CIS 371 Web Application Programming Cloud DataBase Firebase User Authentication



Lecturer: Dr. Yong Zhuang

Based on the original version by Professor Hans Dulimarta.

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?)
- Cloud Messaging
- ML Kit
- Cloud Functions



Setup

yarn init -y yarn add firebase

OR

npm init -y npm install firebase

```
import { createApp } from "vue";
import App from "./App.vue";
import router from "./router";
import { initializeApp } from "firebase/app";
```

```
const firebaseConfig = {
   apiKey: "your-api-key-goes-here",
   authDomain: "your-project-name.firebaseapp.com",
   projectId: "your-project-id",
   storageBucket: "your-project-name.appspot.com",
   messagingSenderId: "xxxxxxxxxx",
};
```

```
initializeApp(firebaseConfig);
createApp(App).use(router).mount("#app")
```

main.ts

import { getAuth, signInWithPopup, createUserWithEmailAndPassword, signInWithEmailAndPassword, sendEmailVerification, signOut, } from "firebase/auth";

login.vue

```
sendLmaIlVerIfication,
signOut,
} from "firebase/auth";
const auth = getAuth();
createUserWithEmailAndPassword(auth, "user_email", "user_password");
sendEmailVerification(auth.currentUser);
signInWithEmailAndPassword(auth, "user_email", "user_password");
signInWithPopUp(auth, ____);
signOut(auth);
```



Authentication Options

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



Authentication Dashboard: Sign-in Providers





Authentication Dashboard: Authorized Domains





Authentication Dashboard: Advanced Option





Authentication Functions are async Use Promise.then or await to handle the result



Firebase 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);
  });
```



Firebase Auth: Signin With Email

```
import {
 getAuth,
 signInWithEmailAndPassword,
 UserCredential,
} from "firebase/auth";
const auth = getAuth();
signInWithUserWithEmailAndPassword(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);
 });
```



Firebase Auth: Sign In With Providers

```
import { getAuth, GoogleAuthProvider, signInWithPopup } from "firebase/auth";
const auth = getAuth();
const provider = new GoogleAuthProvider();
signInWithPopup(auth, provider)
  .then((result) => {
    const cred = GoogleAuthProvider.credentialFromResult(result);
    console.log("Signed in as", cred.user?.email);
  })
  .catch((err: any) => {
                                              Make sure you have added the login provider in
    console.error("Oops", err);
                                              the Authentication Dashboard
  });
```



Monitor Authentication in Background





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

			Configure provider (Step 2 of 2)	
			Q GitHub	Enable
			Client ID	
と Firebase	CIS371 Demo 👻	0 🖻 🌲 💙		
	Authoptication		A client ID is required	
Project Overview	Authentication		Client secret	
Project shortcuts	Users Sign-in method Templates	Usage Settings 😻 Extensions		
奈 Firestore Database			A client secret is required	
Storage	Sign-in providers		To complete set up, add this authorization callback URL to your GitHub	app configuration.
		Add new provider	https://cis371-demo-770dc.firebaseapp.com/_/auth/handler	1
Product categories				2
Duild	Provider	Status		Cancel Save
Build	Email/Password	Enabled		
Release & Monitor 🛛 🗸	G Google	Senabled		

Do this step from Firebase Authentication Dashboard



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

The following GitHub page is under Settings => Developer Setting



Do this step from GitHub Developer Settings



Step D: Copy Client ID & Client Secret





Step D: Copy Client ID & Client Secret

GitHub GitHub				C Enable
Client ID				
Client exerct				
Gnent secret				
To complete set up, a <u>Learn more</u> [2]	add this authorizatio	n callback URL to you	r GitHub app cor	nfiguration.
			_	

Do this step from Firebase Authentication Dashboard





