

AUTHENTICATION



Authentication with Firebase

- Authentication คือ การตรวจสอบสิทธิ์ของผู้ใช้ก่อนการเข้าใช้งานระบบ
- ซึ่ง Firebase ได้เตรียมวิธีการ authen/authen ไว้มากมาย เพื่อให้เหมาะสมกับการทำงานของเรา
- email and password, Google sign in, Facebook login

- ก่อนที่เราจะทำการ authenauthen ได้ นั้น เราจำเป็นต้องไปกำหนด configuration เสียก่อน
- ในที่นี้ เราต้องการเรียกใช้ Authentication เราจึงต้องทำการ install component firebase auth.js เสียก่อน
- ให้ทำการเพิ่ม firebase auth.js ต่อจาก component เดิม

```
<script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-auth.js"></script>
```

```
<head>
  <meta charset="UTF-8">
  <title>Authentication with Firebase</title>

  <script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-app.js"></script>
  <script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-firestore.js"></script>
  <script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-auth.js"></script>

</head>
```

ขั้นตอนการใช้ Firebase Authentication

การใช้งาน firebase authentication มีขั้นตอนการใช้งานไม่ยากมีเพียง 4 ขั้นตอนเท่านั้น

1. เริ่มต้น ให้ทำการเชื่อมต่อไปยัง authentication ของ firebase ด้วยการใช้คำสั่ง

`firebase.auth`

2. การสร้าง user ใหม่ด้วย `.createUserWithEmailAndPassword` (email, pass)

3. การ sign in สำหรับคนที่มี user อยู่แล้ว `.signInWithEmailAndPassword` (email, password)

4. การ sign out ด้วย `.signOut`

นอกจากนี้ เรายังสามารถติดตามสถานะของการ sign in ได้อีกด้วย

2. Create a user

- เราสามารถทำการเพิ่มผู้ใช้ใหม่ โดยการเรียก
- `createUserWithEmailAndPassword()`
- มีการคืนค่า promise กลับมาด้วย
- `createUserWithEmailAndPassword(email, password)` ต้องการ parameter 2 ตัว คือ
 - email คือ email เพื่อใช้เข้าสู่ระบบ
 - password คือ password เพื่อใช้เข้าสู่ระบบ
- หากเพิ่มผู้ใช้สำเร็จ จะมีการคืนค่ารายละเอียด กลับคืนมาให้
- หากใส่รูปแบบ หรือ email ซ้ำ จะไม่สามารถเพิ่มข้อมูลผู้ใช้ได้

```
firebase
  .auth()
  .createUserWithEmailAndPassword(email, password)
  .then(function(user) {
    console.log('User Created')
  })
  .catch(function(error) {
    var errorCode = error.code;
    var errorMessage = error.message;
    console.log('Error', error.code, error.message);
  });
```

3. Sign in existing users

- รูปแบบการใช้งานเหมือนกับ createUserWithEmailAndPassword
- แต่เปลี่ยนไปใช้ signInWithEmailAndPassword

```
firebase
    .auth()
    .signInWithEmailAndPassword(email, password)
    .then(function(user) {
        console.log('User Created')
    })
    .catch(function(error) {
        var errorCode = error.code;
        var errorMessage = error.message;
        console.log('Error', error.code, error.message);
    });
```

4. Sign out

- เพียงเรียกใช้ `.signOut ()` ก็เรียบร้อยแล้ว

```
firebase  
  .auth ()  
  .signOut ()
```


Get the currently signed in user

- หากเราต้องการดูว่า ณ ตอนนี้มีคน sign in อยู่หรือไม่
- Firebase แนะนำให้ใช้ onAuthStateChanged คอยสอดส่องสถานะของผู้ใช้

```
firebase.auth().onAuthStateChanged(function (user) {  
    if (user) {  
        // User is Signed in.  
    }else{  
        // No user is signed in.  
    }  
});
```

Get the currently signed in user

นอกจากนี้ เรายังสามารถใช้ currentUser เพื่อดูว่า มีคน sign in อยู่หรือไม่ได้ด้วยเช่นกัน

```
var user = firebase.auth().currentUser;  
  
if (user) {  
    // User is Signed in.  
}else{  
    // No user is signed in.  
}
```

Get a user's profile

- Firebase ได้เตรียมข้อมูลอื่น เพื่อให้เราสามารถเก็บข้อมูลของผู้ใช้ได้มากแค่ email และ password

```
var user = firebase.auth().currentUser;
var name, email, photoUrl, uid, emailVerified;

if (user != null) {
    name = user.displayName;
    email = user.email;
    photoUrl = user.photoURL;
    emailVerified = user.emailVerified;
    uid = user.uid;
}
```

ในการเพิ่มข้อมูลผู้ใช้ครั้งแรก ค่าต่าง ๆ จะเป็นค่า default

Update a user's profile

- เราสามารถแก้ไข profile ของผู้ใช้ได้ ผ่านทาง updateProfile

```
var user = firebase.auth().currentUser;

user.updateProfile({
  displayName: "Bellkung",
  photoURL: "http://www.it.mut.ac.th/images/personal/bell.jpg"
}).then(function() {
  console.log('Update Successful.')
})
.catch(function(error) {
  console.log('An error happened. ', error.message);
});
```

Authentication Example

- 01-auth.html

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title>Authentication with Firebase</title>
6
7      <script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-app.js"></script>
8      <script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-firestore.js"></script>
9      <script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-auth.js"></script>
10
11    </head>
12    <body>
13      <input type="text" id="emailText" placeholder="Email">
14      <br>
15      <input type="password" id="passwordText" placeholder="Password">
16      <br>
17      <button id="signInButton">Sign In</button>
18      <button id="signUpButton">Sign Up</button>
19      <button id="signOutButton">Sign Out</button>
20
21      <script src="01-auth.js"></script>
22    </body>
23  </html>
```

- 01-auth.js

```
// Your web app's Firebase configuration
var firebaseConfig = {
  apiKey: "AIzaSyD5gqz_VK_sXupiVizX8fc60Azs12mo4-8",
  authDomain: "my-project-b8ca5.firebaseio.com",
  projectId: "my-project-b8ca5",
  storageBucket: "my-project-b8ca5.appspot.com",
  messagingSenderId: "866862940608",
  appId: "1:866862940608:web:95db1d1039685a3f960825"
};
// Initialize Firebase
firebase.initializeApp(firebaseConfig);

var database = firebase.firestore();
```

- 01-auth.js

ทำการอ้างอิงไปยัง element ที่เราต้องการ

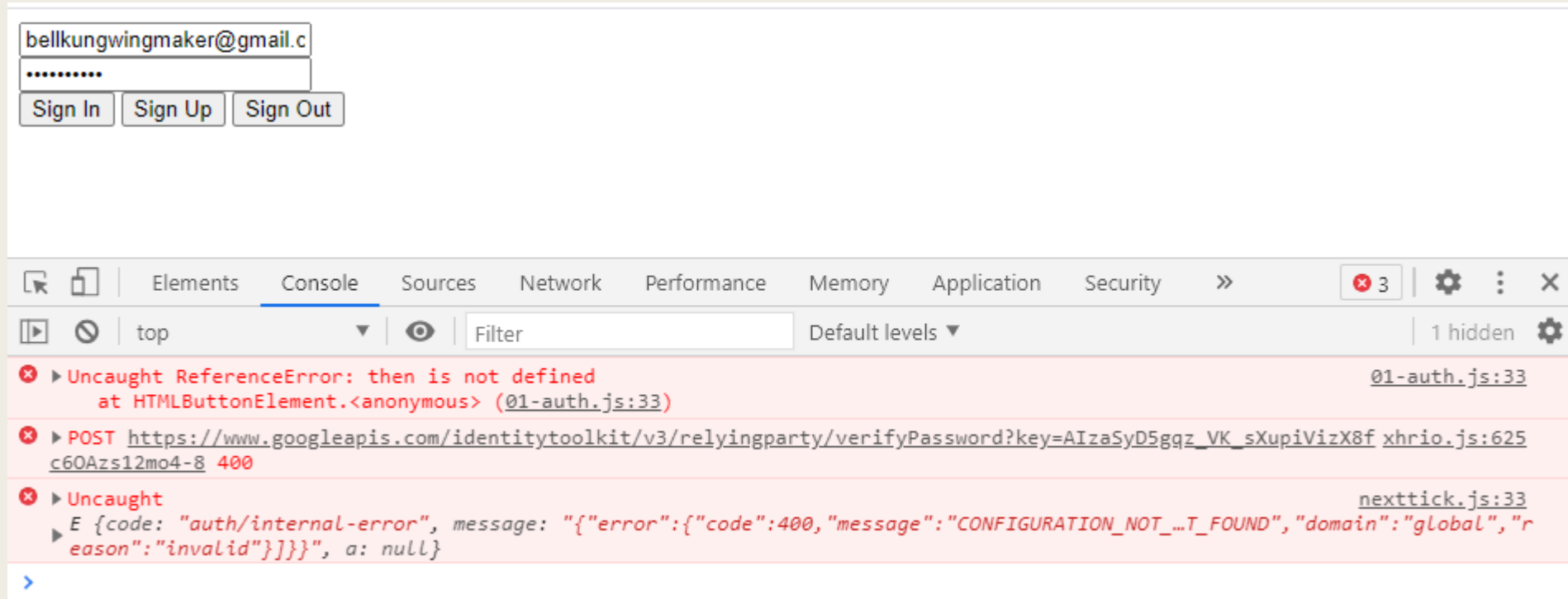
```
var database = firebase.firestore();  
// Get Element  
const emailText = document.querySelector("#emailText");  
const passwordText = document.querySelector("#passwordText");  
const signInButton = document.querySelector("#signInButton");  
const signUpButton = document.querySelector("#signUpButton");  
const signOutButton = document.querySelector("#signOutButton");
```

- 01-auth.js

ทำปุ่ม Sign In

```
21 signInButton.addEventListener("click", function () {
22     // Get email and password
23     const email = emailText.value;
24     const pass = passwordText.value;
25
26     // อ้างอิงไปยัง auth() ผ่าน firebase
27
28     const auth = firebase.auth();
29
30     // ทำการ sign in ผ่าน signInWithEmailAndPassword
31     auth
32         .signInWithEmailAndPassword(email, pass)
33         .then(function (user) {
34             console.log(user);
35         })
36         .catch(function (error) {
37             console.log('Got an error', error.message);
38         });
39     });
```

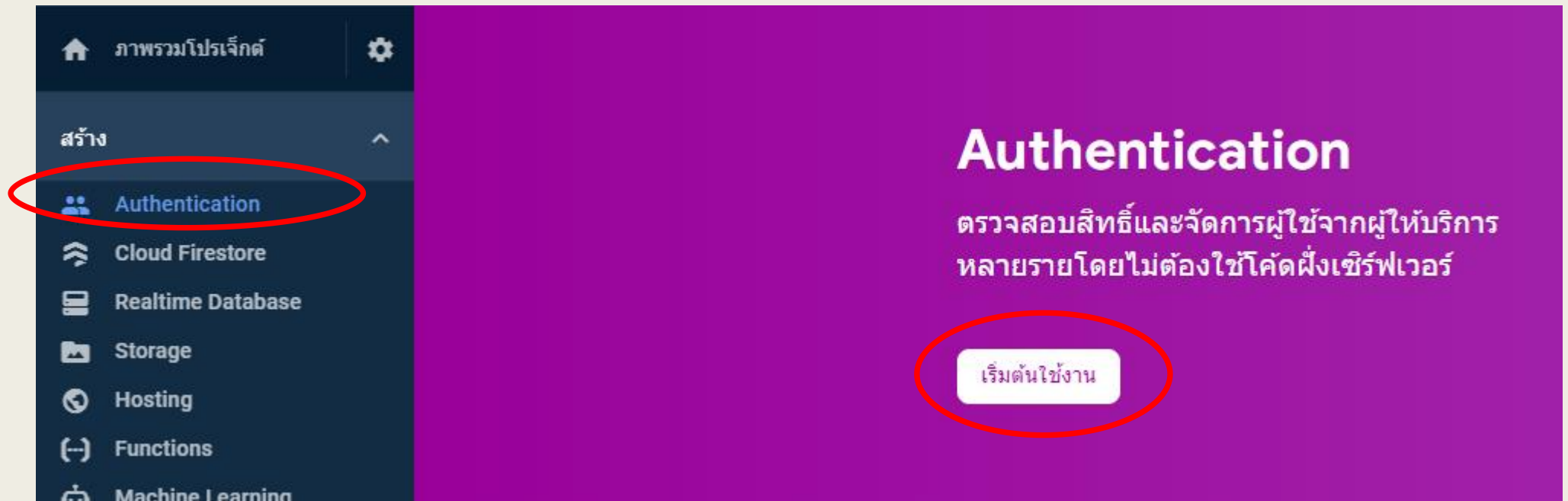

ทดลอง Sign in



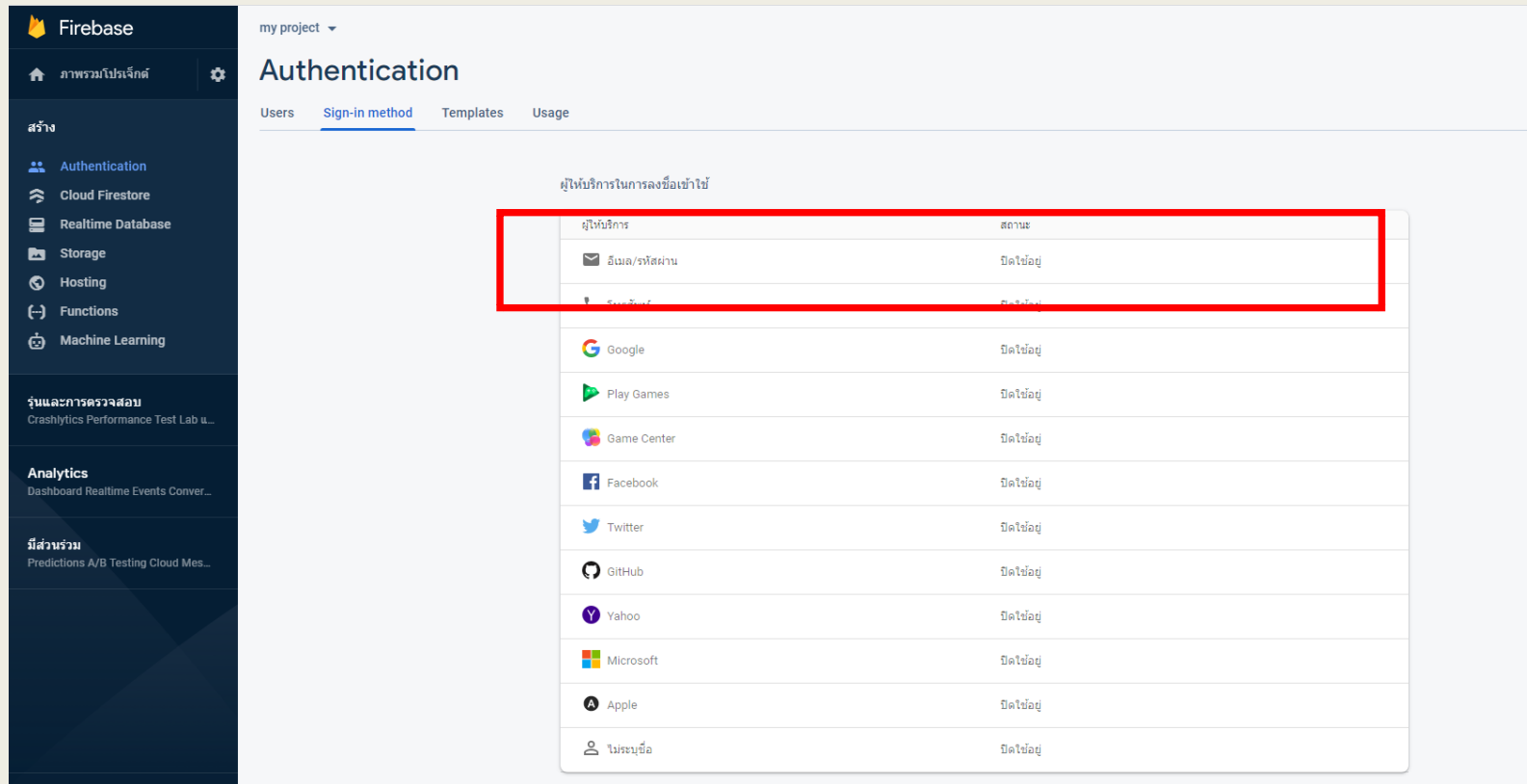
- Firebase แจ้งว่า ยังไม่มี user คนนี้
- เราจะไปทำ Sign Up กันต่อ

เราจำเป็นต้องเปิดใช้งาน auth

- ไปที่ Console
- กดที่ Authentication



กดเปิดใช้งาน



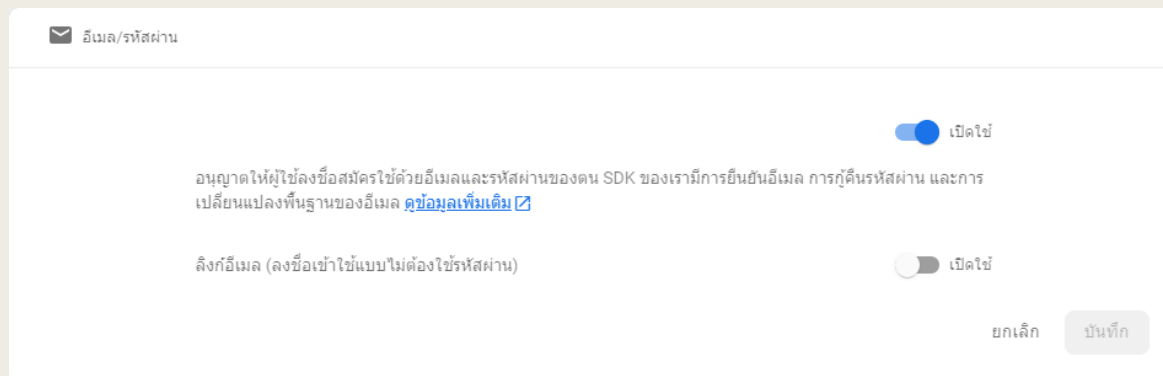
my project ▾

Authentication

Users Sign-in method Templates Usage

ผู้ให้บริการในการลงชื่อเข้าใช้

ผู้ให้บริการ	สถานะ
อีเมล/รหัสผ่าน	ปิดใช้งาน
Google	เปิดใช้งาน
Play Games	เปิดใช้งาน
Game Center	เปิดใช้งาน
Facebook	เปิดใช้งาน
Twitter	เปิดใช้งาน
GitHub	เปิดใช้งาน
Yahoo	เปิดใช้งาน
Microsoft	เปิดใช้งาน
Apple	เปิดใช้งาน
ไม่ระบุชื่อ	เปิดใช้งาน



อีเมล/รหัสผ่าน

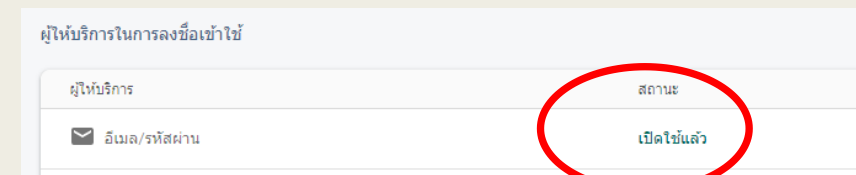
เปิดใช้

อนุญาตให้ผู้ลงชื่อสมัครใช้ด้วยอีเมลและรหัสผ่านของคุณ SDK ของเรามีการยืนยันอีเมล การกู้คืนรหัสผ่าน และการเปลี่ยนแปลงพื้นฐานของอีเมล [ดูข้อมูลเพิ่มเติม](#)

ลิงก์อีเมล (ลงชื่อเข้าใช้แบบไม่ต้องใช้รหัสผ่าน)

เปิดใช้

ยกเลิก บันทึก



ผู้ให้บริการในการลงชื่อเข้าใช้

ผู้ให้บริการ	สถานะ
อีเมล/รหัสผ่าน	เปิดใช้แล้ว

- 01-auth.js

ทำปุ่ม Sign Up

```
41 signUpButton.addEventListener("click", function () {
42
43     const email = emailText.value;
44     const pass = passwordText.value;
45     const auth = firebase.auth();
46
47     // sing up
48     auth
49         .createUserWithEmailAndPassword(email, pass)
50         .then(function(user) {
51             console.log("user, ", user);
52         })
53         .catch(function(error) {
54             console.log('Got an error', error.message);
55         });
56 });
```

bellkungwingmaker@gmail.c

861px × 150px

.....

Sign In

Sign Up

Sign Out

Elements

Console

Sources

Network

Performance

Memory

Application

Security

»

⚙

⋮

✕

🔍

🚫

top

▼

👁

Filter

Default levels ▼

1 hidden ⚙

user,

01-auth.js:51

▼ {user: hh, credential: null, additionalUserInfo: Hr, operationType: "signIn"} ⓘ

▶ additionalUserInfo: Hr {providerId: "password", isNewUser: true}

credential: null

operationType: "signIn"

▶ user: hh {j: Array(0), l: "AIzaSyD5gqz_VK_sXupiVizX8fc60Azs12mo4-8", m: "[DEFAULT]", s: "my-project-b8ca5.f...

▶ __proto__: Object

>

Authentication

Users

Sign-in method

Templates

Usage

✦ สร้างต้นแบบและทดสอบจากต้นทางถึงปลายทางด้วยชุดโปรแกรมจำลองภายใน ซึ่งตอนนี้มีการตรวจสอบสิทธิ์ Firebase แล้ว

เริ่มต้นใช้งาน

🔍 ค้นหาตามอีเมล หมายเลขโทรศัพท์ หรือ UID ของผู้ใช้

เพิ่มผู้ใช้

🔄

⋮

สถานะ	ผู้ให้บริการ	สร้างแล้ว	วันที่ลงชื่อเข้าใช้	UID ของผู้ใช้ ↑
bellkungwingmaker@gmail.c...	📧	10 ก.พ. 2021	10 ก.พ. 2021	vpQYKZkeGfatiLHacXfl5XvUF2P2

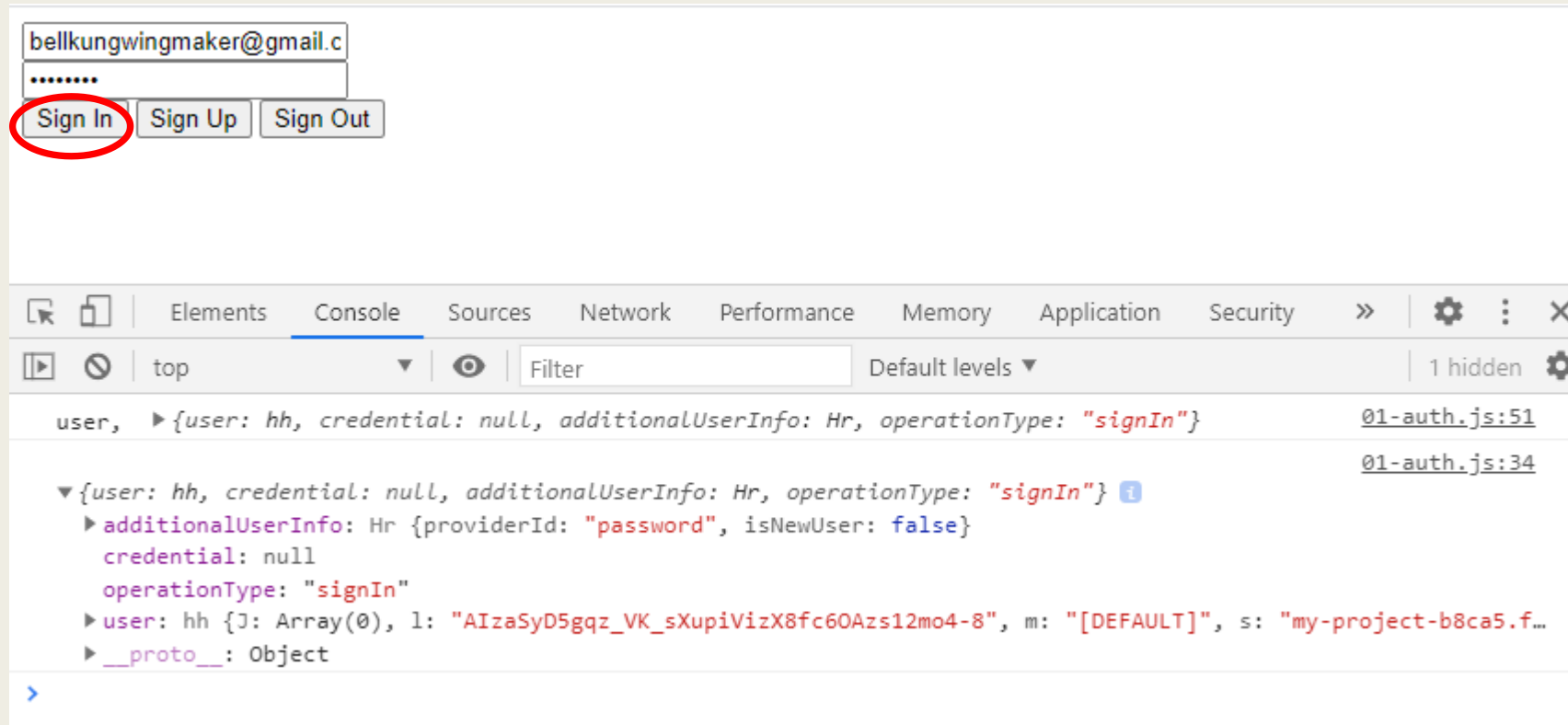
แถวต่อหน้า: 50

1 - 1 of 1

⏪

⏩

ทดลอง Sign In ด้วย user ที่เพิ่งสร้างไป



สามารถ Sign In ได้แล้ว

ทำปุ่ม Sign Out

- 01-auth.js

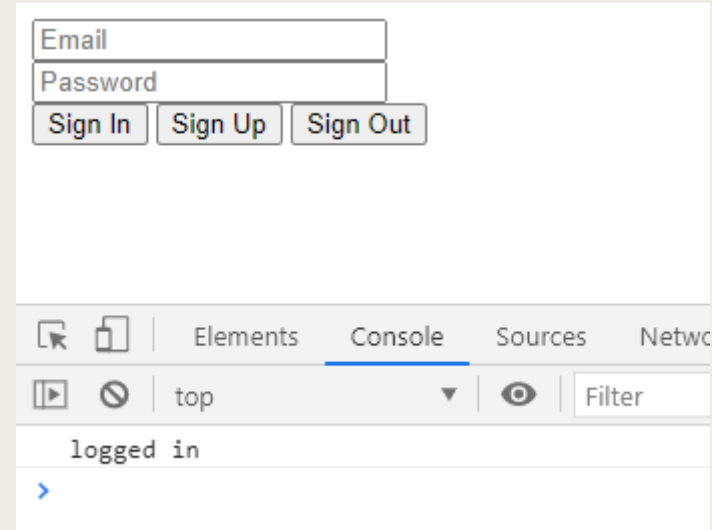
```
58 | signOutButton.addEventListener("click", function () {  
59 |     |     firebase  
60 |     |         .auth()  
61 |     |         .signOut();  
62 | }) ;
```

ตรวจสอบว่า sign in อยู่หรือไม่

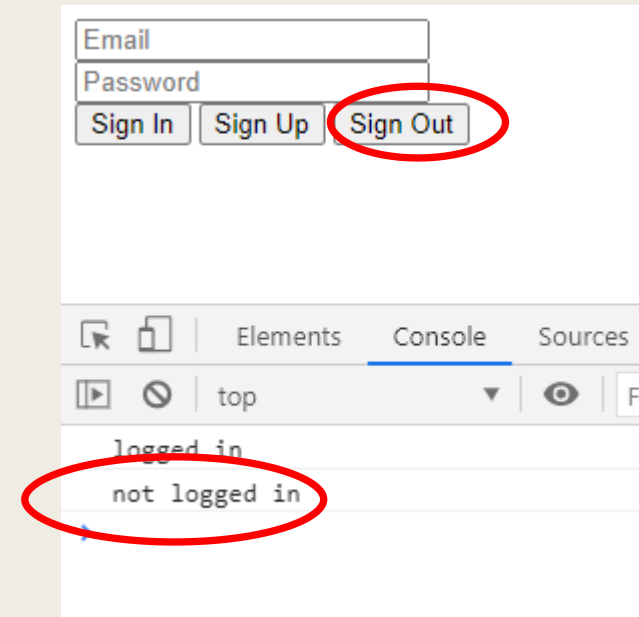
- หากเราต้องการตรวจสอบว่า ณ ตอนนี้ มี user ได้ทำการ Sign in หรือไม่ เราสามารถเขียนคำสั่งโดยใช้ onAuthStateChanged ได้ (Realtime)
- เขียนโดยไม่ต้องอยู่ภายใน function ใด
- 01-auth.js

```
64 //add a realtime listener
65 firebase
66     .auth()
67     .onAuthStateChanged(function (user) {
68         if (user) {
69             console.log("logged in");
70         }else{
71             console.log("not logged in");
72         }
73     });
```


หากเรา Sign In อยู่ จะแสดงข้อความ logged in
ใน console



ทดลองกดปุ่ม Sign Out
สถานะก็จะเปลี่ยนไป



ดู Profile ของ user

- เรายังสามารถดู profile ของ user ที่ sign in ได้ด้วย
- 01-auth.html

```
17 <button id="signInButton">Sign In</button>
18 <button id="signUpButton">Sign Up</button>
19 <button id="signOutButton">Sign Out</button>
20 <button id="getUserProfile">Get User Profile</button>
21
22
23 <script src="01-auth.js"></script>
24 </body>
25
```

- 01-auth.js

```
75 //Get User Profile
76 const getUserProfile = document.querySelector("#getUserProfile");
77 getUserProfile.addEventListener("click", function () {
78     var user = firebase.auth().currentUser;
79
80     if (user) {
81         //user logged in
82         console.log("email: ", user.email);
83         console.log("Display Name: ", user.displayName);
84         console.log("Photo: ", user.photoURL);
85     }
86 });
```

แก้ไข Profile ของ user

- เรายังสามารถแก้ไข profile ของ user ได้ด้วย
- 01-auth.html

```
12 <body>
13     <input type="text" id="emailText" placeholder="Email">
14     <br>
15     <input type="password" id="passwordText" placeholder="Password">
16     <br>
17     <button id="signInButton">Sign In</button>
18     <button id="signUpButton">Sign Up</button>
19     <button id="signOutButton">Sign Out</button>
20     <button id="getUserProfile">Get User Profile</button>
21     <br><br>
22     <input type="text" id="displayNameText">
23     <button id="setUserProfile">Set User Profile</button>
24
25
26 <script src="01-auth.js"></script>
```

- 01-auth.js

```
88 //Set User Profile
89 const setUserProfile = document.querySelector("#setUserProfile");
90 const displayNameText = document.querySelector("#displayNameText");
91
92 setUserProfile.addEventListener("click", function () {
93     var displayname = displayNameText.value;
94     var user = firebase.auth().currentUser;
95
96     if (user) {
97         //user logged in
98         user
99         .updateProfile({
100             displayName: displayname
101         });
102     }
103 });
```

ผลลัพธ์

The screenshot displays a web application interface at the top and a browser's developer console at the bottom. In the interface, the email field contains 'bellkungwingmaker@gmail.c' and the password field is masked with dots. The 'Sign In' and 'Get User Profile' buttons are circled in red. Below the interface, the browser console shows a sequence of log messages: 'not logged in', a JSON object for the sign-in operation, 'logged in', and a detailed user profile object. The user profile object is highlighted with a red box and contains the following data:

email:	bellkungwingmaker@gmail.com	01-auth.js:82
Display Name:	null	01-auth.js:83
Photo:	null	01-auth.js:84

ผลลัพธ์

The screenshot displays a web application interface at the top and a browser's developer console at the bottom. The interface includes input fields for email (bellkungwingmaker@gmail.c) and password (masked with dots), and buttons for 'Sign In', 'Sign Up', 'Sign Out', 'Get User Profile', and 'Set User Profile'. The 'Get User Profile' button is circled in red and labeled with a red '3'. Below the input fields, the name 'Bellkung' is entered in a text box, which is circled in red and labeled with a red '1'. The 'Set User Profile' button is also circled in red and labeled with a red '2'. The browser console shows a series of log messages. The first message is 'not logged in' from 01-auth.js:71. The second is a JSON object: {user: hh, credential: null, additionalUserInfo: Hr, operationType: "signIn"} from 01-auth.js:34. The third is 'logged in' from 01-auth.js:69. The subsequent messages show user profile data: email: bellkungwingmaker@gmail.com (01-auth.js:82), Display Name: null (01-auth.js:83), Photo: null (01-auth.js:84), and email: bellkungwingmaker@gmail.com (01-auth.js:82). The 'Display Name: Bellkung' message is highlighted with a red box. The final message is 'Photo: null' from 01-auth.js:84.

1

2

3

not logged in 01-auth.js:71

{user: hh, credential: null, additionalUserInfo: Hr, operationType: "signIn"} 01-auth.js:34

logged in 01-auth.js:69

email: bellkungwingmaker@gmail.com 01-auth.js:82

Display Name: null 01-auth.js:83

Photo: null 01-auth.js:84

email: bellkungwingmaker@gmail.com 01-auth.js:82

Display Name: Bellkung 01-auth.js:83

Photo: null 01-auth.js:84

ส่ง verification email

- Firebase เตรียม method สำหรับส่งเมลไปยังเมลที่ register เข้ามา
- เพื่อทำการยืนยัน email นั้น ว่ามีอยู่จริงหรือไม่
- การทำงานไม่ยาก เพียงแต่ต้องทำงานก็ต่อเมื่อได้ทำการ create user เรียบร้อยแล้ว
- โดยปกติแล้ว เมื่อทำการ create user เรียบร้อยแล้ว firebase จะทำการ sign in โดยอัตโนมัติ

```
var user = firebase.auth().currentUser;

user.sendEmailVerification().then(function() {
    //Email Sent.
}) .catch(function(error) {
    // error
});
```


ตรวจสอบว่าได้ทำการ verified email email แล้วหรือยัง

- เราจะสามารถตรวจสอบว่า email นี้ได้รับการยืนยันแล้วหรือไม่ ก็ต่อเมื่อ ได้มีการ sign in
- เข้าสู่ระบบแล้ว
- Firebase มี property สำหรับตรวจสอบว่า ได้มีการยืนยัน email แล้วหรือไม่ คือ **emailVerified**
 - ค่าเป็น true เมื่อได้ทำการยืนยัน email เรียบร้อยแล้ว
 - ค่าเป็น false เมื่อยังไม่ได้ทำการยืนยัน email เรียบร้อยแล้ว

```
var user = firebase.auth().currentUser;  
console.log('Verified: ', user.emailVerified)
```

Quiz

ให้นักศึกษาทำการแก้ไขคำสั่งจากตัวอย่าง
โดยให้มีการทำงานดังนี้

1. กรอกข้อมูล
2. เมื่อกด Sign Up ให้ เพิ่มข้อมูล user โดยใช้ authentication ของ firebase
3. กำหนด displayname โดยการเอา
firstname และ lastname มาต่อกัน

