

Front-End Development



# React & Firebase

**#3** Firebase Auth with Various Providers



# Firebase



Authentication

**Firebase Authentication** provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

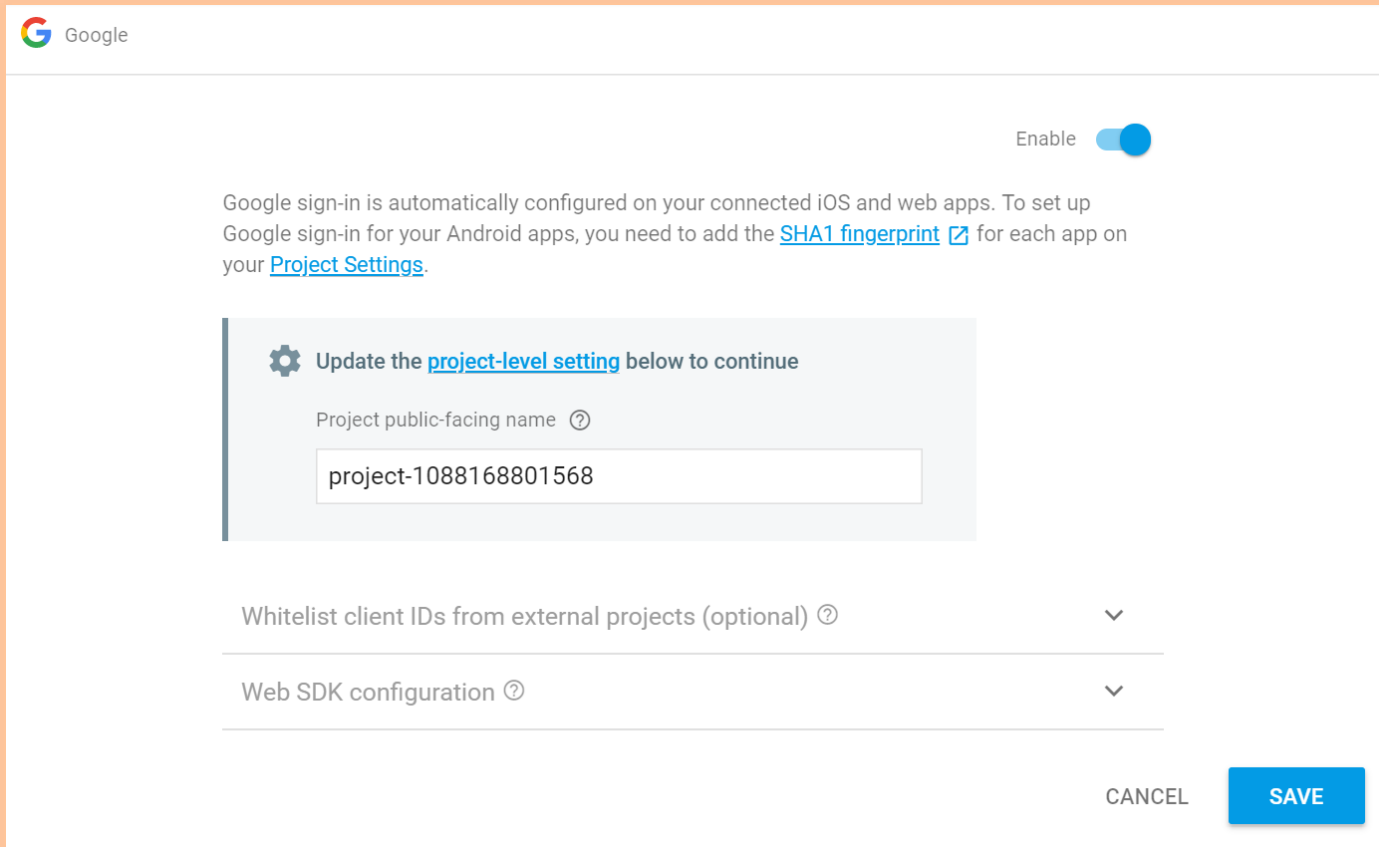
Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.



# How to Setup React and Firebase Authentication with Google Provider?

# #1 Setup Firebase Auth

- Login to **Firestore** Console, choose/create a project then go to **Firestore Authentication**. On **Sign-In Method** tab, enable **Google** sign-in provider then Save it!



The screenshot shows the 'Google' sign-in configuration page in the Firebase console. At the top left is the Google logo. On the right, there is an 'Enable' toggle switch that is turned on. Below this, a paragraph states: 'Google sign-in is automatically configured on your connected iOS and web apps. To set up Google sign-in for your Android apps, you need to add the [SHA1 fingerprint](#) for each app on your [Project Settings](#).' A light blue box contains a gear icon and the text 'Update the [project-level setting](#) below to continue'. Inside this box is a label 'Project public-facing name' with a help icon, followed by a text input field containing 'project-1088168801568'. Below the box are two expandable sections: 'Whitelist client IDs from external projects (optional)' and 'Web SDK configuration', each with a downward arrow and a help icon. At the bottom right are 'CANCEL' and 'SAVE' buttons.

Google

Enable ☒

Google sign-in is automatically configured on your connected iOS and web apps. To set up Google sign-in for your Android apps, you need to add the [SHA1 fingerprint](#) for each app on your [Project Settings](#).

⚙️ Update the [project-level setting](#) below to continue

Project public-facing name ⓘ

project-1088168801568

Whitelist client IDs from external projects (optional) ⓘ

Web SDK configuration ⓘ

CANCEL SAVE

## #2 Firebase Snippet

- Go to **Project Overview**, choose **Add Firebase to Your Web App** then copy its snippet code.

### Add Firebase to your web app

Copy and paste the snippet below at the bottom of your HTML, before other script tags.

```
<script src="https://www.gstatic.com/firebasejs/4.11.0/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyAtQf_vMPce7JTfYQesg2isg8pZftJwIII",
    authDomain: "linyoa-37d9d.firebaseio.com",
    databaseURL: "https://linyoa-37d9d.firebaseio.com",
    projectId: "linyoa-37d9d",
    storageBucket: "linyoa-37d9d.appspot.com",
    messagingSenderId: "922933748183"
  };
  firebase.initializeApp(config);
</script>
```

COPY

# #3 Create/Prepare Your React App

- Install create-react-app utility globally on your PC (if you have not done yet before)

```
$ npm install -g create-react-app
```

- Create a new react app or prepare yours

```
$ create-react-app lin_auth
```

```
$ cd lin_auth
```

- Install Firebase packages needed

```
$ npm install -g firebase-tools
```

```
$ npm install firebase
```

```
import firebase from 'firebase';

var config = {
  apiKey: "a1b2c3d4e5f6g7h8i9j0",
  authDomain: "lintang-auth.firebaseio.com",
  databaseURL: "https://lintang-auth.firebaseio.com",
  projectId: "lintang-auth",
  storageBucket: "lintang-auth.appspot.com",
  messagingSenderId: "1234567890"
};

firebase.initializeApp(config);
export const ref = firebase.database().ref();
export const auth = firebase.auth;
export const provider = new
firebase.auth.GoogleAuthProvider();
```

```
import React, { Component } from 'react';
import { provider, auth } from './fire';

class App extends Component {
  constructor(){
    super();
    this.state = {user: null}
  }
  async login() {
    const result = await auth().signInWithPopup(provider);
    this.setState({user: result.user});
  }
  async logout(){
    await auth().signOut()
    this.setState({user: null});
  }
}
```

• • • • •

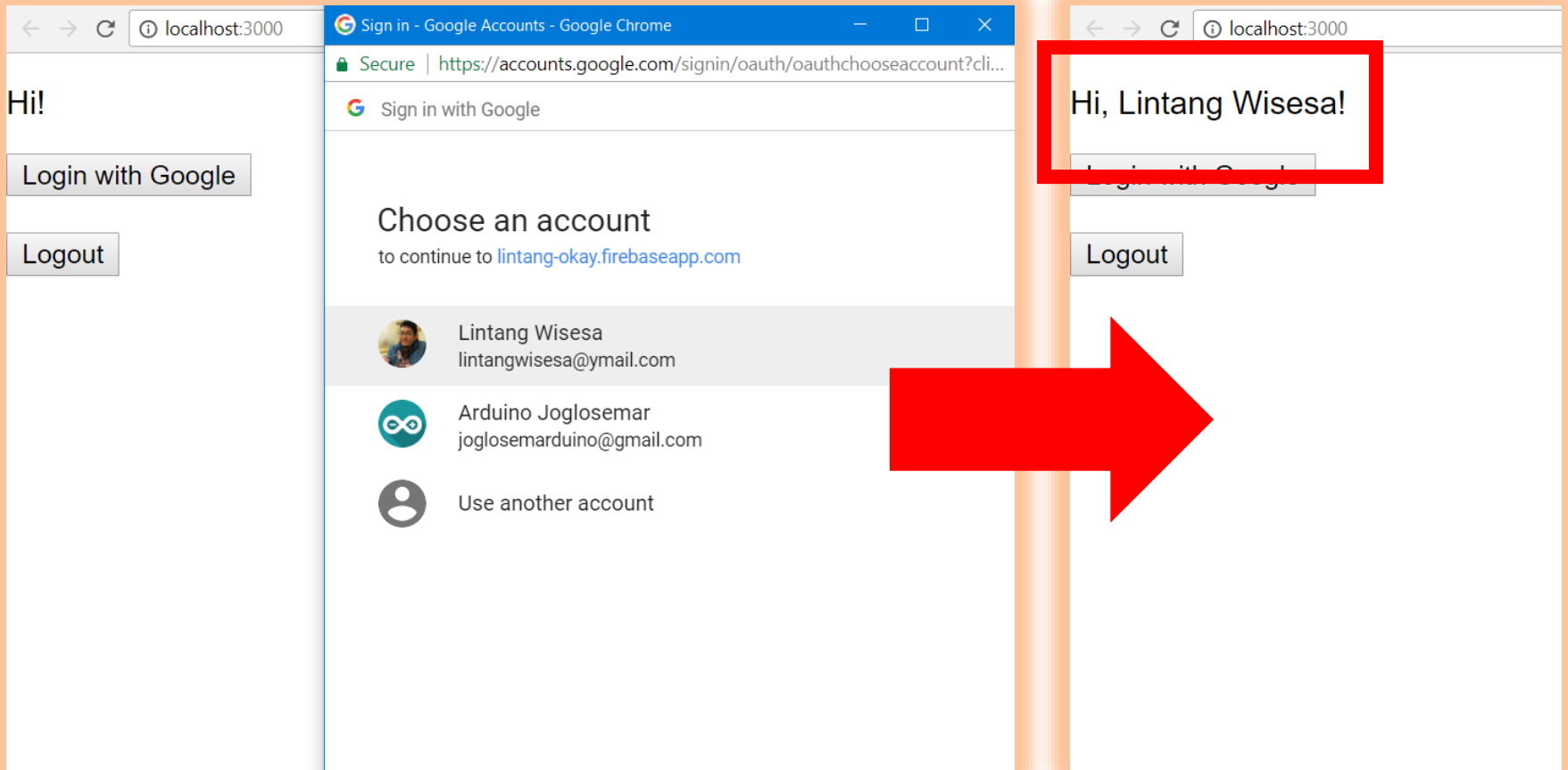


• • • • •

```
render() {  
  return (  
    <div>  
      <p>  
        {this.state.user ? `Hi,  
        ${this.state.user.displayName}!` : 'Hi!'}  
      </p>  
      <button onClick={this.login.bind(this)}>  
        Login with Google  
      </button>  
      <button onClick={this.logout.bind(this)}>  
        Logout  
      </button>  
    </div>  
  );  
}
```

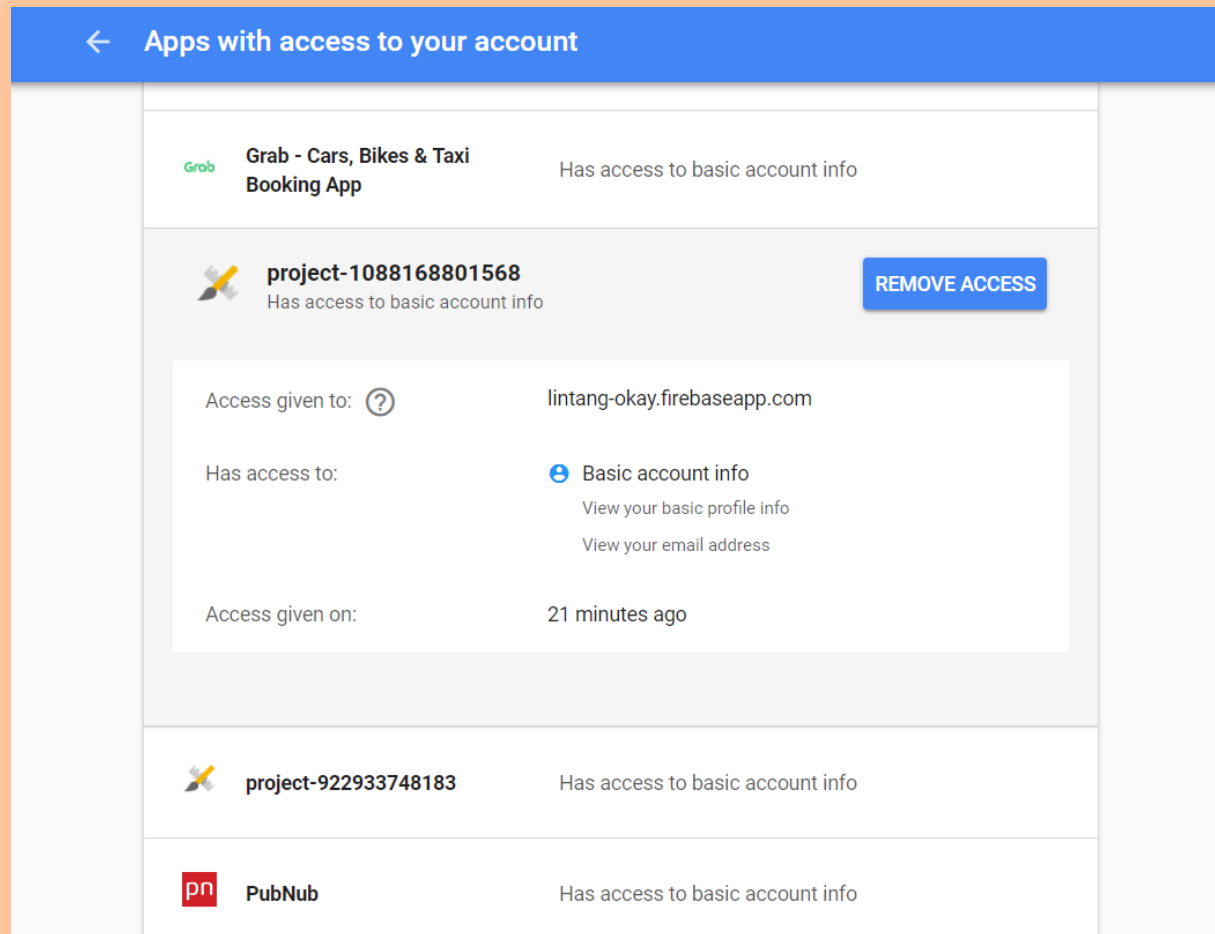
export default App;

# React, Firebase Auth With Google Provider



# Revoke Google Permission

- Go to <https://myaccount.google.com/permissions> then simply remove its access.

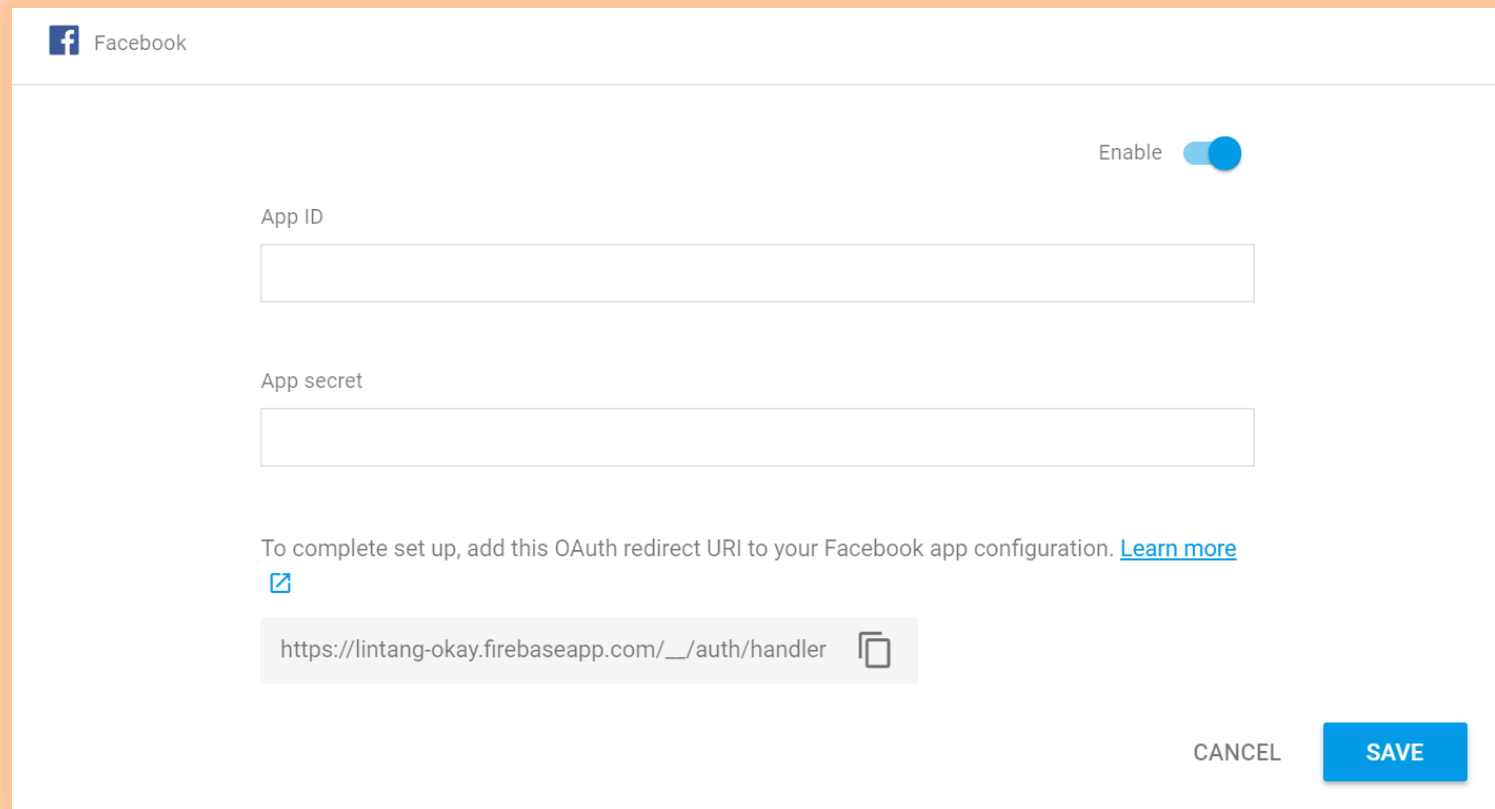




# How to Setup React and Firebase Authentication with Facebook Provider?

# #1 Setup Firebase Auth

- Login to ***Firebase*** Console, choose/create a project then go to ***Firebase Authentication***. On ***Sign-In Method*** tab, enable ***Facebook*** sign-in provider!



The screenshot shows the Facebook sign-in provider configuration page in the Firebase Authentication console. At the top left is the Facebook logo and the word "Facebook". On the right, there is an "Enable" toggle switch that is turned on. Below this are two input fields: "App ID" and "App secret", both of which are currently empty. Further down, a text instruction reads: "To complete set up, add this OAuth redirect URI to your Facebook app configuration. [Learn more](#)". Below the text is a checkbox that is checked. Underneath the checkbox is a text box containing the URL "https://lintang-okay.firebaseio.com/\_/auth/handler" followed by a copy icon. At the bottom right, there are two buttons: "CANCEL" and "SAVE".

## #2 Facebook Dev

- Login to Facebook, then go to Facebook Dev at <https://developers.facebook.com>. Click on **My Apps** → **Add a new App**, insert Display Name & Contact Email then click **Create App ID**.

### Create a New App ID

Get started integrating Facebook into your app or website

Display Name

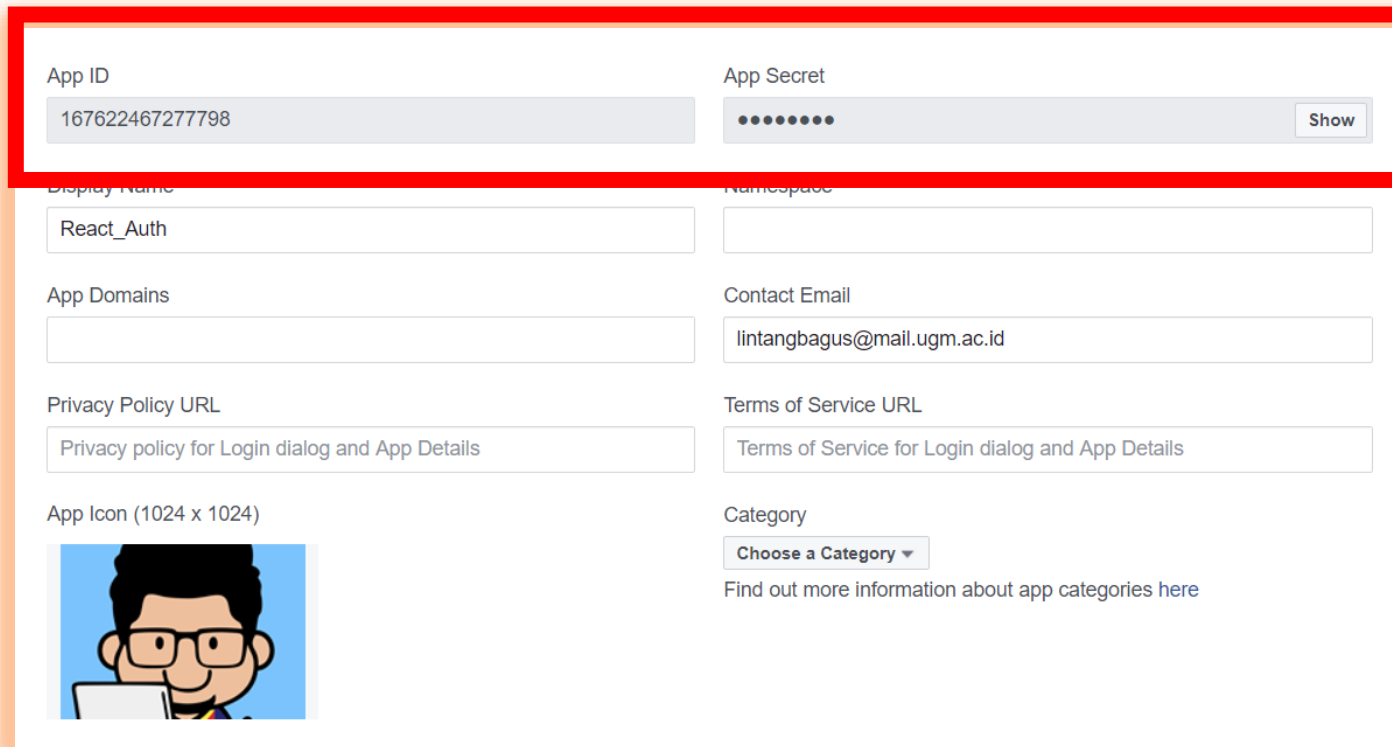
Contact Email

By proceeding, you agree to the Facebook Platform Policies


[Cancel](#) [Create App ID](#)

# #3 App ID & Secret

- On your Facebook dev dashboard, go to **Settings** menu → **Basic**, you will see your App ID & App Secret. Save its configuration!



The screenshot shows the Facebook App Settings page, specifically the 'Basic' tab. A red rectangular box highlights the 'App ID' and 'App Secret' fields at the top. The 'App ID' field contains the value '167622467277798'. The 'App Secret' field is masked with dots and has a 'Show' button to its right. Below these fields, the 'Display Name' is 'React\_Auth' and the 'Namespace' is empty. The 'App Domains' field is empty, and the 'Contact Email' is 'lintangbagus@mail.ugm.ac.id'. The 'Privacy Policy URL' and 'Terms of Service URL' fields both contain the text 'Privacy policy for Login dialog and App Details'. The 'App Icon (1024 x 1024)' field shows a cartoon character icon. The 'Category' field has a dropdown menu labeled 'Choose a Category' and a link to 'Find out more information about app categories here'.

App ID	App Secret
167622467277798	..... Show
Display Name	Namespace
React_Auth	
App Domains	Contact Email
	lintangbagus@mail.ugm.ac.id
Privacy Policy URL	Terms of Service URL
Privacy policy for Login dialog and App Details	Terms of Service for Login dialog and App Details
App Icon (1024 x 1024)	Category
	Choose a Category ▼
	<a href="#">Find out more information about app categories here</a>

# #4 Finishing Firebase Auth

- Go back to Firebase Auth, insert our App ID & App Secret on Facebook sign-in provider. Then copy ***OAuth redirect URI***. We'll need this on Facebook dev.

Facebook

Enable ☒

App ID

167622467277798

App secret

606db6f5685e2f773a29f4a640564d30

To complete set up, add this OAuth redirect URI to your Facebook app configuration. [Learn more](#)

[https://lintang-okay.firebaseio.com/\\_/auth/handler](https://lintang-okay.firebaseio.com/_/auth/handler)

Copy OAuth redirect URI

CANCEL SAVE



# #5 Facebook Login

- Go back to Facebook dev, **Add a Product**, choose **Facebook Login**. On its **Settings**, insert our **OAuth redirect URI** from Firebase. Save & you're done.

The screenshot shows the 'Client OAuth Settings' page in the Facebook Developer console. It contains several toggle switches for various OAuth features. A red rectangular box highlights the 'Valid OAuth Redirect URIs' section at the bottom. Inside this box, a text input field contains the URL 'https://lintang-okay.firebaseio.com/\_\_/auth/handler' with a small 'x' icon to its right. Below the input field, there is a 'Login from Devices' toggle switch which is currently set to 'No'.

Client OAuth Settings

☒ Yes ☐ No Client OAuth Login  
Enables the standard OAuth client token flow. Secure your application and prevent abuse by locking down which token redirect URIs are allowed with the options below. Disable globally if not used. [?]

☒ Yes ☐ No Web OAuth Login  
Enables web-based Client OAuth Login. [?]

☐ Yes ☒ No Enforce HTTPS  
Enforce the use of HTTPS for Redirect URIs and the JavaScript SDK. Strongly recommended. [?]

☐ Yes ☒ No Force Web OAuth Reauthentication  
When on, prompts people to enter their Facebook password in order to log in on the web. [?]

☐ Yes ☒ No Embedded Browser OAuth Login  
Enable webview Redirect URIs for Client OAuth Login. [?]

☒ Yes ☐ No Use Strict Mode for Redirect URIs  
Only allow redirects that use the Facebook SDK or that exactly match the Valid OAuth Redirect URIs. Strongly recommended. [?]

Valid OAuth Redirect URIs

☐ Yes ☒ No Login from Devices  
Enables the OAuth client login flow for devices like a smart TV [?]

## #6 Firebase Snippet

- Go to **Firestore Project Overview**, choose **Add Firebase to Your Web App** then copy its snippet code.

### Add Firebase to your web app



Copy and paste the snippet below at the bottom of your HTML, before other script tags.

```
<script src="https://www.gstatic.com/firebasejs/4.11.0/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyAtQf_vMPce7JTfYQesg2isg8pZftJwIII",
    authDomain: "linyoa-37d9d.firebaseio.com",
    databaseURL: "https://linyoa-37d9d.firebaseio.com",
    projectId: "linyoa-37d9d",
    storageBucket: "linyoa-37d9d.appspot.com",
    messagingSenderId: "922933748183"
  };
  firebase.initializeApp(config);
</script>
```

COPY

# #7 Create/Prepare Your React App

- Install create-react-app utility globally on your PC (if you have not done yet before)

```
$ npm install -g create-react-app
```

- Create a new react app or prepare yours

```
$ create-react-app lin_auth  
$ cd lin_auth
```

- Install Firebase packages needed

```
$ npm install -g firebase-tools  
$ npm install firebase
```

```
import firebase from 'firebase';

var config = {
  apiKey: "a1b2c3d4e5f6g7h8i9j0",
  authDomain: "lintang-auth.firebaseio.com",
  databaseURL: "https://lintang-auth.firebaseio.com",
  projectId: "lintang-auth",
  storageBucket: "lintang-auth.appspot.com",
  messagingSenderId: "1234567890"
};

firebase.initializeApp(config);
export const ref = firebase.database().ref();
export const auth = firebase.auth;
export const provider = new
firebase.auth.FacebookAuthProvider();
```

```
import React, { Component } from 'react';
import { provider, auth } from './fire';

class App extends Component {
  constructor(){
    super();
    this.state = {user: null}
  }
  async login() {
    const result = await auth().signInWithPopup(provider);
    this.setState({user: result.user});
  }
  async logout(){
    await auth().signOut()
    this.setState({user: null});
  }
}
```

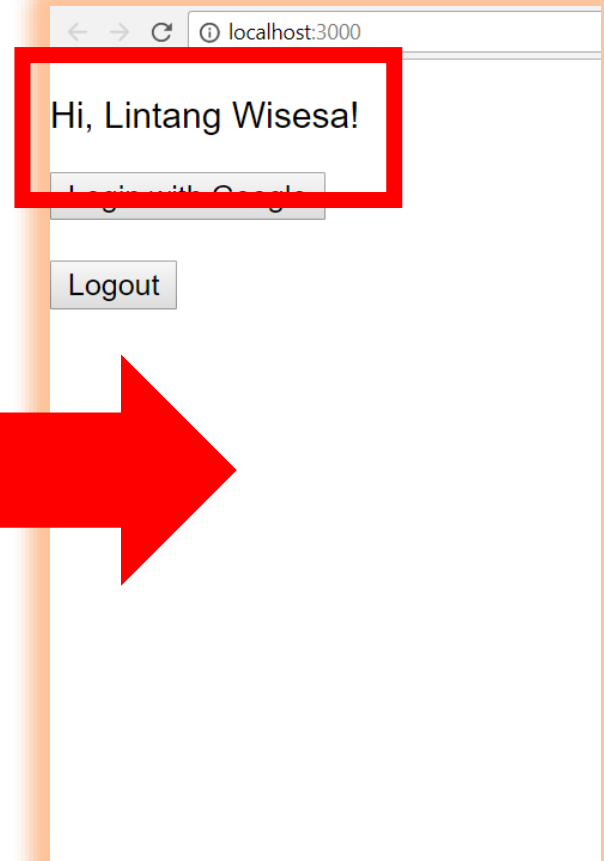
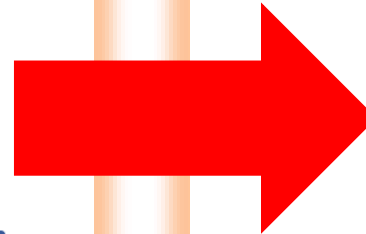
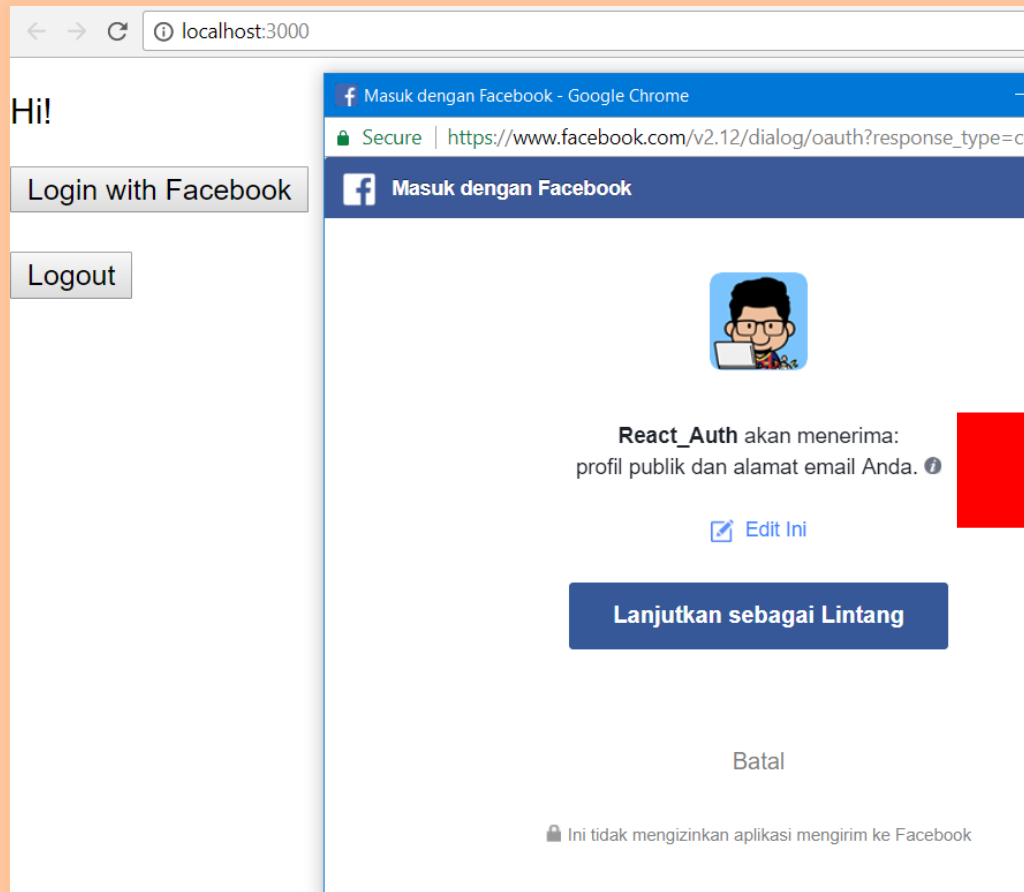
• • • • •

• • • • •

```
render() {  
  return (  
    <div>  
      <p>  
        {this.state.user ? `Hi,  
        ${this.state.user.displayName}!` : 'Hi!'}  
      </p>  
      <button onClick={this.login.bind(this)}>  
        Login with Facebook  
      </button>  
      <button onClick={this.logout.bind(this)}>  
        Logout  
      </button>  
    </div>  
  );  
}
```

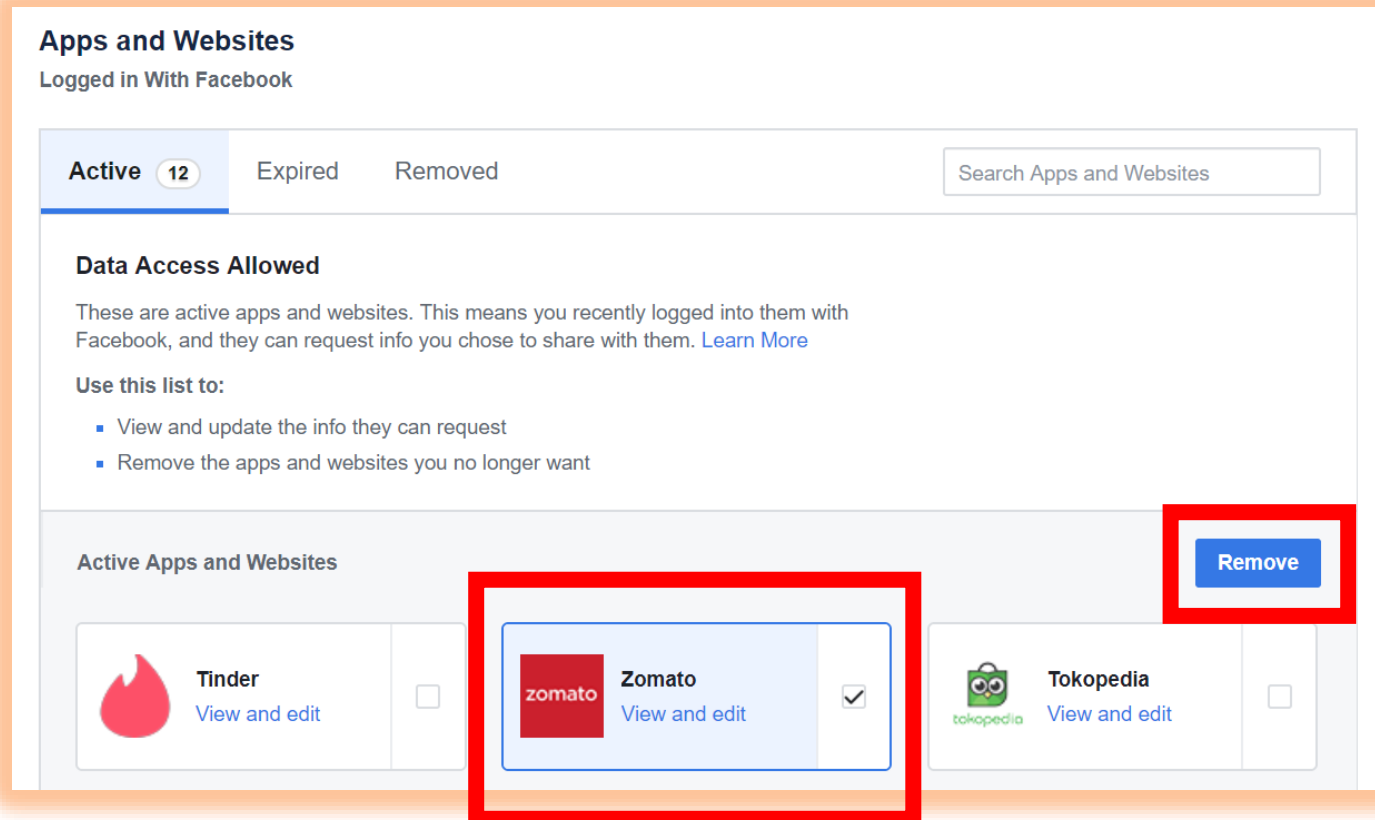
export default App;

# React, Firebase Auth With Facebook Provider



# Revoke Facebook Permission

- Go to Facebook.com, on **Settings** → **Apps & Websites**, tick on our app then simply **remove** it.



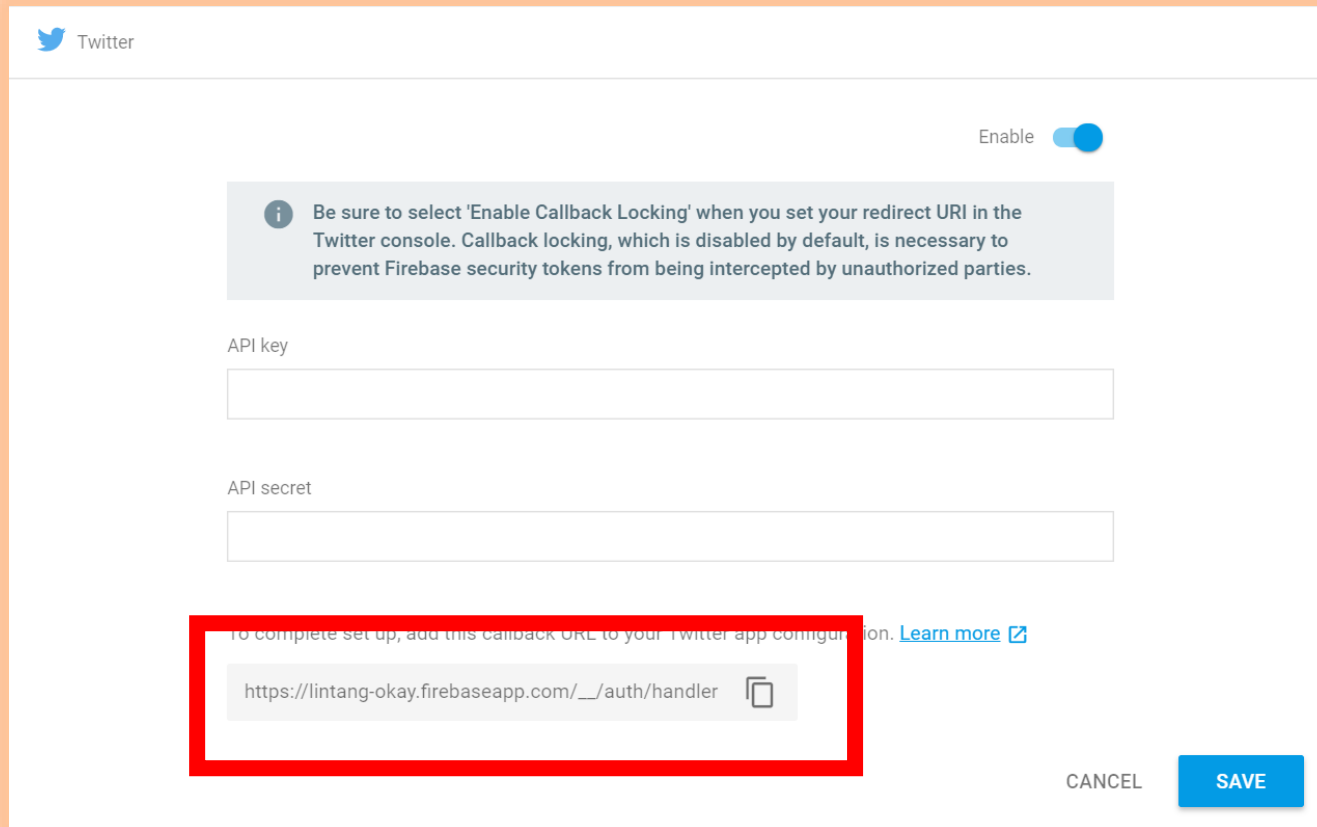




# How to Setup React and Firebase Authentication with Twitter Provider?

# #1 Setup Firebase Auth

- Login to **Firestore** Console, choose/create a project then go to **Firestore Authentication**. On **Sign-In Method** tab, enable **Twitter** sign-in provider & copy its **Callback URL**!



The screenshot shows the 'Twitter' sign-in provider configuration page in the Firebase console. At the top left is the Twitter logo and name. On the right, there is an 'Enable' toggle switch that is turned on. Below this, a grey information box contains a note: 'Be sure to select 'Enable Callback Locking' when you set your redirect URI in the Twitter console. Callback locking, which is disabled by default, is necessary to prevent Firebase security tokens from being intercepted by unauthorized parties.' Underneath the info box are two text input fields labeled 'API key' and 'API secret'. At the bottom, a red rectangle highlights a text box containing the callback URL 'https://lintang-okay.firebaseio.com/\_/auth/handler' with a copy icon to its right. Above this URL, a line of text says 'To complete set up, add this callback URL to your Twitter app configuration. [Learn more](#)'. At the bottom right of the form are 'CANCEL' and 'SAVE' buttons.

Twitter

Enable ☒

*i* Be sure to select 'Enable Callback Locking' when you set your redirect URI in the Twitter console. Callback locking, which is disabled by default, is necessary to prevent Firebase security tokens from being intercepted by unauthorized parties.

API key

API secret

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

[https://lintang-okay.firebaseio.com/\\_/auth/handler](https://lintang-okay.firebaseio.com/_/auth/handler)

CANCEL SAVE

# #2 Twitter Dev

- Login to **Twitter**, go to <https://dev.twitter.com/apps/new> or <https://apps.twitter.com/> and **Create New App**. Fill in the forms, insert our **Callback URL** then click **Create Your Twitter Application**.

### Application Details

**Name \***

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

**Description \***

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

**Website \***

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.  
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

**Callback URL**

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth\_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

### Developer Agreement

☒ Yes, I have read and agree to the [Twitter Developer Agreement](#).

# #3 API Key & Secret

- Go to **Keys and Access Tokens** to get our API Key & API Secret. Copy these code to our Firebase Auth & Save!

## Lintang\_React\_Auth

[Details](#)[Settings](#)[Keys and Access Tokens](#)[Permissions](#)

### Application Settings

*Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.*

Consumer Key (API Key)

PzZItOV0XF100jmeePKLzZm0h

Consumer Secret (API Secret)

kCkuHRsMSeQoCaGrnjPmFtfY8Mbf8brOnfnMQt0HoY7eldE1WJ

Access Level

Read and write ([modify app permissions](#))

Owner

Lintang\_Wisesa

Owner ID

2445138667

## #4 Firebase Snippet

- Go to ***Firebase Project Overview***, choose ***Add Firebase to Your Web App*** then copy its snippet code.

### Add Firebase to your web app

Copy and paste the snippet below at the bottom of your HTML, before other script tags.

```
<script src="https://www.gstatic.com/firebasejs/4.11.0/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyAtQf_vMPce7JTfYQesg2isg8pZftJwIII",
    authDomain: "linyoa-37d9d.firebaseio.com",
    databaseURL: "https://linyoa-37d9d.firebaseio.com",
    projectId: "linyoa-37d9d",
    storageBucket: "linyoa-37d9d.appspot.com",
    messagingSenderId: "922933748183"
  };
  firebase.initializeApp(config);
</script>
```

COPY

# #5 Create/Prepare Your React App

- Install create-react-app utility globally on your PC (if you have not done yet before)

```
$ npm install -g create-react-app
```

- Create a new react app or prepare yours

```
$ create-react-app lin_auth
```

```
$ cd lin_auth
```

- Install Firebase packages needed

```
$ npm install -g firebase-tools
```

```
$ npm install firebase
```

```
import firebase from 'firebase';

var config = {
  apiKey: "a1b2c3d4e5f6g7h8i9j0",
  authDomain: "lintang-auth.firebaseio.com",
  databaseURL: "https://lintang-auth.firebaseio.com",
  projectId: "lintang-auth",
  storageBucket: "lintang-auth.appspot.com",
  messagingSenderId: "1234567890"
};

firebase.initializeApp(config);
export const ref = firebase.database().ref();
export const auth = firebase.auth;
export const provider = new
firebase.auth.TwitterAuthProvider();
```

```
import React, { Component } from 'react';
import { provider, auth } from './fire';

class App extends Component {
  constructor(){
    super();
    this.state = {user: null}
  }
  async login() {
    const result = await auth().signInWithPopup(provider);
    this.setState({user: result.user});
  }
  async logout(){
    await auth().signOut()
    this.setState({user: null});
  }
}
```

• • • • •

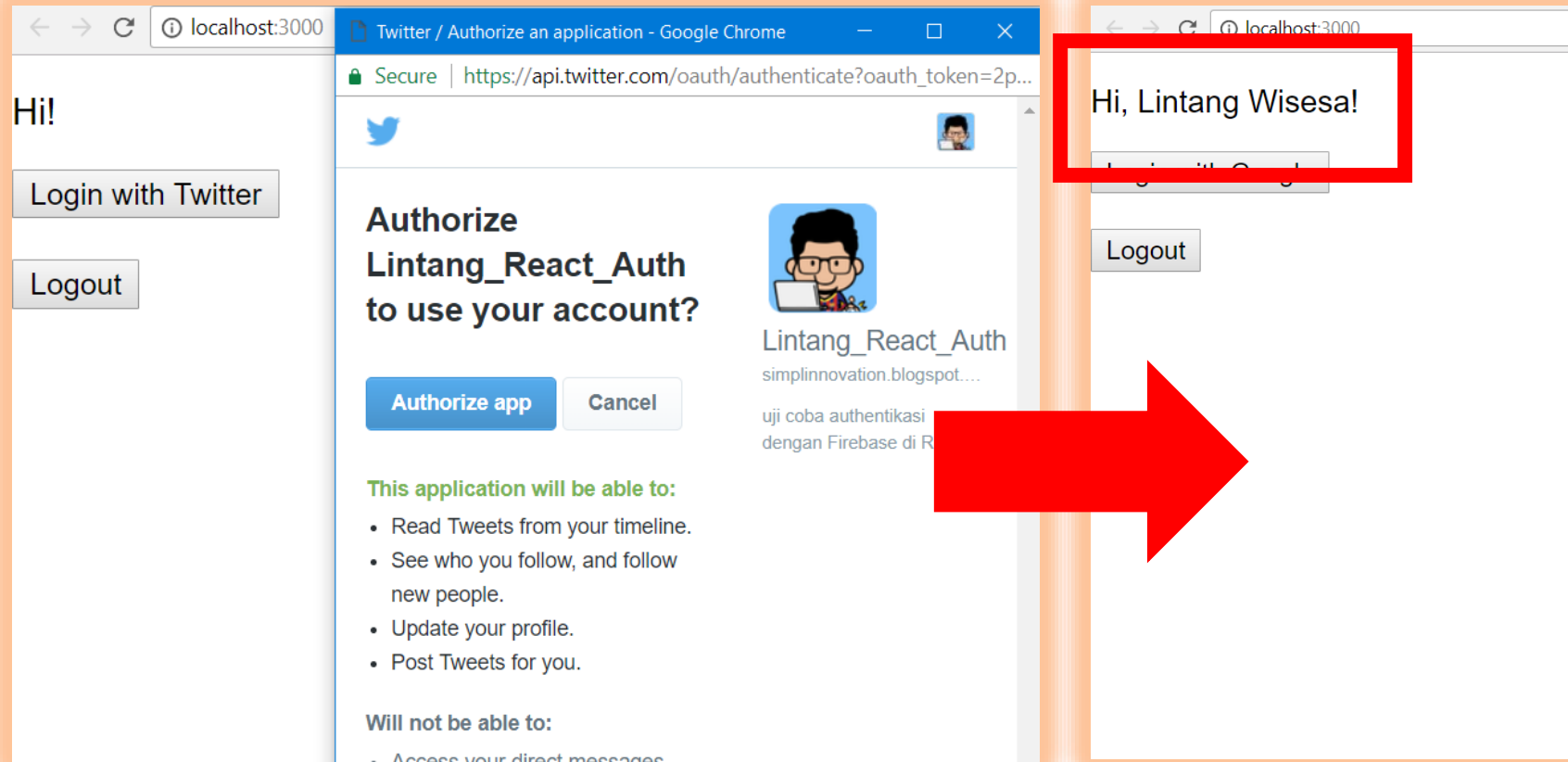


• • • • •

```
render() {  
  return (  
    <div>  
      <p>  
        {this.state.user ? `Hi,  
        ${this.state.user.displayName}!` : 'Hi!'}  
      </p>  
      <button onClick={this.login.bind(this)}>  
        Login with Twitter  
      </button>  
      <button onClick={this.logout.bind(this)}>  
        Logout  
      </button>  
    </div>  
  );  
}
```

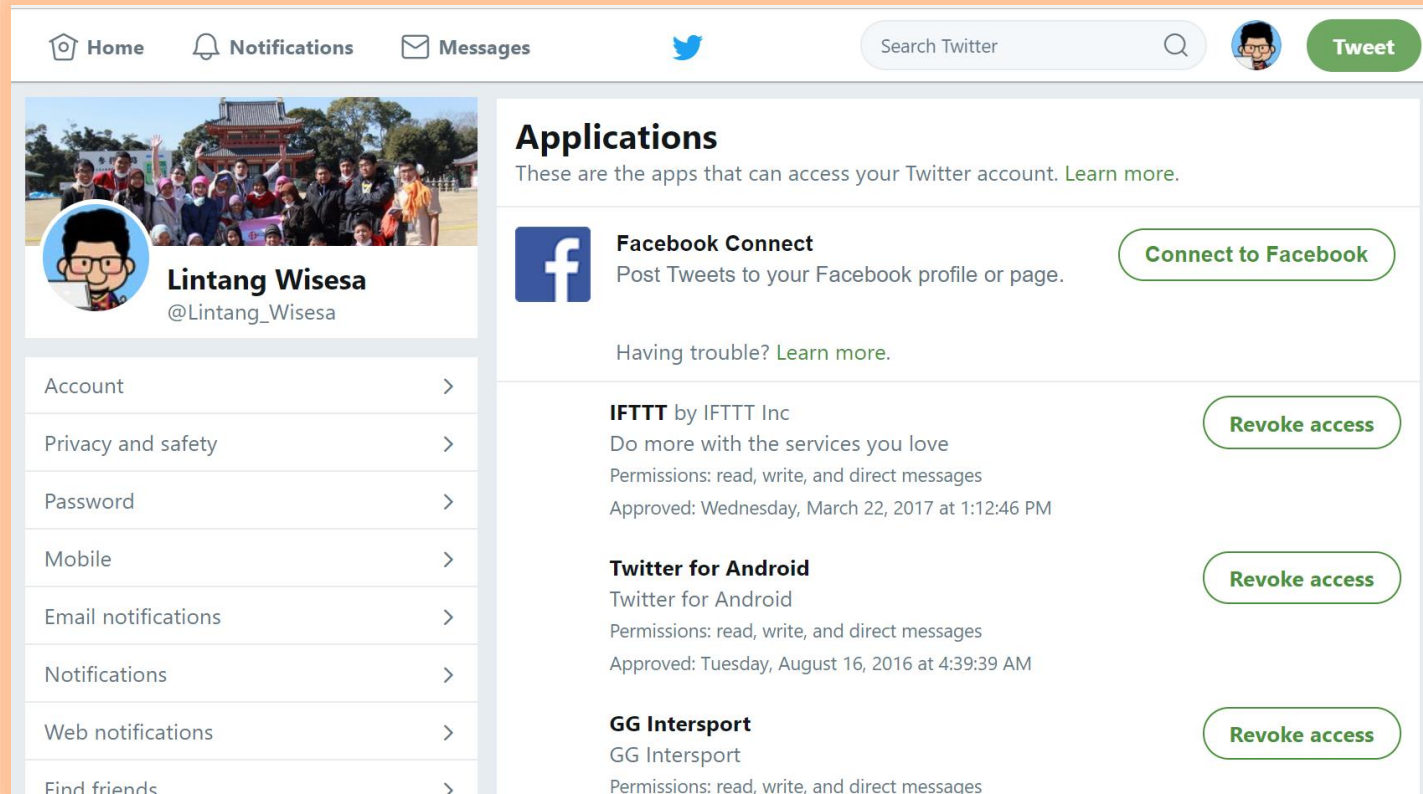
export default App;

# React, Firebase Auth With Twitter Provider



# Revoke Twitter Permission

- Go to Twitter.com, on **Settings** → **Apps** simply remove its access on an application.



Front-End Development



# React & Firebase

**#3** Firebase Auth with Various Providers