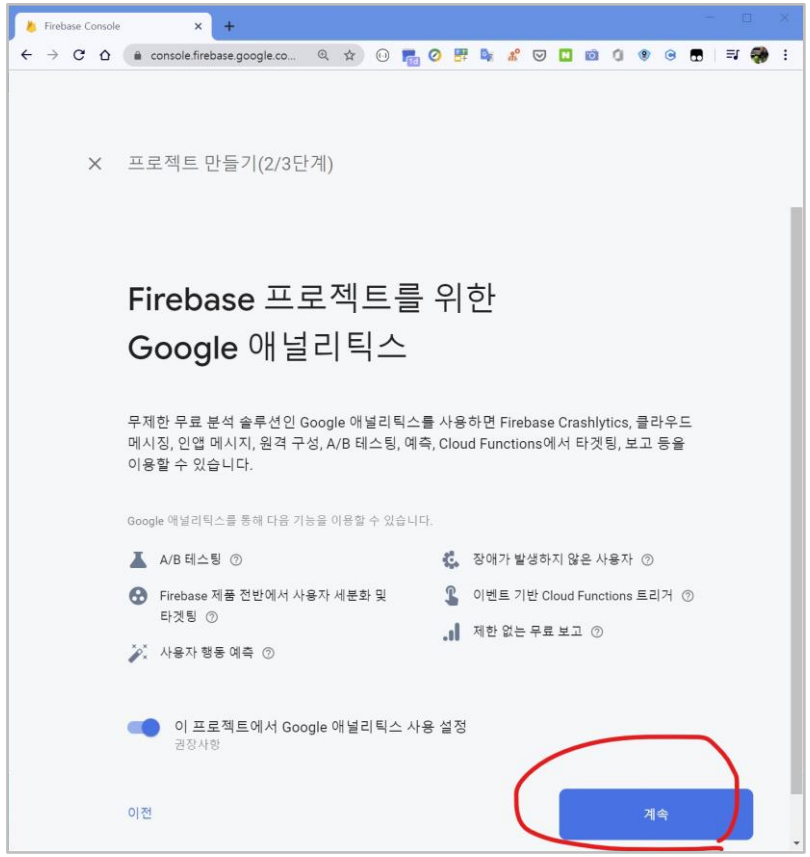
Week11

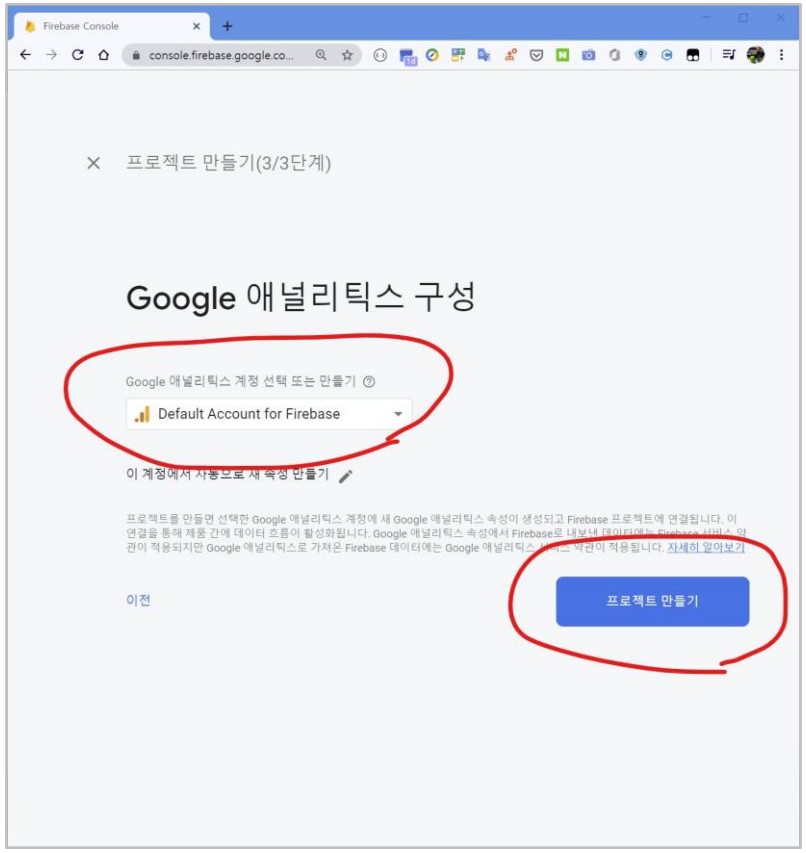
프로젝트 만들기

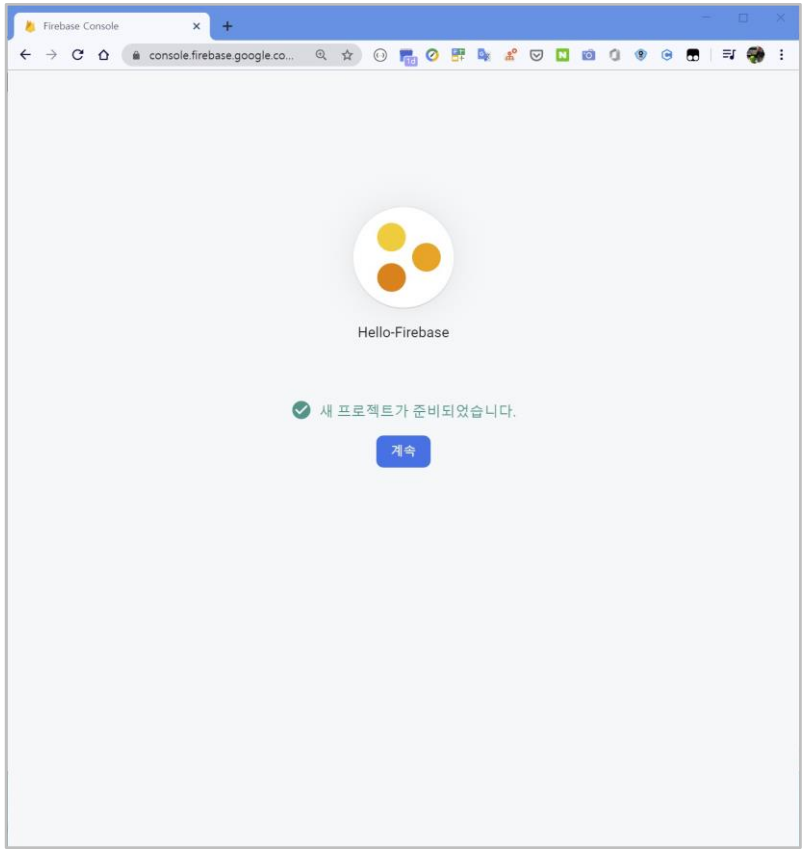

Firebase Console
console.firebase.google.co...
Firebase
최근 프로젝트
프로젝트 추가
데모 프로젝트 살펴보기
Hello-Firebase-2020
hello-firebase-2020
DHT11-2020-1
dht11-2020-1
Firebase-Demo
fir-demo-2eba0
esp-data-logger
esp-data-logger-38e76
LED-ON-OFF-2020
led-on-off-2020-b7a1e


Firebase Console
console.firebase.google.co...
프로젝트 만들기(1/3단계)
프로젝트 이름을 지정하여 시작하기
프로젝트 이름
Hello-Firebase
hello-firebase-44751
계속

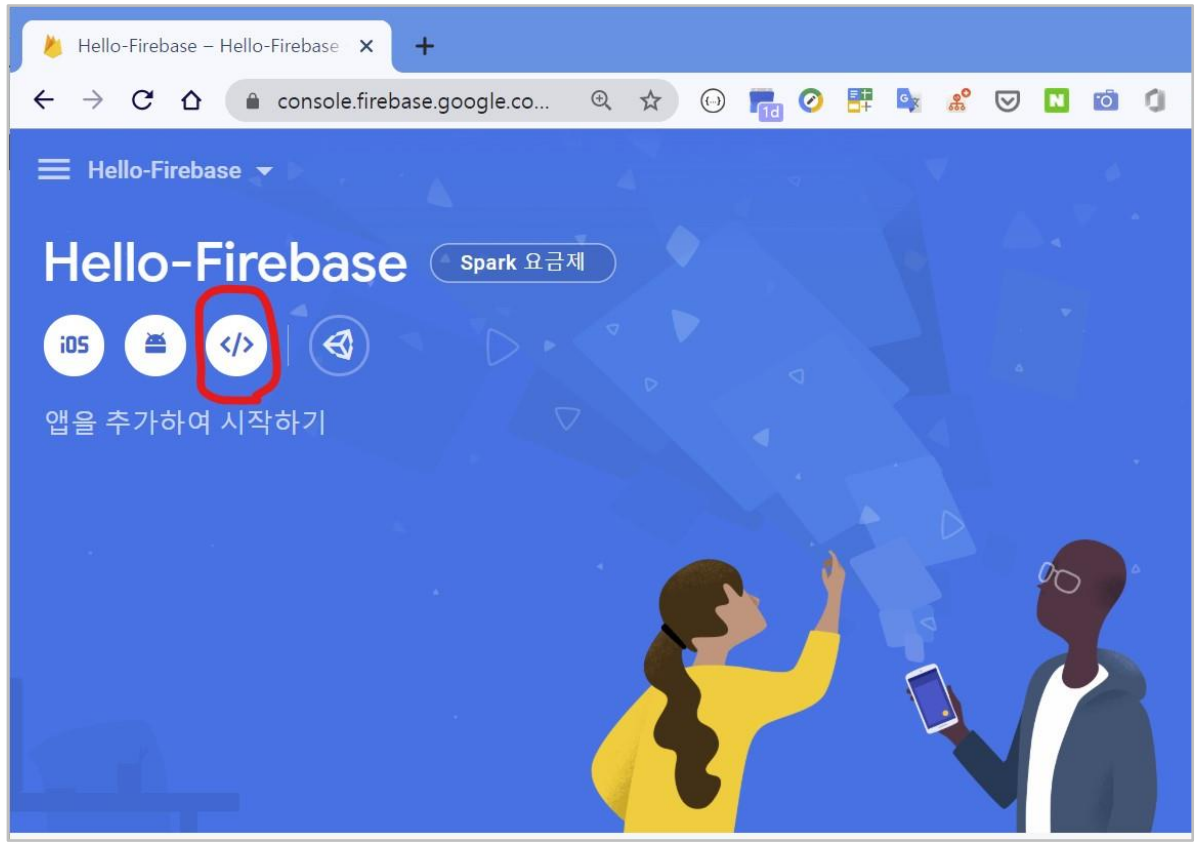
프로젝트 만들기

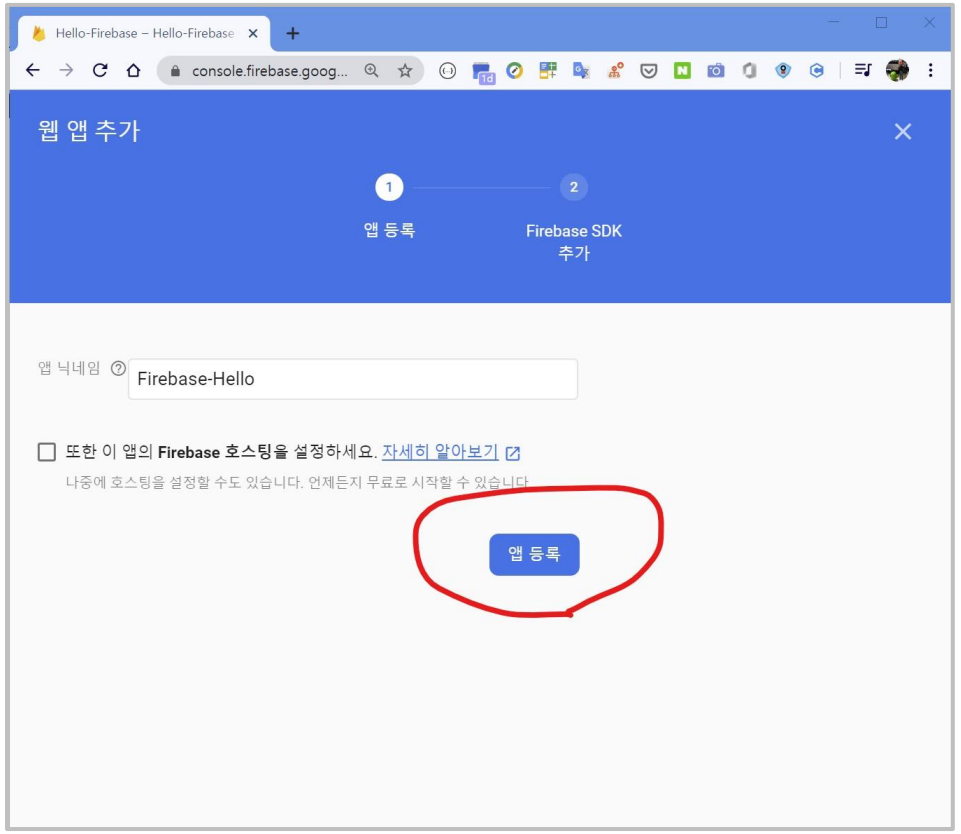

Firebase Console
console.firebase.google.co...
프로젝트 만들기(2/3단계)
Firebase 프로젝트를 위한
Google 애널리틱스
무제한 무료 분석 솔루션인 Google 애널리틱스를 사용하면 Firebase Crashlytics, 클라우드 메시징, 인앱 메시지, 원격 구성, A/B 테스팅, 예측, Cloud Functions에서 타겟팅, 보고 등을 이용할 수 있습니다.
Google 애널리틱스를 통해 다음 기능을 이용할 수 있습니다.
A/B 테스팅
장애가 발생하지 않은 사용자
Firebase 제품 전반에서 사용자 세분화 및 타겟팅
이벤트 기반 Cloud Functions 트리거
제한 없는 무료 보고
사용자 행동 예측
이 프로젝트에서 Google 애널리틱스 사용 설정
권장사항
이전
계속


Firebase Console
console.firebase.google.co...
프로젝트 만들기(3/3단계)
Google 애널리틱스 구성
Google 애널리틱스 계정 선택 또는 만들기
Default Account for Firebase
이 계정에서 자동으로 새 속성 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 내보낸 데이터에는 Firebase 서비스 약관이 적용되지만 Google 애널리틱스로 가져온 Firebase 데이터에는 Google 애널리틱스 서비스 약관이 적용됩니다. 자세히 알아보기
이전
프로젝트 만들기


Firebase Console
console.firebase.google.co...
Hello-Firebase
새 프로젝트가 준비되었습니다.
계속

웹 앱 추가

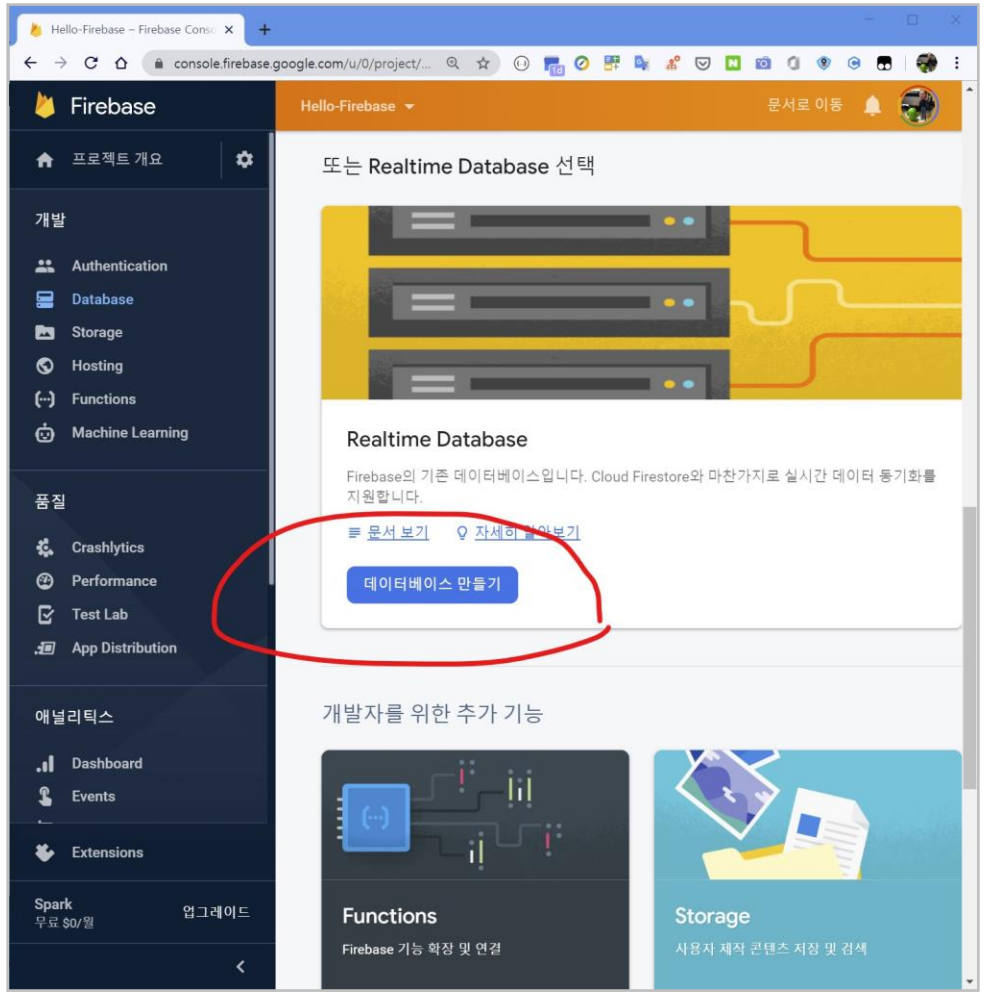

Hello-Firebase – Hello-Firebase
console.firebase.google.co...
Hello-Firebase
Hello-Firebase
Spark 요금제
iOS
앱을 추가하여 시작하기

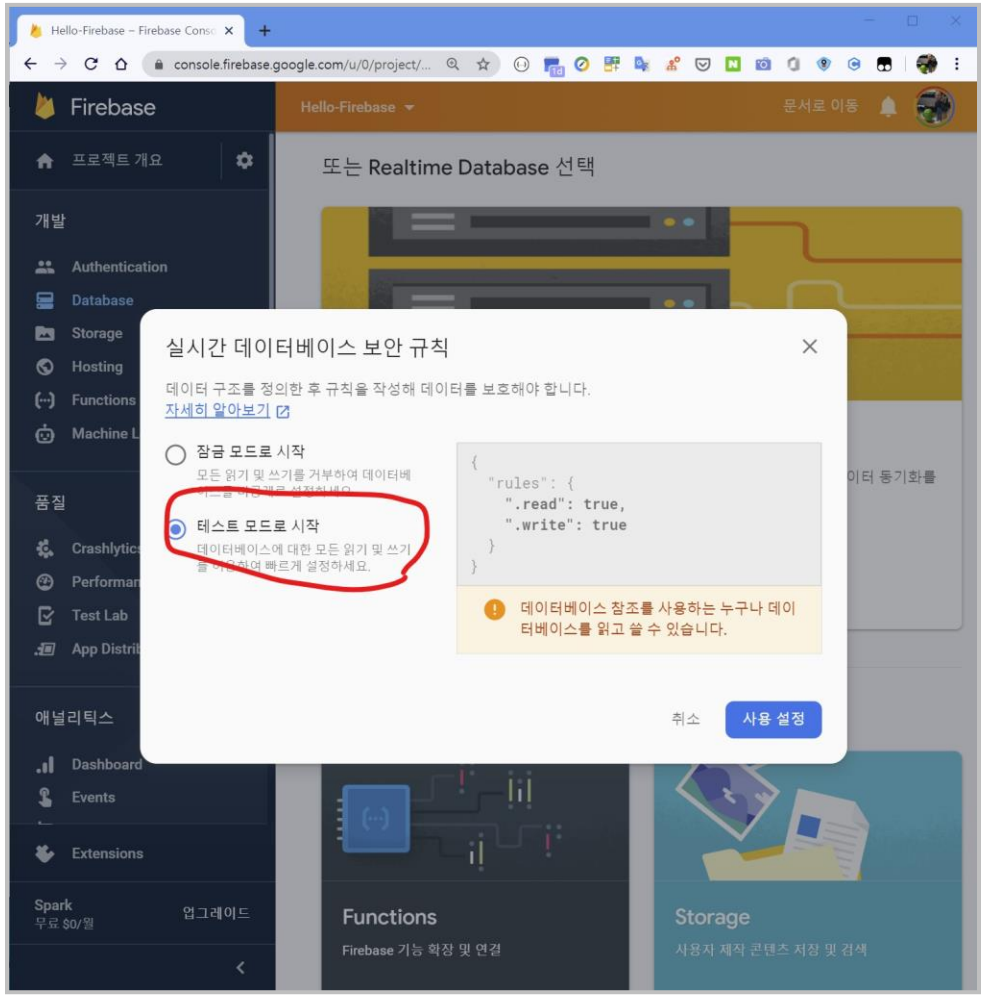

Hello-Firebase – Hello-Firebase
console.firebase.goog...
웹 앱 추가
1
앱 등록
2
Firebase SDK
추가
앱 닉네임
Firebase-Hello
또한 이 앱의 Firebase 호스팅을 설정하세요. 자세히 알아보기
나중에 호스팅을 설정할 수도 있습니다. 언제든지 무료로 시작할 수 있습니다.
앱 등록

Index.html 파일 만들기

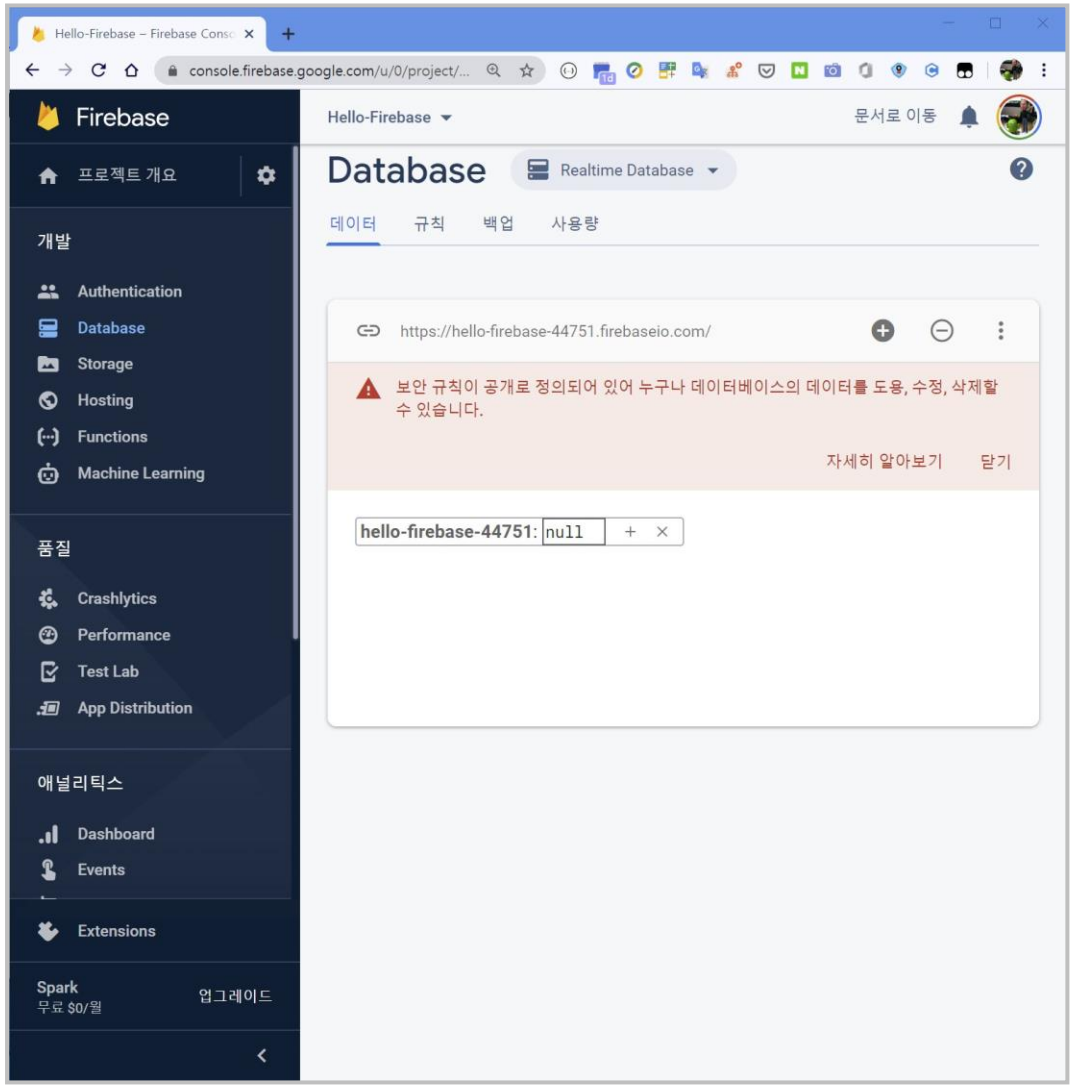
```
<html>
<body>
	<p>Getting started with Firebase</p>
	<h1 id="bigOne"></h1>
	<script>
		// your firebase JavaScript code here
		bigOne.innerText = “Hello Firebase”;
	</script>

  </body>
</html>
```

데이터베이스 만들기


Firebase
Hello-Firebase
문서로 이동
프로젝트 개요
개발
Authentication
Database
Storage
Hosting
Functions
Machine Learning
품질
Crashlytics
Performance
Test Lab
App Distribution
애널리틱스
Dashboard
Events
Extensions
Spark
무료 $0/월
업그레이드
또는 Realtime Database 선택
Realtime Database
Firebase의 기존 데이터베이스입니다. Cloud Firestore와 마찬가지로 실시간 데이터 동기화를 지원합니다.
문서 보기
자세히 알아보기
데이터베이스 만들기
개발자를 위한 추가 기능
Functions
Firebase 기능 확장 및 연결
Storage
사용자 제작 콘텐츠 저장 및 검색


Firebase
Hello-Firebase
문서로 이동
프로젝트 개요
개발
Authentication
Database
Storage
Hosting
Functions
Machine Learning
품질
Crashlytics
Performance
Test Lab
App Distribution
애널리틱스
Dashboard
Events
Extensions
Spark
무료 $0/월
업그레이드
또는 Realtime Database 선택
실시간 데이터베이스 보안 규칙
데이터 구조를 정의한 후 규칙을 작성해 데이터를 보호해야 합니다.
자세히 알아보기
잠금 모드로 시작
테스트 모드로 시작
데이터베이스에 대한 모든 읽기 및 쓰기를 허용하여 빠르게 설정하세요.
{
"rules": {
".read": true,
".write": true
}
}
데이터베이스 참조를 사용하는 누구나 데이터베이스를 읽고 쓸 수 있습니다.
취소
사용 설정
Functions
Firebase 기능 확장 및 연결
Storage
사용자 제작 콘텐츠 저장 및 검색

데이터베이스 만들기


Hello-Firebase – Firebase Conso
console.firebase.google.com/u/0/project/...
Firebase
Hello-Firebase
문서로 이동
프로젝트 개요
Database
Realtime Database
개발
데이터
규칙
백업
사용량
Authentication
Database
Storage
Hosting
Functions
Machine Learning
https://hello-firebase-44751.firebaseio.com/
보안 규칙이 공개로 정의되어 있어 누구나 데이터베이스의 데이터를 도용, 수정, 삭제할 수 있습니다.
자세히 알아보기
닫기
품질
hello-firebase-44751: null
Crashlytics
Performance
Test Lab
App Distribution
애널리틱스
Dashboard
Events
Extensions
Spark
무료 $0/월
업그레이드

웹앱에 Firebase SDK 추가 : <script 태크 추가>

웹 앱에 Firebase 추가

앱 등록

2 Firebase SDK 추가

npm 사용 <script> 태그 사용

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

```
<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from "https://www.gstatic.com/firebasejs/9.5.0/fireb
  import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.5.0/fireba
  // 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: "AIzaSyAKhw5lRmQ7f8kt5kQTSeELqLlmjrZRdRY",
    authDomain: "hello-firebase02-aaf67.firebaseapp.com",
    projectId: "hello-firebase02-aaf67",
    storageBucket: "hello-firebase02-aaf67.appspot.com",
    messagingSenderId: "70829304852",
    appId: "1:70829304852:web:e3384fd16e1353653acb01",
    measurementId: "G-5TBBQQRYNT"
  };

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

    const firebaseConfig = {
            apiKey: "AIzaSyAqD5Ng6TZfpx5CRmqLby5PWp13JSbhcho",
            authDomain: "hello-firebase-126d0.firebaseapp.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 DB 연결, 이벤트 처리, 데이터 읽기 등~~~~
</script>
</body>
```

Firebase SDK 추가

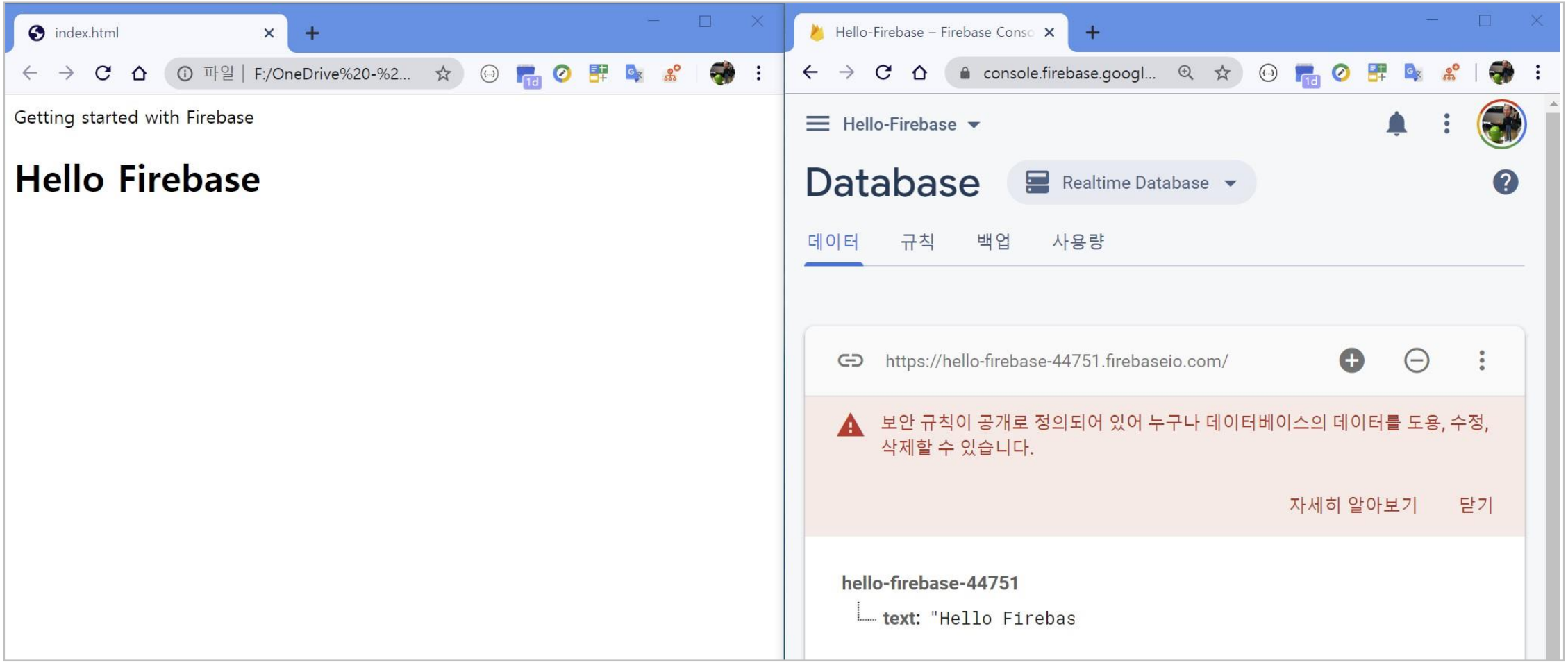
Firebase Configuration 추가

데이터베이스 연결, 데이터 쓰기/ 이벤트 처리/ 웹 페이지 출력

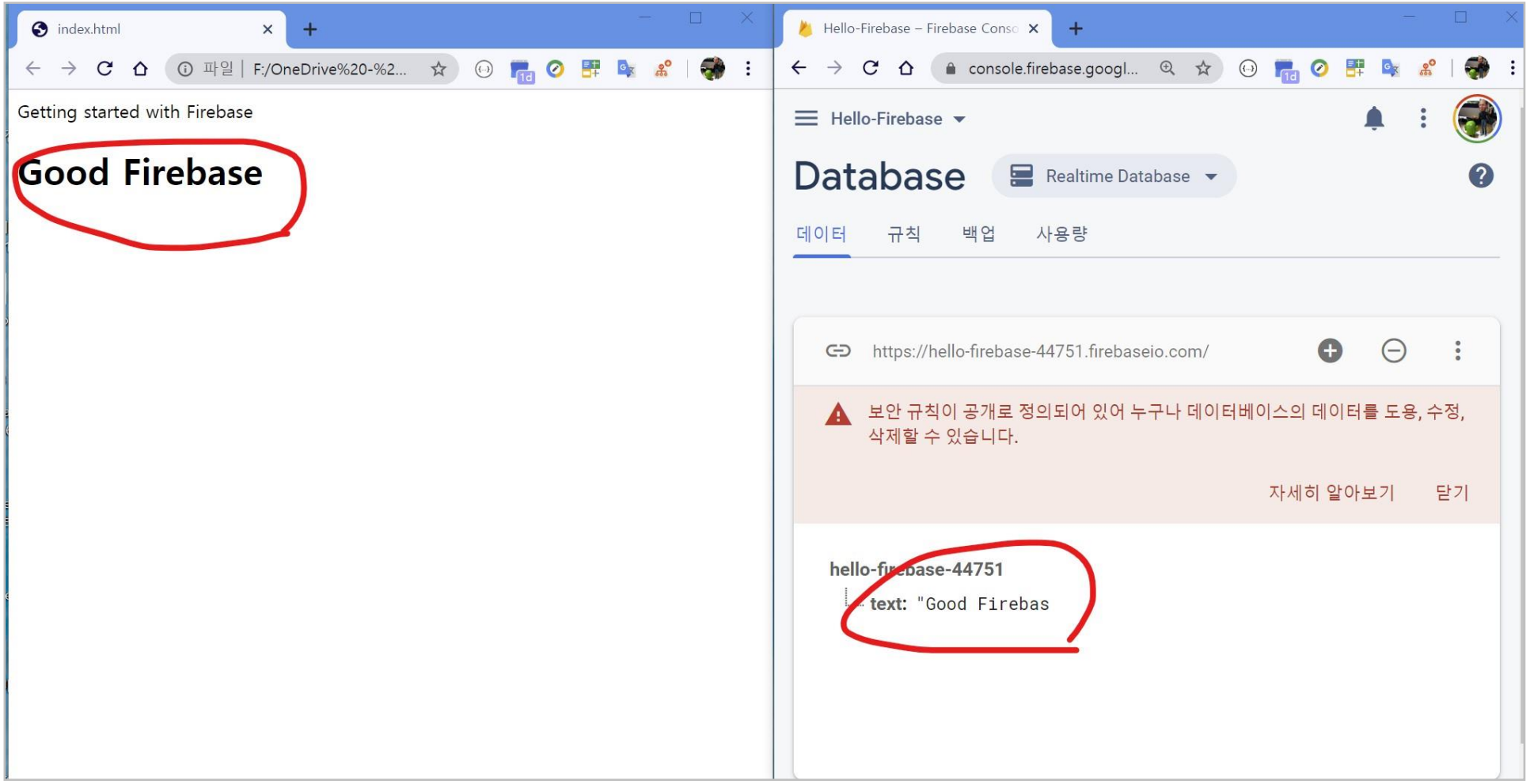
```
<script type="module">
        .....
        const app = initializeApp(firebaseConfig);
        const analytics = getAnalytics(app);
        const db = getDatabase(app);

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

       // 쓰기 이벤트가 방생하면 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

웹앱(index.html) 실행



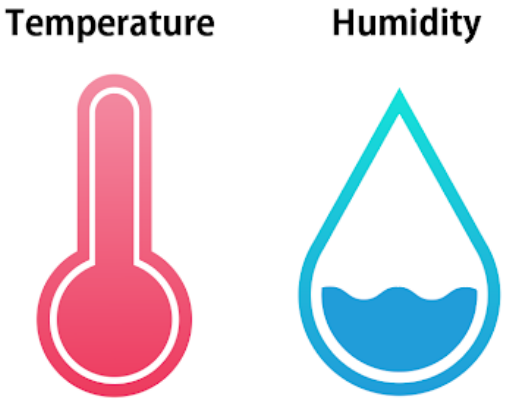
데이터베이스 값 변경 : update



과제 03-01

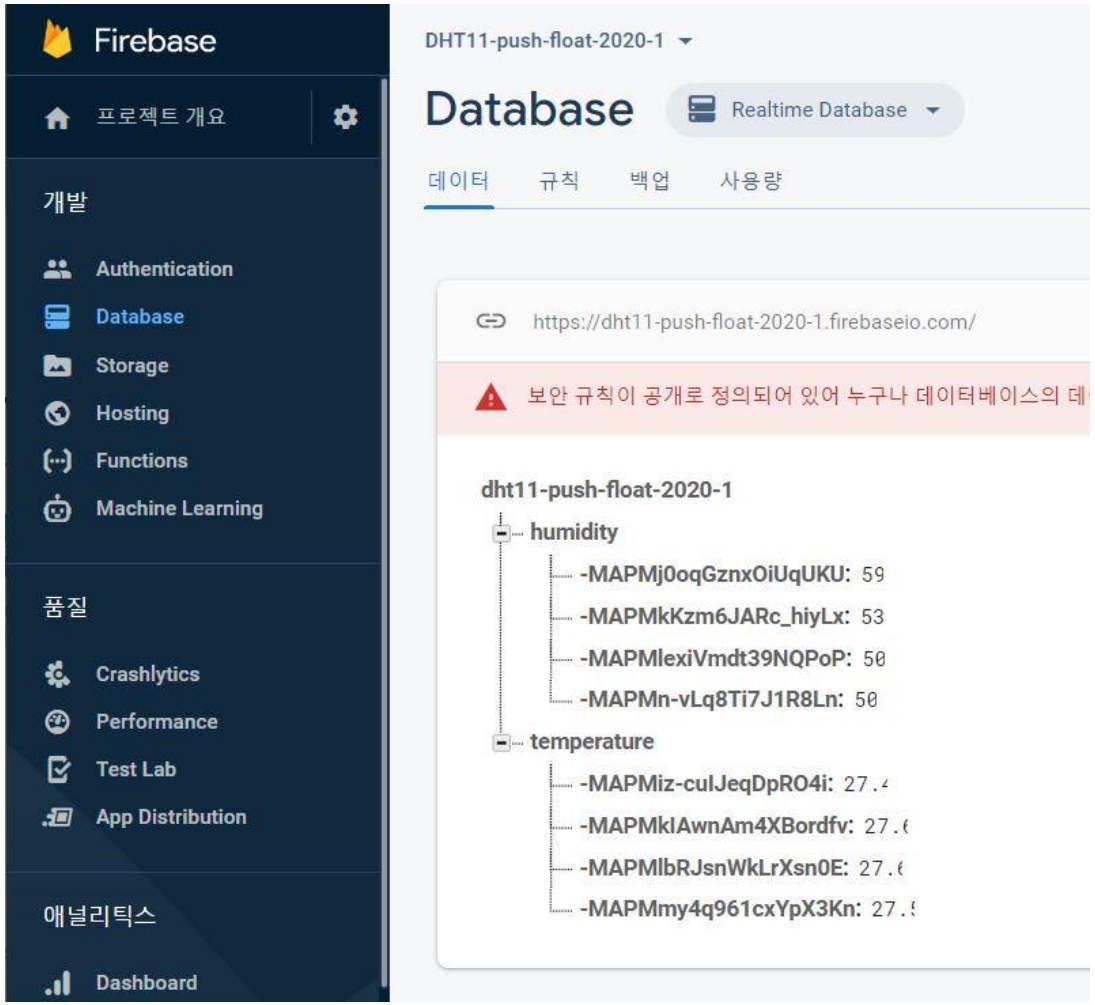
- Firebase 프로젝트 Hello-Firebase를 생성하여, 웹앱에 {“text”, “hello Firebase”}를 출력하게 하시오.
- update 함수를 사용하여 데이터베이스에 저장된 {“text”, “hello Firebase”}가 {“text”, “Good Morning Firebase”}로 수정되도록 하시오.

Firebase 기반 IoT 온습도 모니터링 웹앱 만들기



두개의 객체 사용(pushFloat())

- Firebase – Read Data



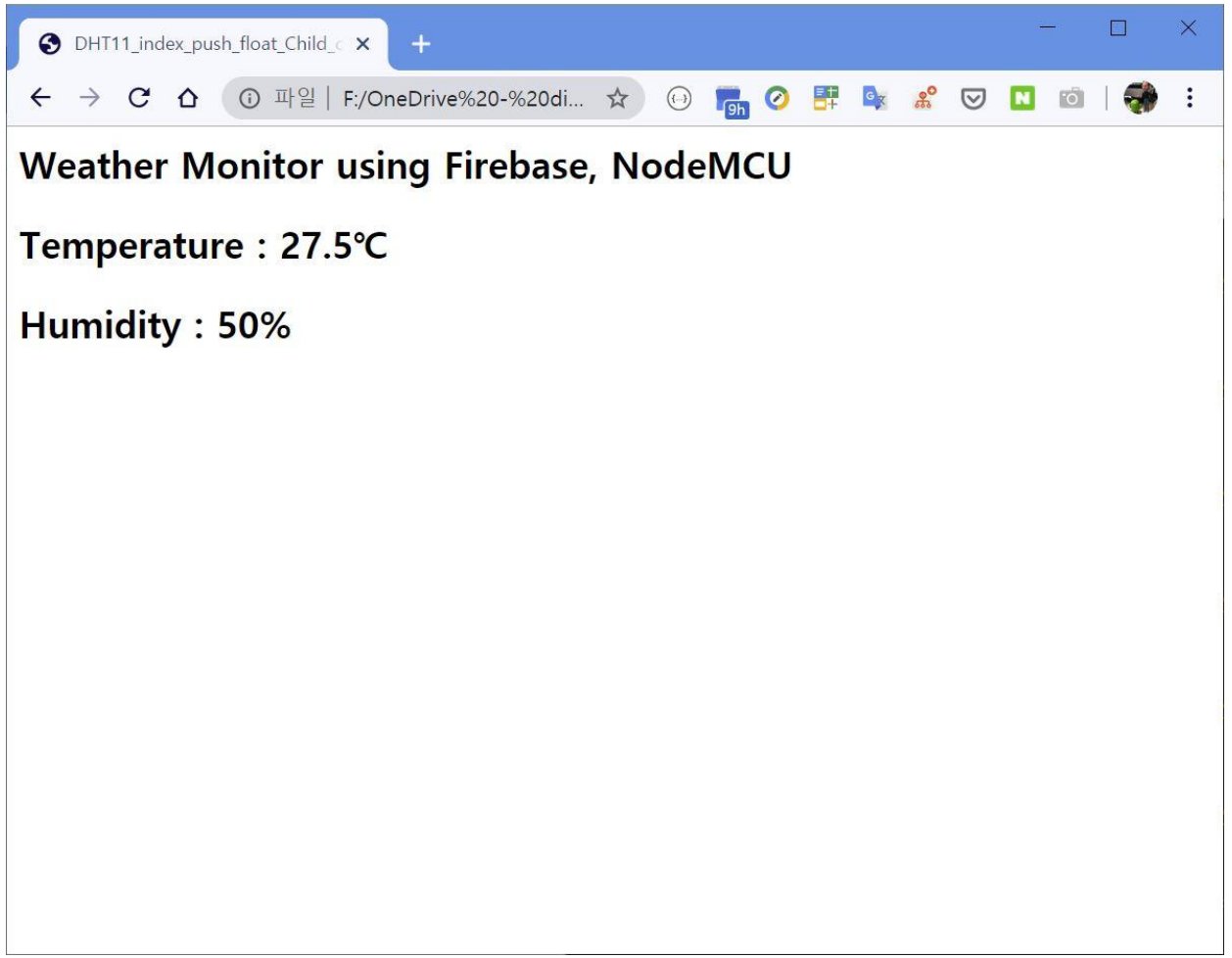

```
// Ardunio Code
void loop() {
  float temp = dht.readTemperature();
  float humi = dht.readHumidity();

  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 + "℃";
        })
    </script>
</body>
</html>
```

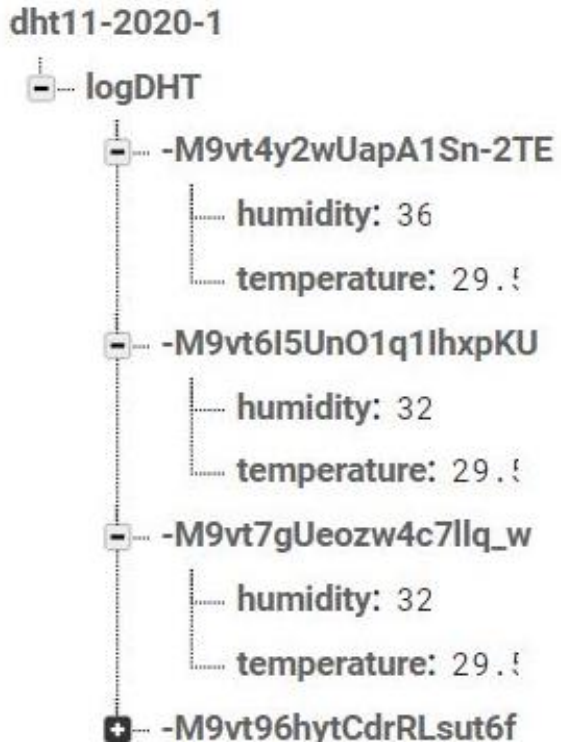
- 전체 소스 코드 : https://github.com/IoT-Lab-02/DHT11-Firebase-Web-PushFloat

웹앱 결과 : temperature, Humidity / pushFloat()



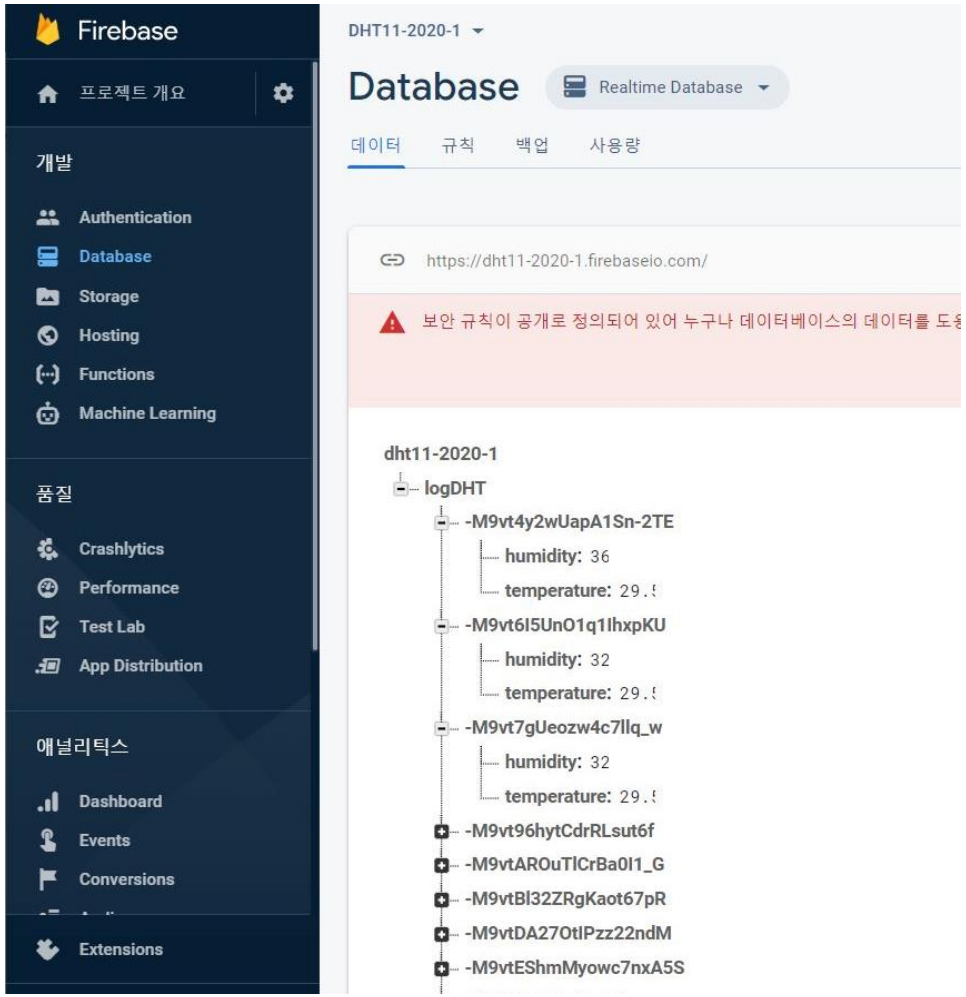
Firebase-Event Types

- 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



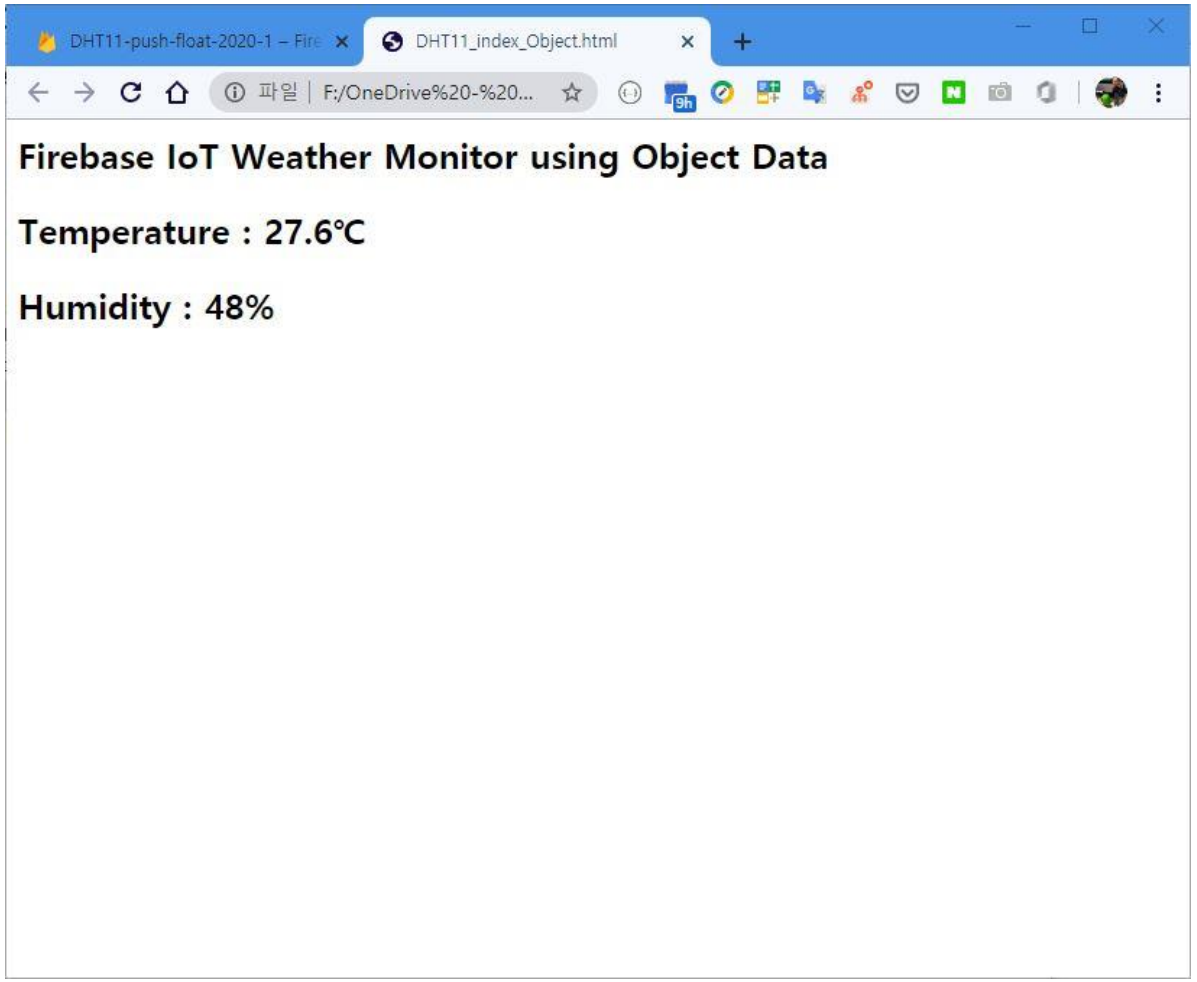

```
// Ardunio Code
void loop() {
  float temp = dht.readTemperature();
  float humi = dht.readHumidity();

  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.innerText = "Humidity : " + newHumi + "%";
       temp.innerText = "Temperature : " + newTemp + "℃";
    </script>
</body>
</html>
```

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

웹앱 결과 : Object/ push()



Firebase Toturial

- https://github.com/IoT-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