
Firebase User Authentication

1

Firebase

A collection of many products

- **Cloud Firestore** (beta since 2017, GA since 2019)
- **Authentication**
- **Cloud Storage**
- **Realtime DB** (beta since 2012, GA since 2014?)
- ML Kit
- Cloud Functions

2

Setup

```
# Change directory to your project  
  
# Then add the firebase package  
yarn add firebase # Version 9.x
```

```
import {initializeApp} from "firebase/app";  
import {getAuth} from "firebase/auth";  
  
const app = initializeApp(your_firebase_project_config);  
const auth = getAuth();  
  
// Use myAuth in subsequent authentication function calls  
createUserWithEmailAndPassword(auth, "user_email", "user_password")  
sendEmailVerification(auth.currentUser);  
signInWithEmailAndPassword(auth, "user_email", "user_password")  
signInWithPopUp(auth, _____);  
signOut(auth);
```

3

Quick Video Introduction



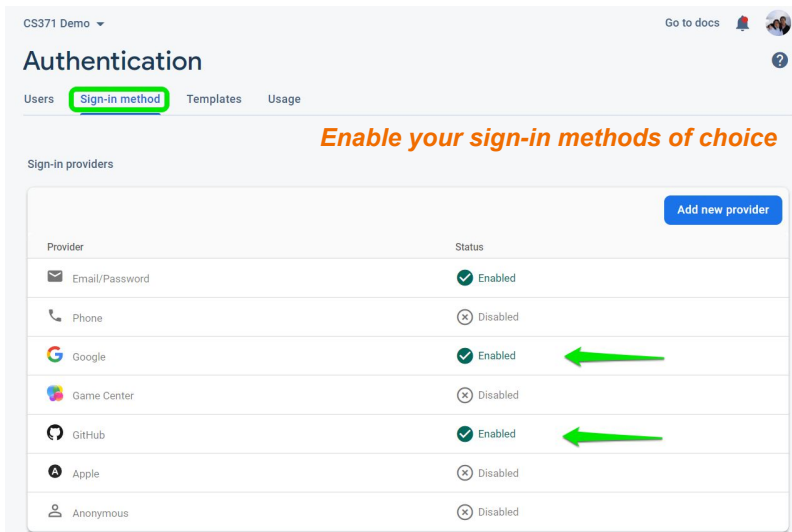
4

Authentication Options

- Email/Password
- Facebook accounts
- GitHub accounts
- Google accounts
- Twitter accounts
- Microsoft accounts
- Yahoo accounts
- Phone numbers
- Online documentation: firebase.auth

5

Authentication Dashboard: Sign-in Providers



The screenshot shows the Firebase Authentication dashboard for a project named 'CS371 Demo'. The 'Sign-in method' tab is selected and highlighted with a green box. A red banner at the top of the dashboard reads 'Enable your sign-in methods of choice'. Below this, a table lists various sign-in providers with their status. Green arrows point to the 'Enabled' status of the Google and GitHub providers.

| Provider | Status |
|----------------|----------|
| Email/Password | Enabled |
| Phone | Disabled |
| Google | Enabled |
| Game Center | Disabled |
| GitHub | Enabled |
| Apple | Disabled |
| Anonymous | Disabled |

6

Authentication Dashboard: Authorized Domains

Authorized domains ⓘ

Add domain

| Authorized domain | Type |
|---------------------------|---------|
| localhost | Default |
| cs371-demo.firebaseio.com | Default |
| 127.0.0.1 | Custom |


7

Authentication Dashboard: Advanced Option

Advanced

One account per email address

Preventing users from creating multiple accounts using the same email address with different authentication providers.

[Learn more](#) 

[Change](#)

Allow multiple accounts per email

(in case your users link their GitHub, Facebook, Apple accounts to the same email)

8

Authentication Functions are async

Use Promise.then or await to handle the result

10

Firestore Auth: Create A New Account

```
import {getAuth, createUserWithEmailAndPassword, UserCredential} from "firebase/auth";

const auth = getAuth();
createUserWithEmailAndPassword(auth, "me@sample.com", "1q2w3e4r5")
  .then((cred: UserCredential) => {
    sendEmailVerification(cred.user);
    console.log("Verification email has been sent to", cred.user?.email);
    auth.signOut();
  })
  .catch((err: any) => {
    console.error("Oops", err);
  });
```

11

Live Demo

12

Firestore Auth: Signin With Email

```
import {getAuth, signInWithEmailAndPassword, UserCredential} from "firebase/auth";

const auth = getAuth();
signInWithEmailAndPassword(auth, "me@sample.com", "1q2w3e4r5")
  .then((cred: UserCredential) => {
    if (cred.user?.emailVerified)
      console.log("Signed in as", cred.user?.email);
    else {
      console.log("Please verify your email first");
      auth.signOut();
    }
  })
  .catch((err: any) => {
    console.error("Oops", err);
  });
```

13

Firestore Auth: Sign In With Providers

```
import { GoogleAuthProvider, UserCredential, signInWithPopup,
  getAuth } from "firebase/auth";

const auth = getAuth();
const provider = new GoogleAuthProvider();

signInWithPopup(auth, provider)
  .then((cred:UserCredential) => {

    console.log("Signed in as", cred.user?.email);
  })
  .catch((err: any) => {
    console.error("Oops", err);
  });
```

14

Monitor Authentication in Background

Use **onAuthStateChanged** if you need to monitor login status in background

```
import {User, getAuth, onAuthStateChanged} from "firebase/auth";

const auth = getAuth();
onAuthStateChanged(auth, (user:User|null) => {
  if (currentUser == null & user != null) {
    currentUser = user;
    /* User just logged in */
  } else if (currentUser != null & user == null) {
    /* User just logged out, do clean up work */
  }
});
```

15

3rd party Account Providers

| Account Provider | Provider Class |
|------------------|----------------------|
| Facebook | FacebookAuthProvider |
| GitHub | GitHubAuthProvider |
| Google | GoogleAuthProvider |
| Phone | PhoneAuthProvider |
| Twitter | TwitterAuthProvider |

Example: GitHub SignIn

Step A: Enable Github Login



Do this step from *Firebase Authentication Dashboard*

Enable

Client ID

Client secret

To complete set up, add this authorization callback URL to your GitHub app configuration. [Learn more](#)

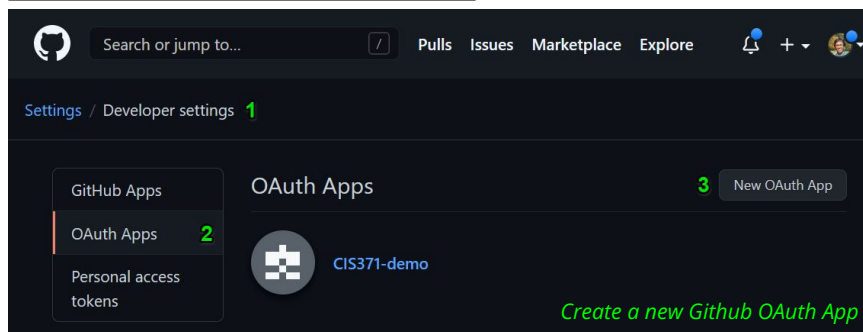
Step A: copy this URL (needed by Github in Step C)

Cancel Save

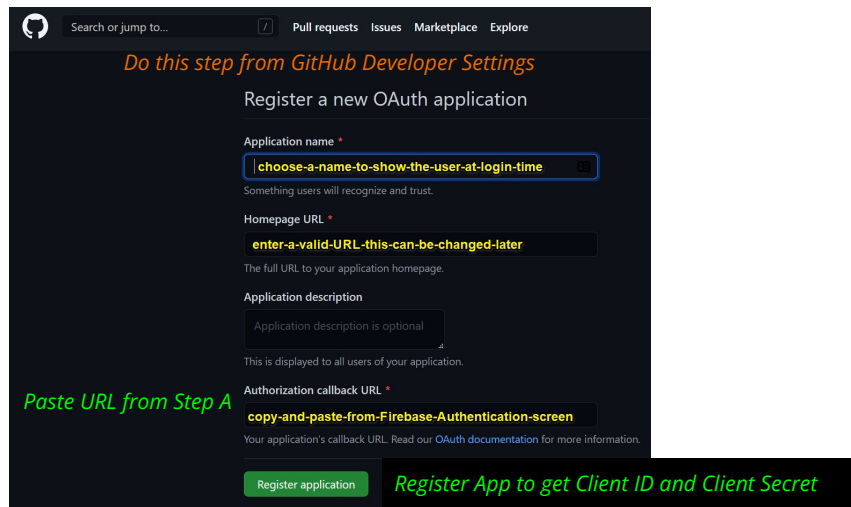
Step B: Create an OAuth App (On Github Settings)

Do this step from *GitHub Developer Settings*

The following GitHub page is under Settings => Developer Setting



Step C: Register OAuth App (on Github Settings)



Do this step from *GitHub Developer Settings*

Register a new OAuth application

Application name *

choose-a-name-to-show-the-user-at-login-time

Something users will recognize and trust.

Homepage URL *

enter-a-valid-URL-this-can-be-changed-later

The full URL to your application homepage.

Application description

Application description is optional

This is displayed to all users of your application.

Authorization callback URL *

copy-and-paste-from-Firebase-Authentication-screen


Your application's callback URL. Read our [OAuth documentation](#) for more information.

Paste URL from Step A

Register application **Register App to get Client ID and Client Secret**

20

Step D: Copy Client ID & Client Secret



GitHub

Do this step from *Firebase Authentication Dashboard*

Enable

Client ID

Client secret

Copy Client ID and Client Secret from Step C

To complete set up, add this authorization callback URL to your GitHub app configuration. [Learn more](#)

`https://cs371-demo.firebaseio.com/_/auth/handler`

Cancel Save

21