

Lab 9

Authentication with AWS

(Amplify + Cognito)

Software Studio
Datalab, CS, NTHU
2023

Gitlab

lab-weathermood-auth

The screenshot displays the GitLab web interface for the project 'lab-weathermood-auth'. The left sidebar contains navigation options: Project information, Repository, Issues (0), Merge requests (0), Security & Compliance, Deployments, Monitor, Infrastructure, Packages & Registries, Analytics, Wiki, Snippets, and Settings. The main content area shows a file list with columns for file name, commit type, and time since last commit:

File Name	Commit Type	Time
src	initial commit	27 minutes ago
.gitignore	initial commit	27 minutes ago
package-lock.json	initial commit	27 minutes ago
package.json	initial commit	27 minutes ago
readme.md	update readme	24 minutes ago
webpack.config.js	initial commit	27 minutes ago

Below the file list, the 'readme.md' file is selected, showing the following content:

Assignment - Weathermood Auth

In this assignment, you are asked to add basic authentication function to Weathermood, using AWS Amplify. Only client side code is provided, this can run on localhost, no need to deploy on AWS.

Requirement

1. (30%) Sign up / sign in with username and password.
2. (30%) Sign in with google account.

The README includes a screenshot of a 'Sign in to your account' form. The form features a 'Sign in with Google' button, a 'Username' field with the placeholder 'Enter your username', and a 'Password' field with the placeholder 'Enter your password'. There is a 'Forgot your password? Reset password' link and a 'SIGN IN' button at the bottom right.

Outline

1. AWS Amplify & Cognito
2. Lab-weathermood-auth
3. Some helpful docs / tutorials

AWS Amplify

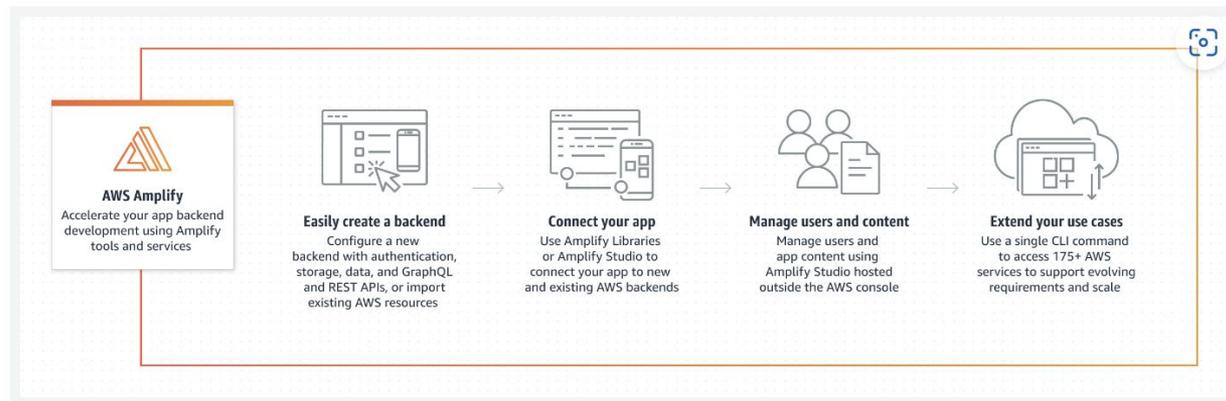
AWS Amplify is a complete solution that lets frontend web and mobile developers easily build, ship, and host full-stack applications on AWS, with the flexibility to leverage the breadth of AWS services as use cases evolve. No cloud expertise needed.

Create an app backend

Build a frontend UI

Host a web app

Create a cross-platform backend for your iOS, Android, Flutter, web, or React Native app with real-time and offline functionality in just a few clicks.



AWS Amplify

Build

Use the [Amplify Studio](#) or [Amplify CLI](#) to configure an app backend, and use the Amplify [libraries](#) and [UI components](#) to connect your app to your backend.

Authentication

Create seamless on-boarding flows with a fully-managed user directory and pre-built sign-up, sign-in, forgot password, and multi-factor auth workflows. Amplify also supports login with a social provider such as Facebook, Google Sign-In, or Login With Amazon and provides fine grained access control to mobile and web applications. Powered by Amazon Cognito.

Push notifications

Improve customer engagement by using marketing and analytics capabilities. Leverage customer insights to segment and target your customers more effectively. You can tailor your content and communicate through multiple channels including email, texts as well as push notifications. Powered by Amazon Pinpoint.

Analytics

Understand the behavior of your web, iOS or Android users. Use auto tracking to track user sessions and web page metrics or create custom user attributes and in-app metrics. Get access to real time data stream and analyze the data for customer insights and build data driven marketing strategies to drive customer adoption, engagement, and retention. Powered by Amazon Pinpoint and Amazon Kinesis.

API

Make secure HTTP requests to [GraphQL](#) and [REST](#) endpoints to access, manipulate, and combine data from one or more data sources such Amazon DynamoDB, Amazon Aurora Serverless, and your custom data sources with AWS Lambda. Amplify enables you to easily build scalable applications that require real-time updates, local data access for offline scenarios, and data synchronization with customizable conflict resolution when devices are back online. Powered by AWS AppSync and Amazon API Gateway.

We will be using the auth service from Amplify, which is powered by Cognito

AWS Cognito

- Cognito provides sign-up and authentication functions to your mobile and web apps
- Also enables authentication through external identity providers, and provides temporary credentials to access your app's backend resources in AWS

The screenshot displays the AWS Cognito console interface. At the top, it shows the navigation path 'Amazon Cognito > User pools'. Below this, there is a section for 'User pools (1) Info' with a description: 'View and configure your user pools. User pools are directories of federated and local user profiles. They provide authentication options for your users.' A search bar is provided with the placeholder text 'Search user pools by name or ID'. Below the search bar is a table with the following columns: 'User pool name', 'User pool ID', and 'Created time'. The table contains one entry: 'labweathermoodauthma0e32e1ed_userpool_Oe32e1ed-dev', 'us-west-2_Cq4SI05zB', and 'Yesterday'. Below the user pools section, there is a navigation bar with tabs: 'Users', 'Groups', 'Sign-in experience', 'Sign-up experience', 'Messaging', 'App integration', and 'User pool properties'. The 'Users' tab is selected. Below the navigation bar, there is a section for 'Users (1) Info' with a description: 'View, edit, and create users in your user pool. Users that are enabled and confirmed can sign in to your user pool.' A search bar is provided with the placeholder text 'Search users by attribute'. Below the search bar is a table with the following columns: 'User name', 'Email address', 'Email verified', and 'Confirmation status'. The table contains one entry: 'lab8weathermood', '.edu.tw', 'Yes', and 'Confirmed'.

Amazon Cognito > User pools

User pools (1) Info

View and configure your user pools. User pools are directories of federated and local user profiles. They provide authentication options for your users.

Search user pools by name or ID

User pool name	User pool ID	Created time
labweathermoodauthma0e32e1ed_userpool_Oe32e1ed-dev	us-west-2_Cq4SI05zB	Yesterday

Users | Groups | Sign-in experience | Sign-up experience | Messaging | App integration | User pool properties

Users (1) Info

View, edit, and create users in your user pool. Users that are enabled and confirmed can sign in to your user pool.

Property: User name Search users by attribute

User name	Email address	Email verified	Confirmation status
lab8weathermood	.edu.tw	Yes	Confirmed

Free tier policy

AWS Amplify Overview Tools ▾ Features ▾ **Pricing** Getting Started

Create a backend

Get started free, then pay as you go

Quickly create a cross-platform AWS backend for your iOS, Android, Flutter, web, or React Native apps using Amplify CLI or Amplify Studio. Connect your apps to a new or existing AWS backend with Amplify Libraries.

AMPLIFY STUDIO, AMPLIFY CLI, AMPLIFY LIBRARIES

No cost

AWS BACKEND RESOURCES

Pay as you go, starting with free offers*

AWS backend services deployed by Amplify CLI or Amplify Studio are offered on the AWS free tier and pay as you go: [API Gateway](#), [AppSync](#), [CloudFront](#), [Cognito](#), [DynamoDB](#), [Elasticsearch](#), [Kinesis](#), [Lambda](#), [Lex](#), [Location Service](#), [Pinpoint](#), [Rekognition](#), and [S3](#). * Kinesis is pay as you go only. Terms and conditions apply in connecting any app to AWS services and products.*

Amazon Cognito Overview Features **Pricing** Getting Started Resources

Cognito User Pools

If you are using Cognito Identity to create a User Pool, you pay based on your monthly active users (MAUs) only. A user is counted a operation related to that user, such as sign-up, sign-in, token refresh, password change, or a user account attribute is updated. You users within that calendar month.

There is separate pricing for users who sign in directly with their credentials from a User Pool and for users who sign in through an

Free Tier

The Cognito Your User Pool feature has a free tier of 50,000 MAUs per account for users who sign in directly to Cognito User Pools based identity providers. The free tier does not automatically expire at the end of your 12 month AWS Free Tier term, and it is avail indefinitely. **Please note - the free tier pricing isn't available for both Your User Pool feature and SAML or OIDC federation in th**

Amazon Cognito Overview Features **Pricing** Getting Started Resources

SMS messages for Multi-Factor Authentication

Separate pricing applies for sending SMS messages for Multi-Factor Authentication (MFA) and phone number verification. Amazon to send SMS messages, and you can reference [Amazon SNS pricing](#).

Use of the Federated Identities feature for authenticating users and generating unique identifiers is provided at no- charge.

Lab-weathermood-auth

In today's lab, we will show you how to

1. Install Amplify CLI
2. Init authentication service
3. Add packages to your project
4. Add authentication to your project
5. Add sign in with Google
6. Get user data
7. Create greeting components and customize the components

Install Amplify

1. `npm install -g @aws-amplify/cli`
 - troubleshooting: Error: EACCES: permission denied see [\[npm Docs\]](#)
2. `amplify configure`
 - An AWS browser page will popup and ask you to create an IAM user
 - Give the user the “AdministratorAccess” permission

```
● (base) sasaya@sasayadeMacBook-Pro lab-weathermood-auth-master % amplify configure
Follow these steps to set up access to your AWS account:

Sign in to your AWS administrator account:
https://console.aws.amazon.com/
Press Enter to continue

Specify the AWS Region
? region: us-west-2
Follow the instructions at
https://docs.amplify.aws/cli/start/install/#configure-the-amplify-cli

to complete the user creation in the AWS console
https://console.aws.amazon.com/iamv2/home#/users/create
Press Enter to continue

Enter the access key of the newly created user:
? accessKeyId: *****
? secretAccessKey: *****
This would update/create the AWS Profile in your local machine
? Profile Name: default

Successfully set up the new user.
```

Install Amplify

3. amplify init

```
• (base) sasaya@sasayadeMacBook-Pro lab-weathermood-auth-master % amplify init
Note: It is recommended to run this command from the root of your app directory
? Enter a name for the project labweathermoodauthma
The following configuration will be applied:

Project information
| Name: labweathermoodauthma
| Environment: dev
| Default editor: Visual Studio Code
| App type: javascript
| Javascript framework: react
| Source Directory Path: src
| Distribution Directory Path: build
| Build Command: npm run-script build
| Start Command: npm run-script start

? Initialize the project with the above configuration? No
? Enter a name for the environment dev
? Choose your default editor: Visual Studio Code
✓ Choose the type of app that you're building · javascript
Please tell us about your project
? What javascript framework are you using react
? Source Directory Path: src
? Distribution Directory Path: dist
? Build Command: npm run-script build
? Start Command: npm run-script start
Using default provider awscloudformation
? Select the authentication method you want to use: AWS profile

For more information on AWS Profiles, see:
https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-profiles.html

? Please choose the profile you want to use default
Adding backend environment dev to AWS Amplify app: d3pvw3tqv8zok3

Deployment completed.
Deploying root stack labweathermoodauthma [ ===== ]
amplify-labweathermoodauthma-... AWS::CloudFormation::Stack CREATE_IN_PROGRES
AuthRole AWS::IAM::Role CREATE_COMPLETE
DeploymentBucket AWS::S3::Bucket CREATE_IN_PROGRES
UnauthRole AWS::IAM::Role CREATE_COMPLETE

✓ Help improve Amplify CLI by sharing non sensitive configurations on failures (y/N) ·
Deployment state saved successfully.
✓ Initialized provider successfully.
✓ Initialized your environment successfully.

Your project has been successfully initialized and connected to the cloud!
```

Install Amplify

More amplify commands

Some next steps:

```
"amplify status" will show you what you've added already and if it's locally configured or deployed  
"amplify add <category>" will allow you to add features like user login or a backend API  
"amplify push" will build all your local backend resources and provision it in the cloud  
"amplify console" to open the Amplify Console and view your project status  
"amplify publish" will build all your local backend and frontend resources (if you have hosting categ
```

Pro tip:

```
Try "amplify add api" to create a backend API and then "amplify push" to deploy everything
```

Note: for Windows users running amplify cli, there will be many option-choosing operation. If you are experiencing difficulty operating it, don't use git bash, use the build-in cmd or power shell.

Add authentication service

1. `amplify add auth`
2. `amplify push`
 - Push local changes to the cloud
 - This will generate an export file, which will be used later

```
● (base) sasaya@sasayadeMacBook-Pro lab-weathermood-auth-master % amplify add auth
Using service: Cognito, provided by: awscloudformation

The current configured provider is Amazon Cognito.

Do you want to use the default authentication and security configuration? Default configuration
Warning: you will not be able to edit these selections.
How do you want users to be able to sign in? Username
Do you want to configure advanced settings? No, I am done.
✔ Successfully added auth resource labweathermoodauthma0e32e1ed locally
```

Lab-weathermood-auth

In today's lab, we will show you how to

1. Install Amplify CLI
2. Init authentication service
3. Add packages to your project
4. Add authentication to your project
5. Add sign in with Google
6. Get user data
7. Create greeting components and customize the components

Add packages to your project

1. Install package for development `npm install --save aws-amplify @aws-amplify/ui-react`
2. Import packages and call Amplify in index.jsx

```
import Amplify from 'aws-amplify';  
import config from './aws-exports';  
  
Amplify.configure(config);
```

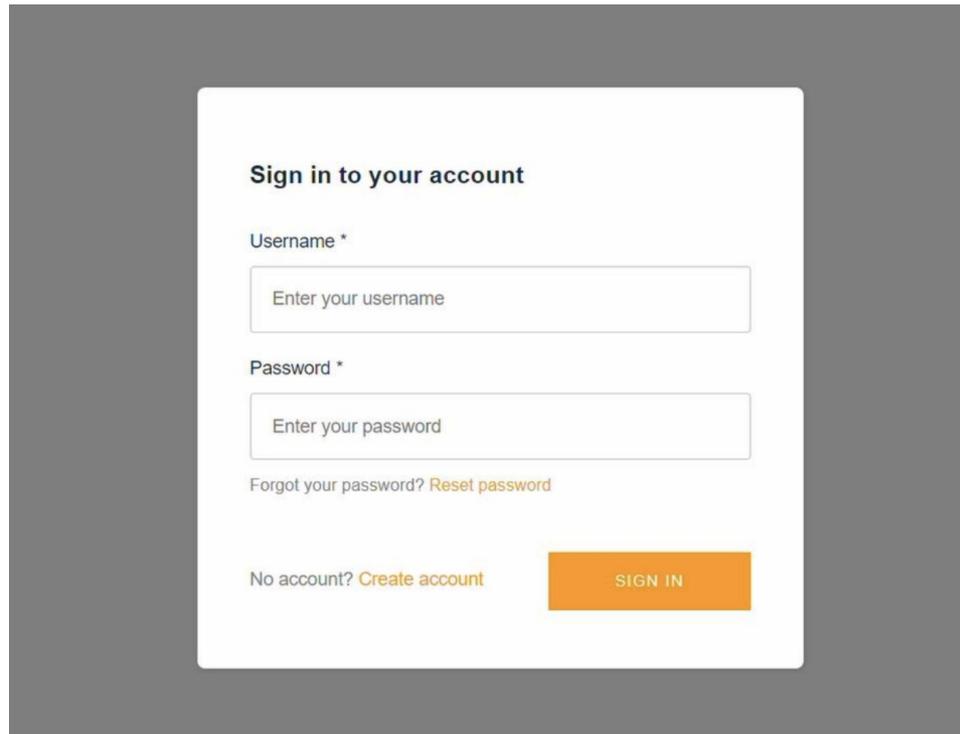
Add authentication in your project

1. In main.jsx, wrap your main component with HOC function
2. `amplify push`
3. `npm run start`
4. Then you could see a login screen in your website

```
import {withAuthenticator} from "@aws-amplify/ui-react";
```

```
export default withAuthenticator(connect(state => ({  
  ...state.main,  
  searchText: state.searchText,  
}))(Main));
```

Add authentication in your project



Sign in to your account

Username *

Password *

Forgot your password? [Reset password](#)

No account? [Create account](#)

Lab-weathermood-auth

In today's lab, we will show you how to

1. Install Amplify CLI
2. Init authentication service
3. Add packages to your project
4. Add authentication to your project
5. **Add sign in with Google**
6. Get user data
7. Create greeting components and customize the components

Add sign in with Google

Create Google client id for Google sign in

1. Google Cloud Console > APIs & Services > Create project
2. APIs & Services > Credentials > Create credentials > OAuth client ID
 - Configure consent screen
 - Configure your url (in this lab, <http://localhost:8080>)
 - Save your client ID and client secret

APIs & Services

Enabled APIs & services

Library

Credentials

OAuth consent screen

Page usage agreements

Credentials

+ CREATE CREDENTIALS DELETED RESTORE DELETED CREDENTIALS

Create credentials to access your project's APIs

API Keys

No API keys to display

OAuth 2.0 Client IDs

Name	Creation date	Type
Web client 1 lab8	May 17, 2023	Web application

API key
Identifies your project using a simple API key to check quota and access

OAuth client ID
Requests user consent so your app can access the user's data

Service account
Enables server-to-server, app-level authentication using robot accounts

Help me choose
Asks a few questions to help you decide which type of credential to use

Add sign in with Google

← Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information. [Learn more](#) about OAuth client types.

Application type *
Web application

Name *
Web client 2

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.



The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

Authorized JavaScript origins

For use with requests from a browser

URIs 1 *
http://localhost:8080

+ ADD URI

Authorized redirect URIs

For use with requests from a web server

URIs 1 *
http://localhost:8080



Add login with Google

```
amplify update auth, then amplify push
```

```
● (base) sasaya@sasayadeMacBook-Pro lab-weathermood-auth-master % amplify update auth
Please note that certain attributes may not be overwritten if you choose to use defaults settings.
Using service: Cognito, provided by: awscloudformation
What do you want to do? Walkthrough all the auth configurations
Select the authentication/authorization services that you want to use: User Sign-Up, Sign-In, connected with AWS IAM controls (Enables per-user Storage features for
Allow unauthenticated logins? (Provides scoped down permissions that you can control via AWS IAM) No
Do you want to enable 3rd party authentication providers in your identity pool? Yes
Select the third party identity providers you want to configure for your identity pool: Google

You've opted to allow users to authenticate via Google. If you haven't already, you'll need to go to https://developers.google.com/identity and create an App ID.

Enter your Google Web Client ID for your identity pool: 543569104988-aticenfgnbdsjvdpag9 5gq.apps.googleusercontent.com
Do you want to add User Pool Groups? No
Do you want to add an admin queries API? No
Multifactor authentication (MFA) user login options: OFF
Email based user registration/forgot password: Enabled (Requires per-user email entry at registration)
Specify an email verification subject: Your verification code
Specify an email verification message: Your verification code is {####}
Do you want to override the default password policy for this User Pool? No
Specify the app's refresh token expiration period (in days): 30
Do you want to specify the user attributes this app can read and write? No
Do you want to enable any of the following capabilities?
Do you want to use an OAuth flow? No
? Do you want to configure Lambda Triggers for Cognito? No
✔ Successfully updated auth resource labweathermoodauthma0e32e1ed locally

✔ Some next steps:
"amplify push" will build all your local backend resources and provision it in the cloud
"amplify publish" will build all your local backend and frontend resources (if you have hosting category added) and provision it in the cloud

✔ Successfully updated resource update locally
```

Add login with Google

In index.jsx, pass your Google client ID to <main>

```
const federated = {
  googleClientId: "543569104988-aticenf      pag93a1s1ut41l5gq.apps.googleusercontent.com"
}

window.onload = function() {
  const composeEnhancers = window.__REDUX_DEVTOOLS_EXTENSION_COMPOSE__ || compose;
  const store = createStore(combineReducers({
    unit, weather, weatherForm, forecast,
    searchText, post, postForm, postItem,
    todoForm, todo,
    main,
  }), composeEnhancers(applyMiddleware(thunkMiddleware/*, loggerMiddleware*/)));

  ReactDOM.render(
    <Provider store={store}>
      <Main federated = {federated}/>
    </Provider>,
    document.getElementById('root')
  );
};
```

Add login with Google

The image shows a web browser window with the following elements:

- Browser tab: WeatherMood
- Address bar: localhost:8080
- Page title: Sign in to your account
- Primary action: A blue button with the Google logo and the text "Sign in with Google".
- Separator: A horizontal line with the word "or" in the center.
- Form fields:
 - Username *: A text input field with the placeholder "Enter your username".
 - Password *: A text input field with the placeholder "Enter your password".
- Links:
 - Forgot your password? [Reset password](#)
 - No account? [Create account](#)
- Final action: An orange button labeled "SIGN IN".

Lab-weathermood-auth

In today's lab, we will show you how to

1. Install Amplify CLI
2. Init authentication service
3. Add packages to your project
4. Add authentication to your project
5. Add sign in with Google
6. Get user data
7. Create greeting components and customize the components

Get user data in your component

You can use Auth to get user data

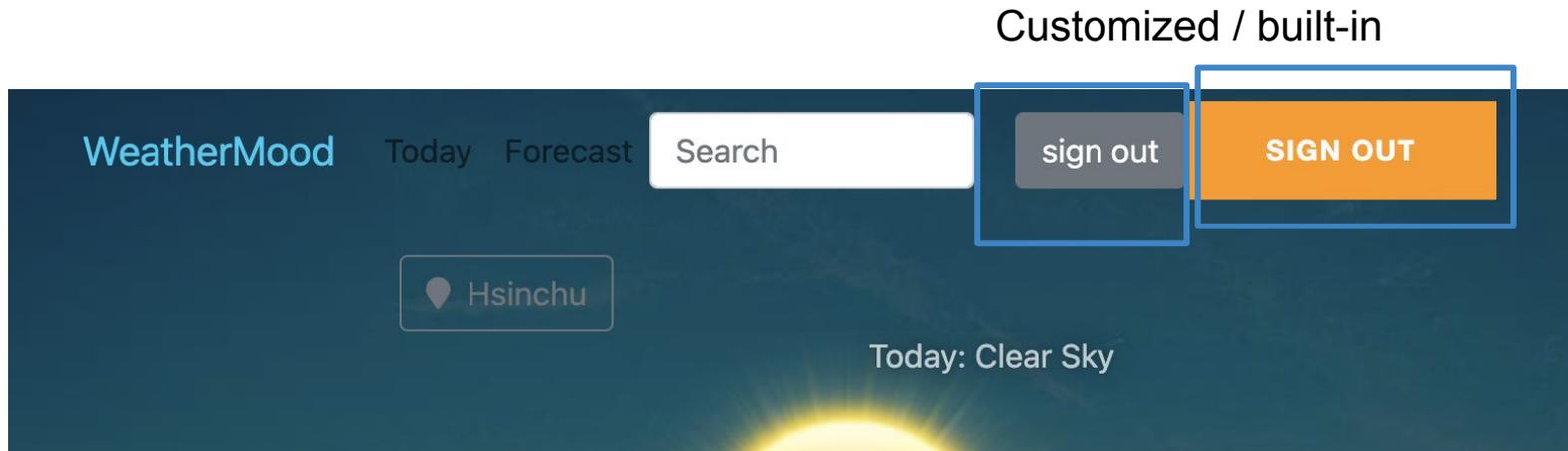
```
Auth.currentAuthenticatedUser().then(user => {console.log(user)})
```

The screenshot shows a web browser displaying a weather application named "WeatherMood". The application shows the current weather for Hsinchu as "Today: Clear Sky" with a temperature of "27°C". Below the weather information, there is a "Posts" section with a "Mood" dropdown and a text input field containing "What's on your mind?". A "Post" button is visible next to the input field. The browser's developer console is open on the right side, showing a log of network requests and a console error. The console error is a "TypeError: Cannot read properties of undefined (reading 'id')", which occurred in the "PostList.jsx" file at line 84. The console also shows a "CognitoUser" object with various attributes, including "attributes", "client", "pool", "signInUserSession", and "storage".

Create customized component

You can use built-in component, for example the sign out button `import {AmplifySignOut} from "@aws-amplify/ui-react"` then use `<AmplifySignOut/>` directly.

Or customize it, e.g., `<Button onClick = {() => Auth.signOut().then(() => window.location.reload())}>sign out</Button>`



There are more built-in components, e.g., `{AmplifyGreeting}`, which are very handy

More tutorials

Amplify Document

- Authentication: getting started
<https://docs.amplify.aws/lib/auth/getting-started/q/platform/js/>
- Tutorial: add authentication
<https://docs.amplify.aws/start/getting-started/auth/q/integration/react/>
- Authenticator
<https://ui.docs.amplify.aws/react/connected-components/authenticator>
- AuthClass methods
<https://aws-amplify.github.io/amplify-js/api/classes/authclass.html>

Some tutorials that might help (api version may be different)

- <https://richardzcode.github.io/Journal-AWS-Amplify-Tutorial/step-02/>
- <https://blog.kylegalbraith.com/2020/03/31/customizing-the-aws-amplify-authentication-ui-with-your-own-react-components/>