Cruise0 Demo - Customisation Notes

Alistair Ferrier

November 2021

Contents

[Demo Installation 3](#_Toc87718141)

[Prerequisites: 3](#_Toc87718142)

[Application Setup 4](#_Toc87718143)

[Classic Universal Login Config 8](#_Toc87718144)

[Enable Direct Signup from page: 10](#_Toc87718145)

[New Universal Login Customisation 11](#_Toc87718146)

[Auth0 CLI Usage for Customisation 11](#_Toc87718147)

[Email Verification Warning 13](#_Toc87718148)

[Social Media Authentication 14](#_Toc87718149)

[Account Linking 15](#_Toc87718150)

# Demo Installation

### Prerequisites:

Node.js <https://nodejs.org/en/>

Node Version Manager (NVM) <https://github.com/coreybutler/nvm-windows/releases/download/1.1.8/nvm-setup.zip>

1. Install Node.js runtime <https://nodejs.org/en/>
2. Install the React SDK (installed by npm)
3. Install NVM
4. Visual Studio Code (maybe optional) <https://code.visualstudio.com/>
   1. Bash terminal option installed
5. Commands for runtime setup (run from VS Code bash terminal)
   1. npm install
   2. nvm install 12 (note: demo uses v12.22.7)
   3. nvm use 12
   4. npm install @auth0/auth0-react
   5. node –version ( should be 12.xx)
   6. npm run dev

Build Notes

# Application Setup

In the Auth0 console, ensure that you have an application provider configured that has the same parameters defined as the auth\_config.json. In this example, we have used “Cruise0” to name the application.

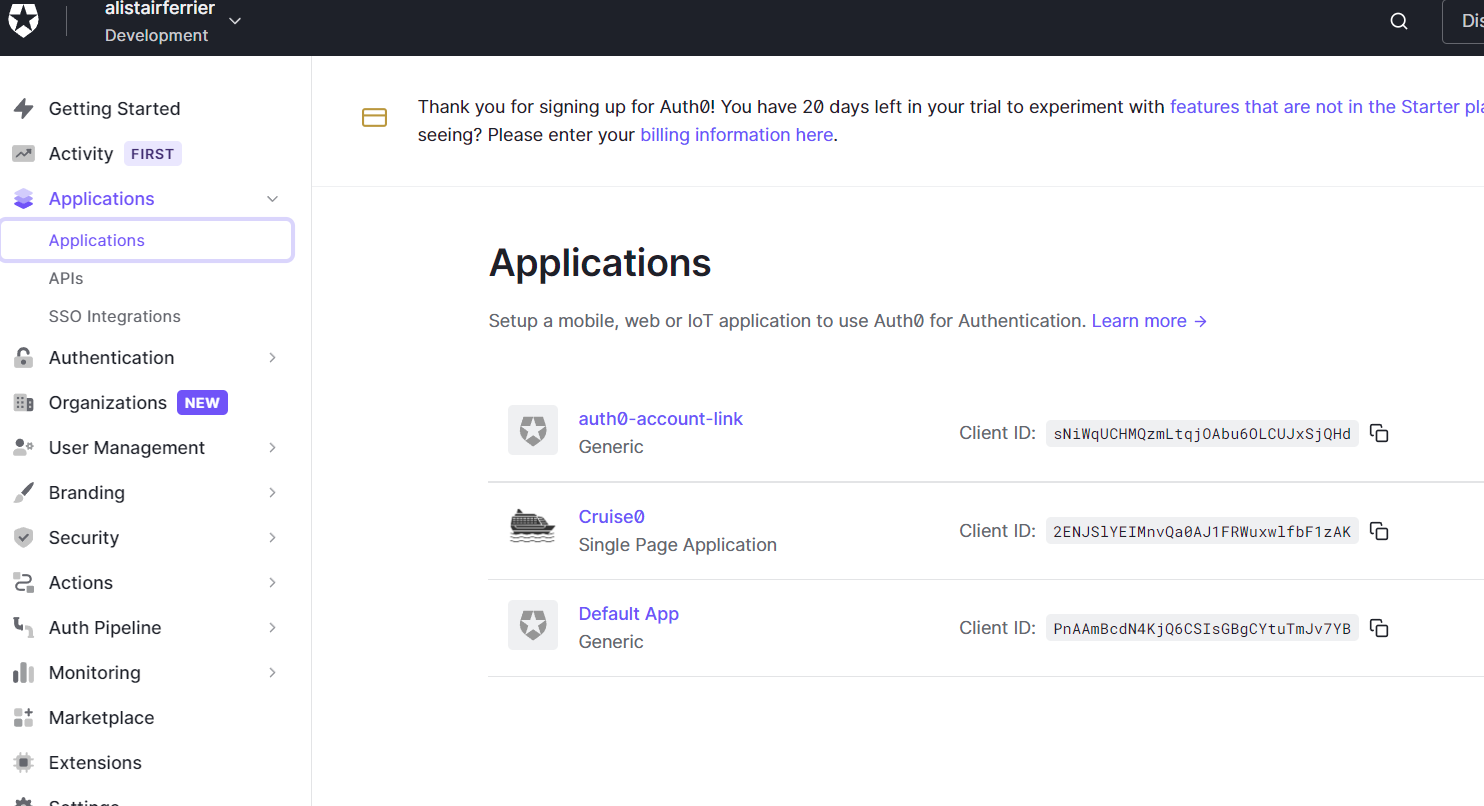
{

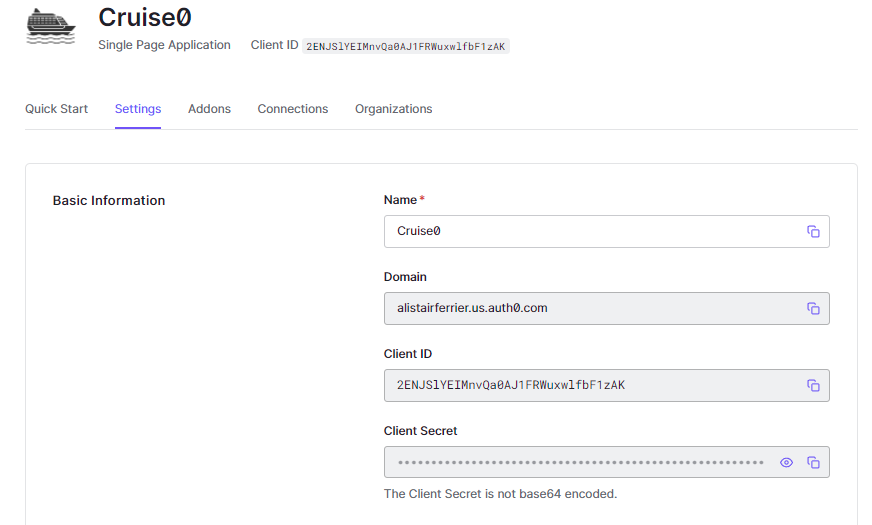
"domain": "alistairferrier.us.auth0.com",

"clientId": "2ENJSlYEIMnvQa0AJ1FRWuxwlfbF1zAK",

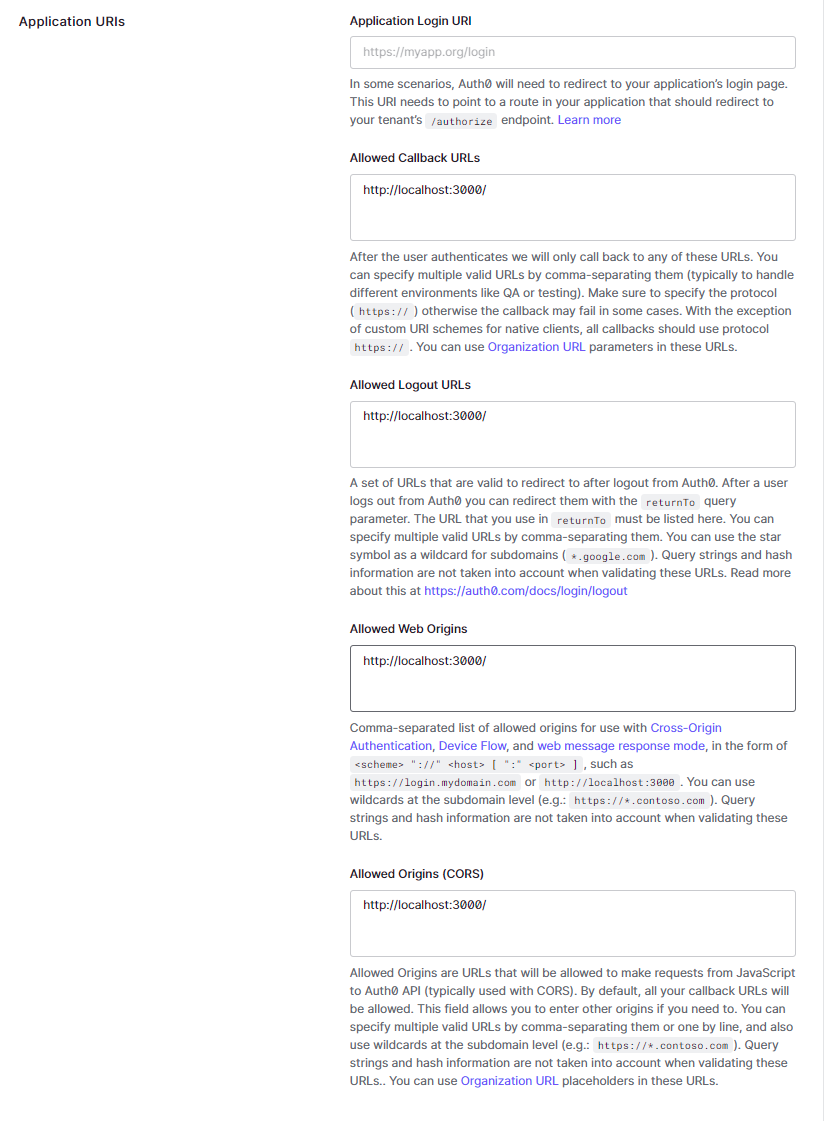
"audience": "YOUR\_API\_IDENTIFIER"

}

Allow the defaults for Domain, ClientID and Client Secret



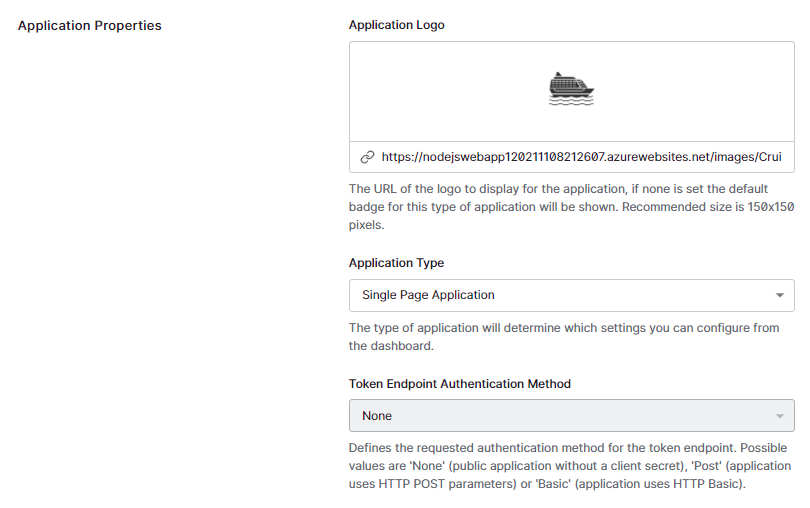
For the example of this demo, set the Allowed URIs to **http://localhost:3000/**



Set the Application Properties:

Logo URL

Application Type to “Single Page Application”



# Classic Universal Login Config

Although the Classic ULP was not used in this demo, this is an example of providing a background image in the HTML template. It was not possible to set the background image in the free tier using the New ULP.

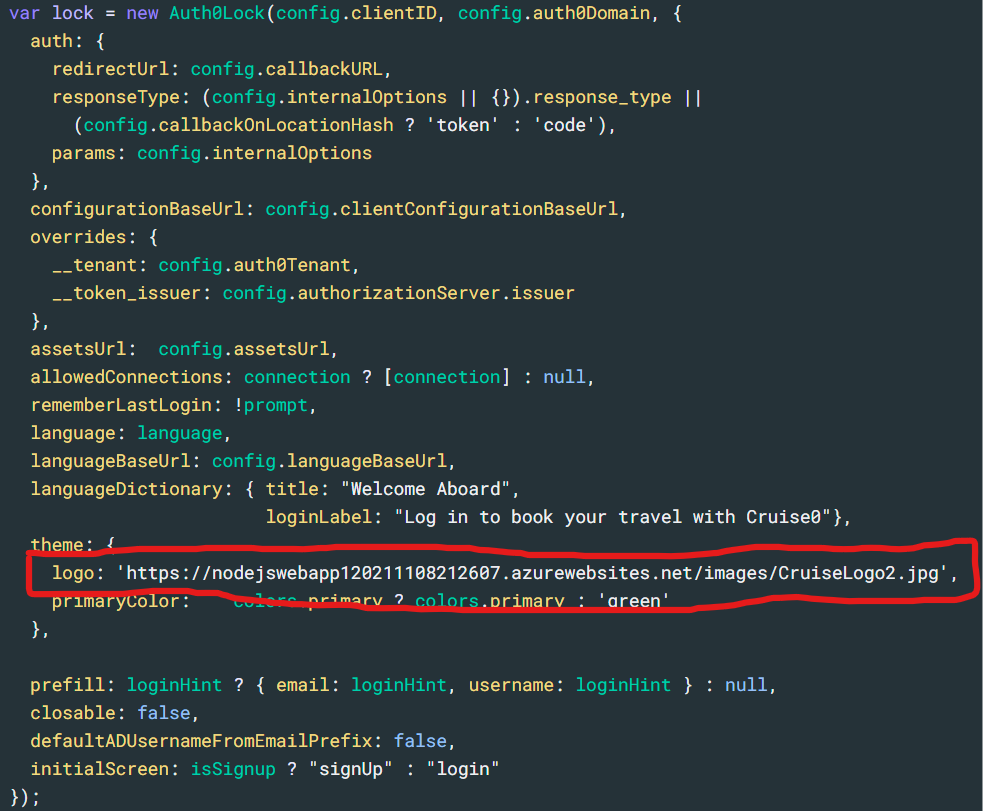
**Customised Text in Login Box:**

<https://auth0.com/docs/libraries/lock/lock-ui-customization#customizing-text>

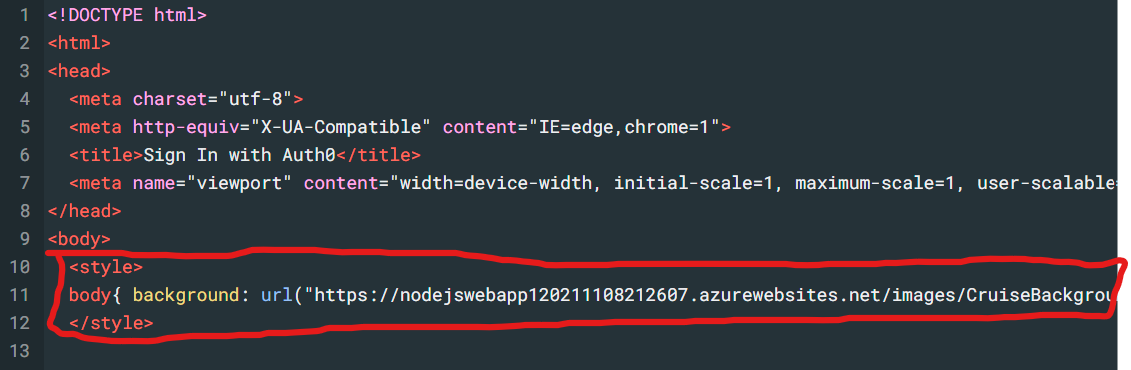
<https://github.com/auth0/lock/blob/master/src/i18n/en.js>

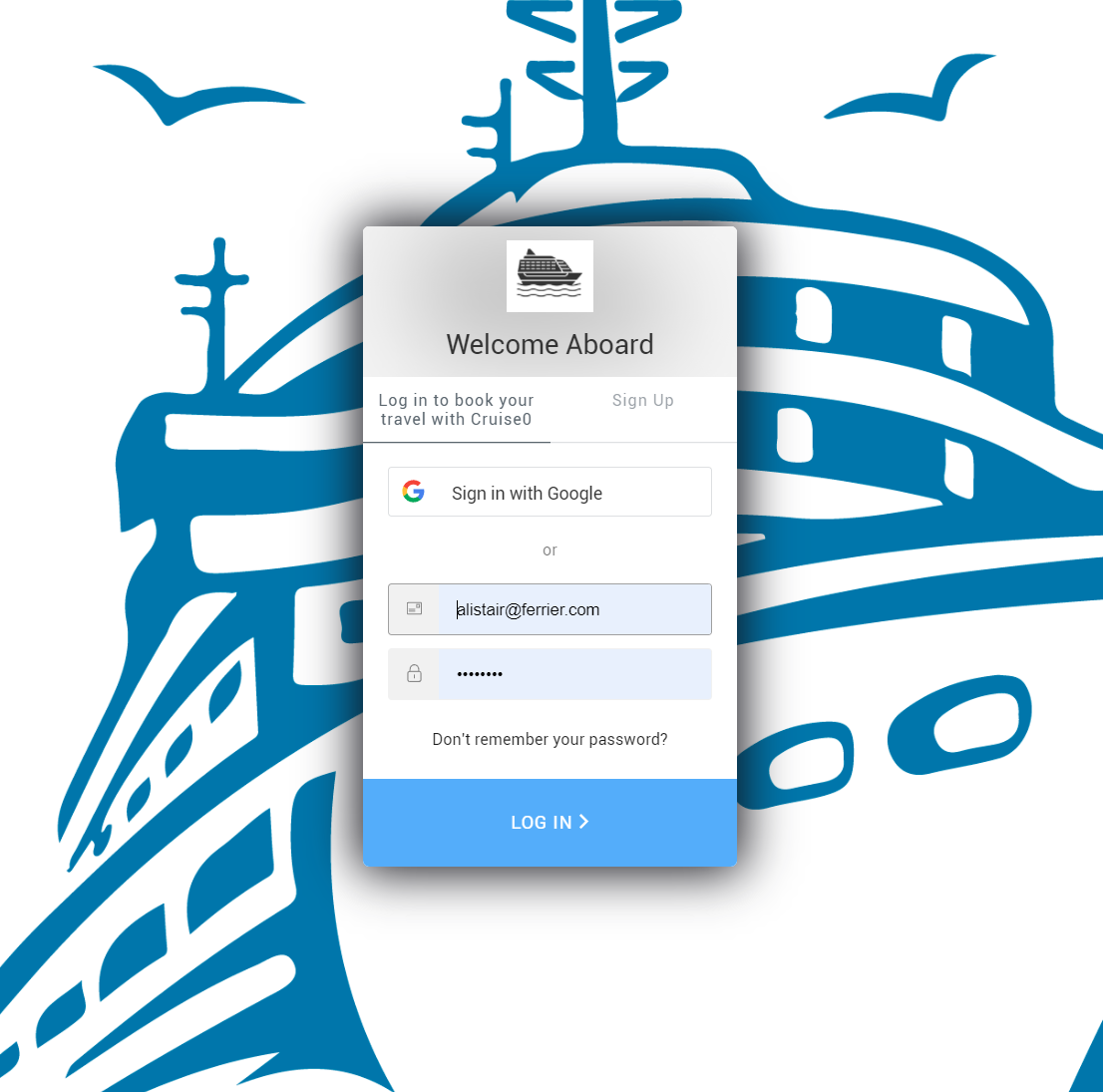
**Custom Cruise ship Icon on Logon Screen:**

Code modifications updated from Branding -> Universal Login in the Auth0 console:



**Custom Background on Login Screen:**





# Enable Direct Signup from page:

The following code was included to pass the parameter **screen\_hint=signup** to the /authorize call. The code also checks if the home is not logged in to display the signup button.



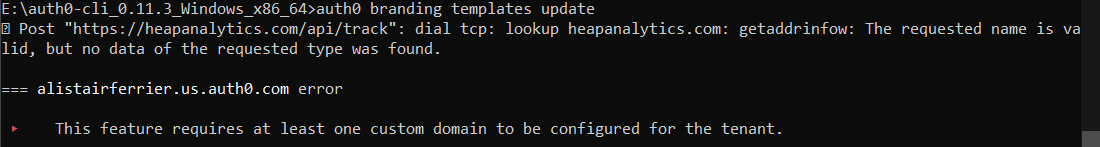
# New Universal Login Customisation

The following section outlines the customisations for the New Universal Login page used in the demo.

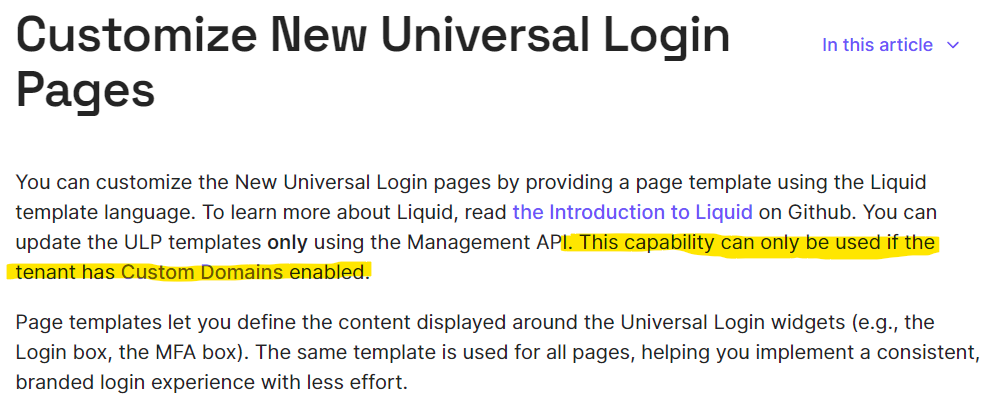
**Note:** To present a background image for the New Universal Login Page, it’s required to update the base template. Documentation states that this is only possible when using custom domains. This is not available on the free-tier that was being used. No branding update appeared to be available to the CLI for modification for the background-image URL. In a commercial/enterprise scenario, we would assume that custom domains would be used and this wouldn’t be a blocker.

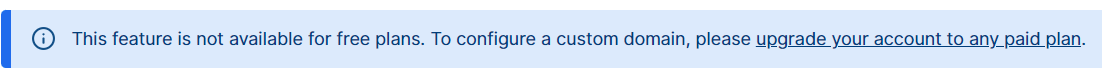
Branding texts were available for update, as per the challenge. They were modified using the Auth0 CLI.

Example of the issue:

****

And reference documentation <https://auth0.com/docs/brand-and-customize/universal-login-page-templates>

****

****

### Auth0 CLI Usage for Customisation

Use the Auth0 CLI, available from GitHub - <https://github.com/auth0/auth0-cli>

Follow read.me and install Windows latest executable and use auth0.exe

Commands:

**auth0 login**

**auth0 branding texts update login**

{

"login": {

"alertListTitle": "Alerts",

"buttonText": "Continue",

"description": "Log in book you travel with Cruise0",

"editEmailText": "Edit",

"emailPlaceholder": "Email address",

"federatedConnectionButtonText": "Continue with ${connectionName}",

"footerLinkText": "Sign up",

"footerText": "Don't have an account?",

"forgotPasswordText": "Forgot password?",

"invitationDescription": "Log in to accept ${inviterName}'s invitation to join ${companyName} on ${clientName}.",

"invitationTitle": "You've Been Invited!",

"logoAltText": "${companyName}",

"pageTitle": "Welcome Aboard",

"passwordPlaceholder": "Password",

"separatorText": "Or",

"signupActionLinkText": "${footerLinkText}",

"signupActionText": "${footerText}",

"title": "Welcome Aboard",

"usernamePlaceholder": "Username or email address"

}

}

**auth0 branding texts update signup**

{

"signup": {

"buttonText": "Continue",

"description": "Sign Up to book you travel with Cruise0",

"emailPlaceholder": "Email address",

"federatedConnectionButtonText": "Continue with ${connectionName}",

"loginActionLinkText": "Log in",

"loginActionText": "Already have an account?",

"logoAltText": "${companyName}",

"pageTitle": "Sign up | ${clientName}",

"passwordPlaceholder": "Password",

"passwordSecurityText": "Your password must contain:",

"separatorText": "Or",

"title": "Welcome Aboard",

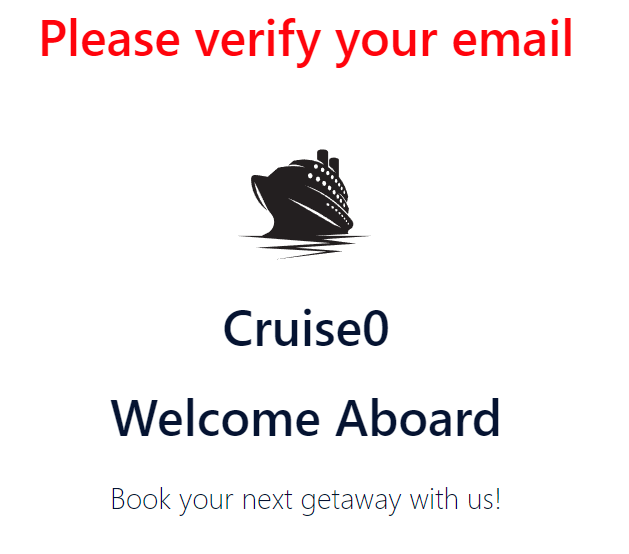
"usernamePlaceholder": "Username"

}

}

# Email Verification Warning

Changes made to home.js to display a warning that the user’s email has not been verified.



import React, { Fragment } from "react";

import { useAuth0 } from "@auth0/auth0-react";

import Hero from "../components/Hero";

import Content from "../components/Content";

const Home = () => {

const { user } = useAuth0();

return <Fragment>

{

**user !== undefined && !user.email\_verified && <h1 style={{**

**color: "red",**

**textAlign: "center"**

**}}>Please verify your email</h1>**

}

<Hero />

<hr />

<Content />

</Fragment>

}

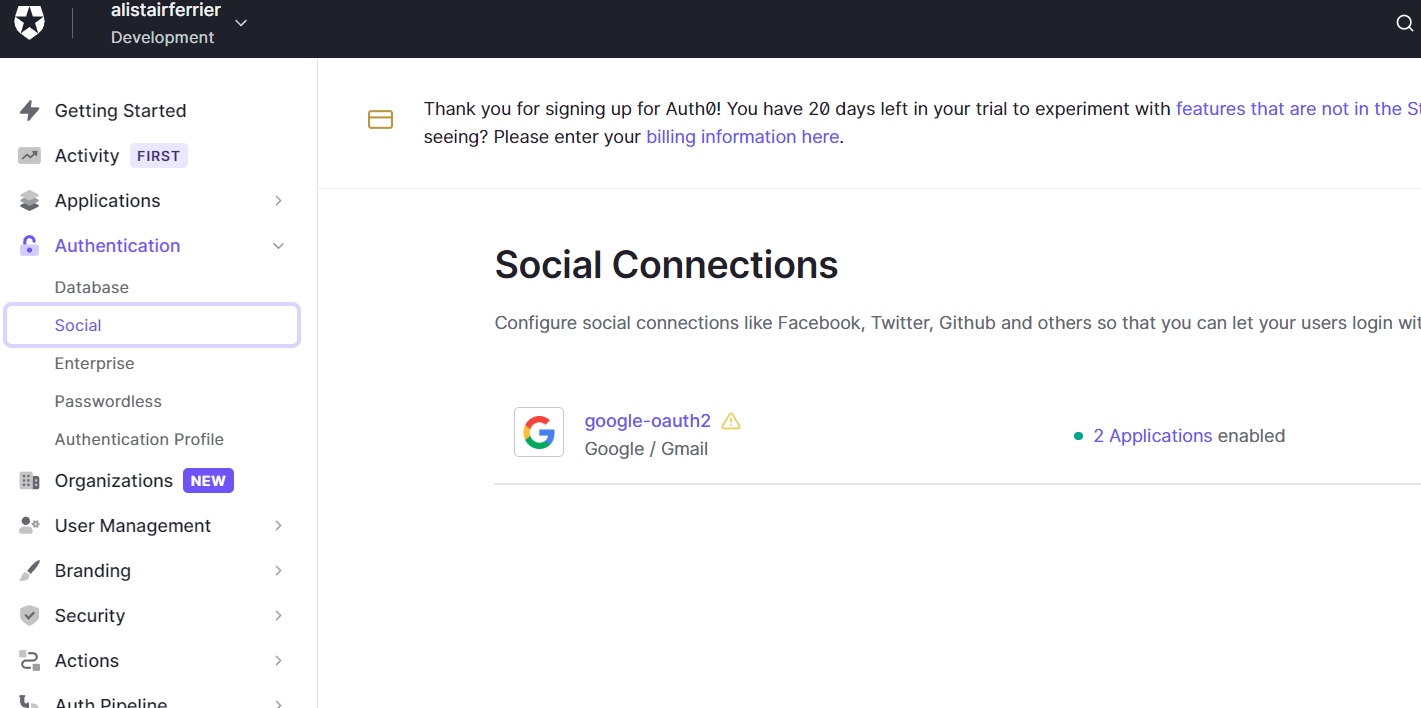
export default Home;

# Social Media Authentication

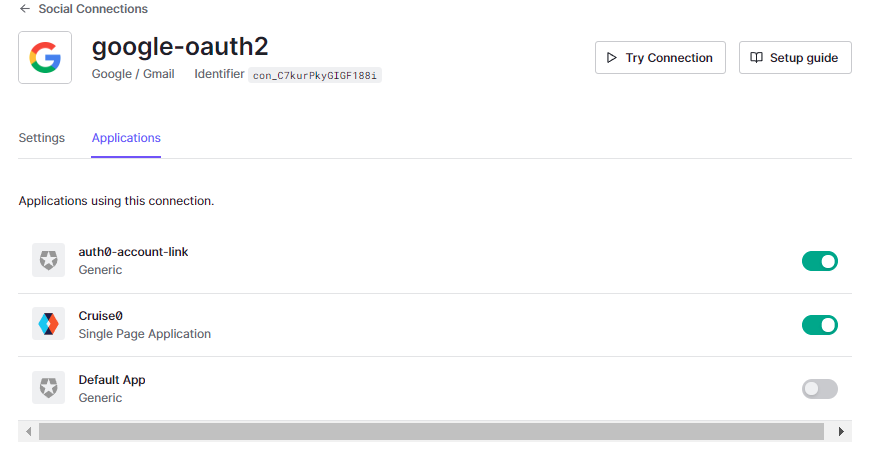
Ensure in the Auth0 console that you’ve added the Google connection available.

Review and complete the social connection instruction at <https://marketplace.auth0.com/integrations/google-social-connection>

Ensure you have a [**Google Developer Account**](https://console.developers.google.com/) and created a corresponding [**Google Project**](https://support.google.com/googleapi/answer/6251787?ref_topic=7014522).



Ensure that **Cruise0** and **auth0-account-link** are enabled.

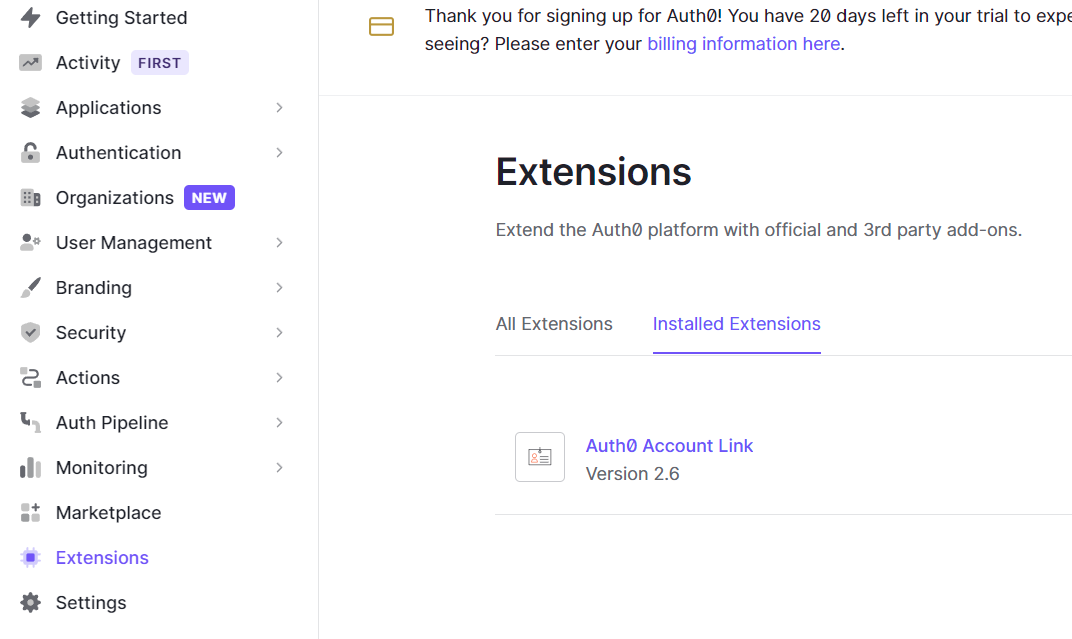


# Account Linking

The configuration of Account Linking resolves the problem of duplicate accounts with the same email address.

Install the Auth0 Account Link extension, under the extensions tab in the Auth0 console.

Docs for the account link extension: <https://auth0.com/docs/extensions/account-link-extension>



Select the **Auth0 Account Link** and update the HTML to customise the messaging for Cruise0 customers and save.

