Quiz Manager Setup

# Introduction

This document explains how to setup the Quiz Manager application such that it can be seen working on localhost. A general description of how to deploy Quiz Manager is also included in less detail.

Quiz Manager consists of two parts; a front-end React app and a back-end Web API. Authentication and authorization are provided by Auth0, a third-party provider.

# Prerequisites

## Software Requirements

Microsoft Visual Studio

Node.js

SQL database

## Other Requirements

Auth0 free account

(Optional) a logo file in svg format

(Optional) a publicly accessible URL to the logo

(Optional) hex codes for brand colours

# Assumptions

Quiz Manager was designed to allow the front-end and back-end to be hosted on separate domains. It is assumed that you will be running the front-end and back-end on separate domains.

It is assumed you will use <http://localhost:3000> as the domain for the front-end.

It is assumed you will use <https://localhost:44341> as the domain for the back-end.

This document will reference the above domains. If you are using different domains, you will need to use replace the domains in this document with the actual domains you are using.

# Setup

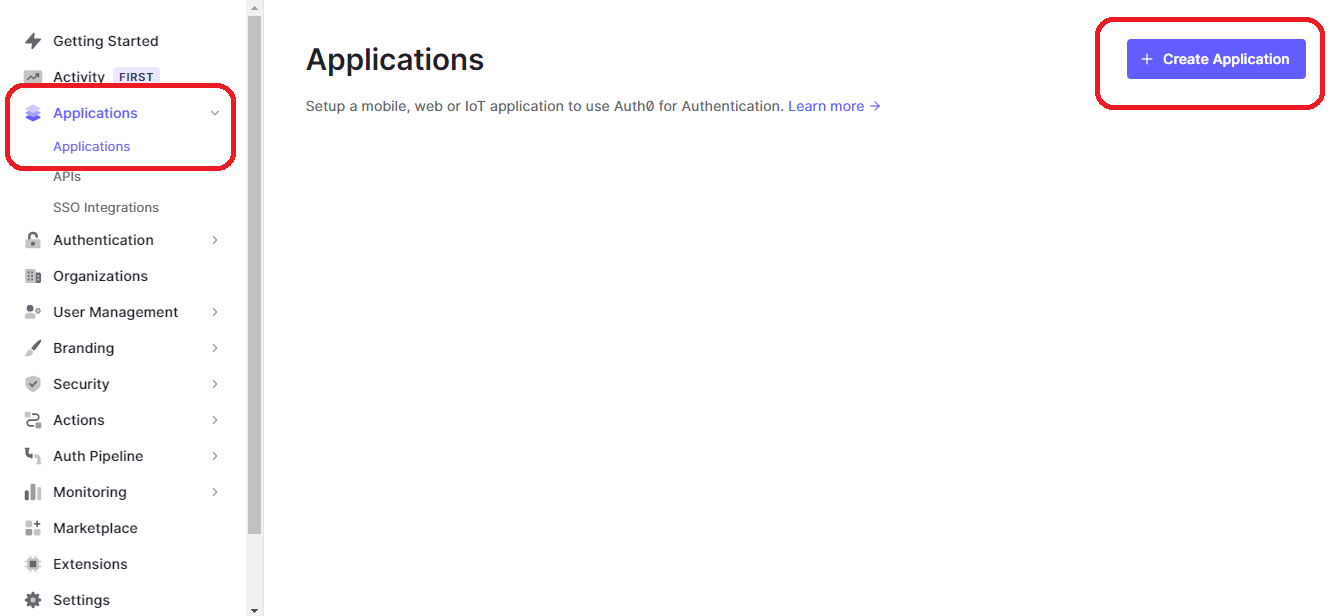
## Auth0

### Create Account

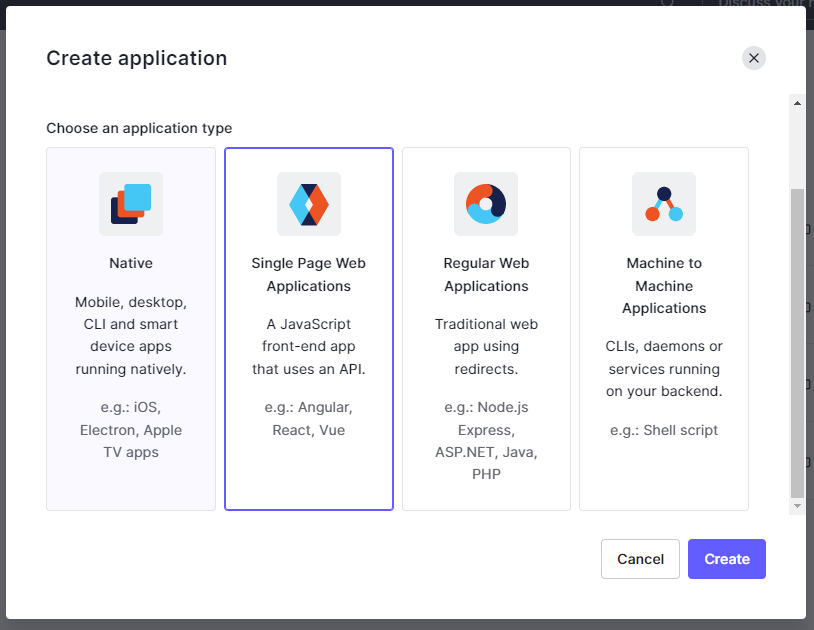
Visit the Auth0 website and create a free account <https://auth0.com/>

### Create Application

Go to Applications > Applications, then click Create Application.



Select Single Page Web Applications then click Create.



Go to the Settings tab.

Enter a name for your application in the Name field. It does not matter what you name the application.

(Optional) enter the URL to your logo in the Application Logo field.

Enter the following text in the Allowed Callback URLs field

http://localhost:3000, <http://localhost:3000/list>

Enter the following text in the Allowed Logout URLs field

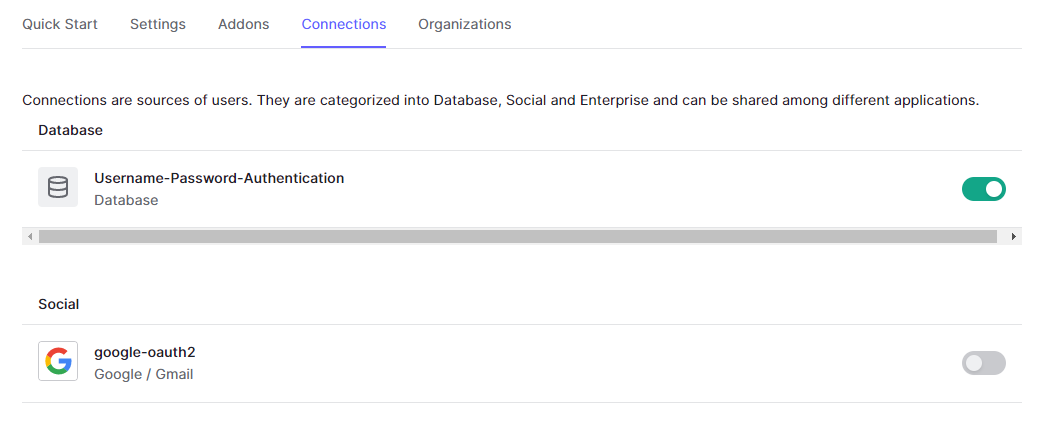
<http://localhost:3000>

Enter the following text in the Allowed Web Origins field

<http://localhost:3000>

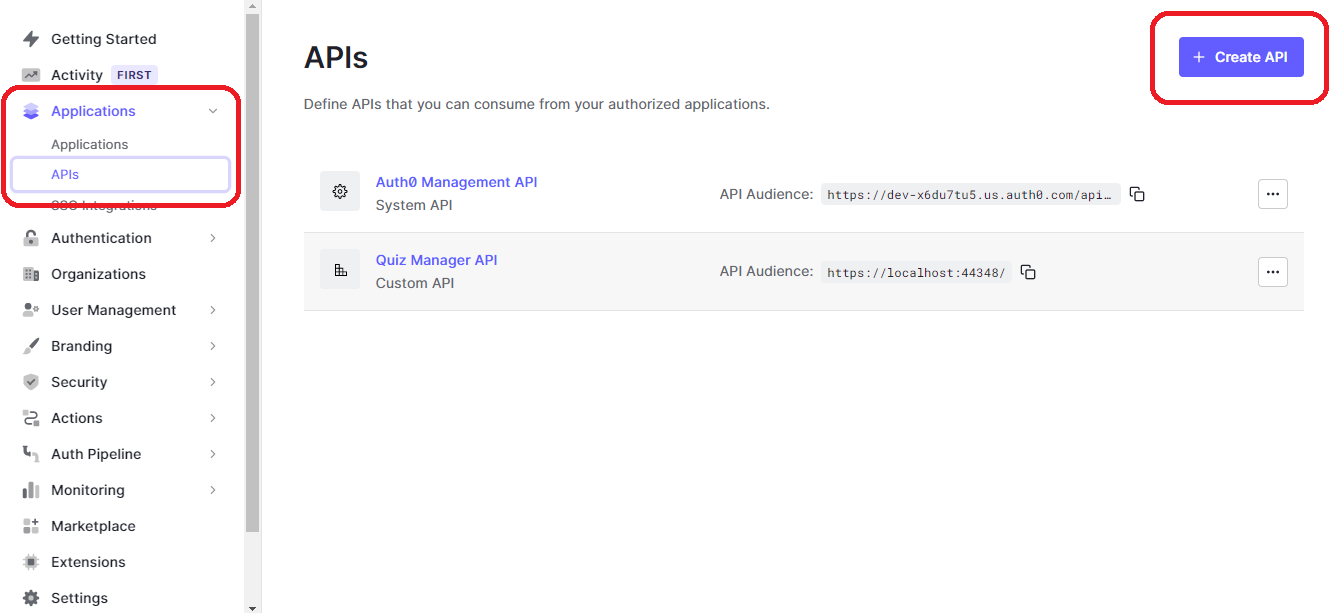
Click the Save Changes button at the bottom of the form.

Go to the Connections tab and disable the google-oauth2 setting

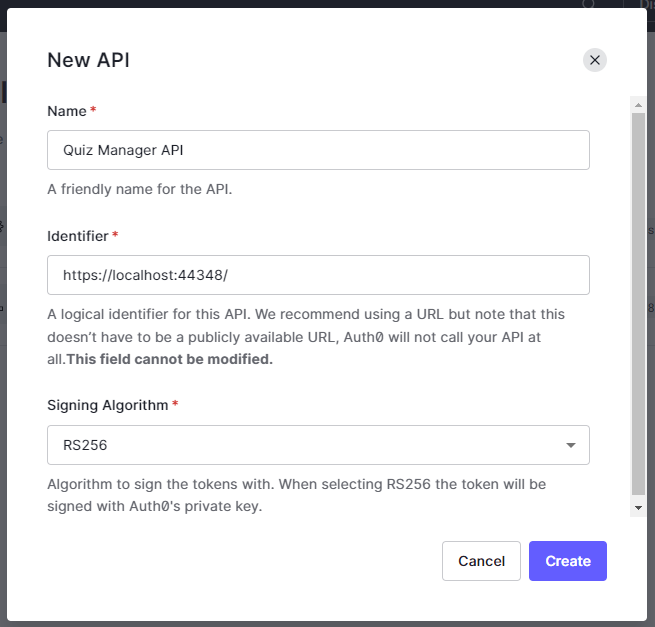


### Create API

Go to Applications > APIs then click Create API



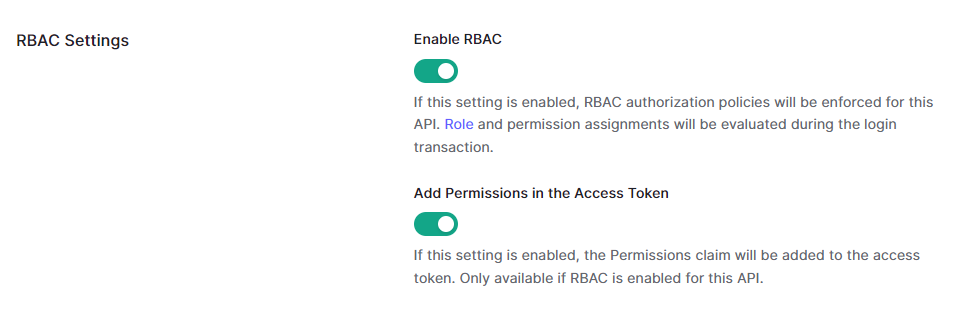
Enter a name and an identifier for your API. It does not matter what you enter into either of these fields.



Go to the Settings tab then scroll down to the RBAC Settings section. Turn on the following options

Enable RBAC

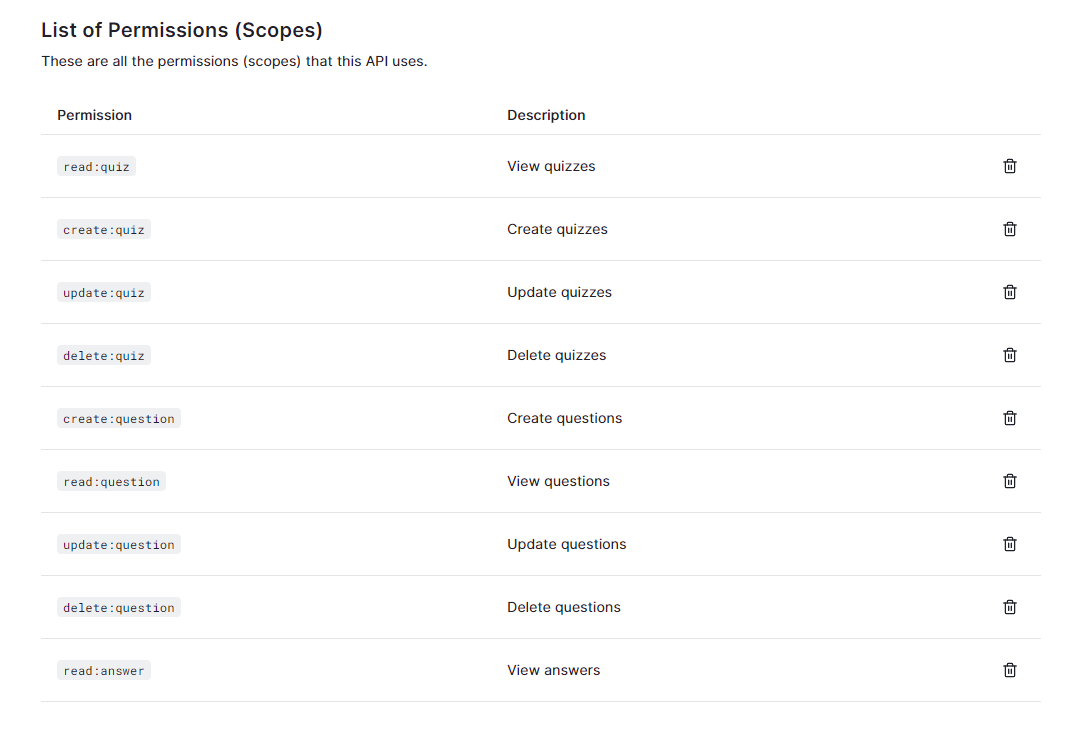
Add Permissions in the Access Token



Scroll down and click Save.

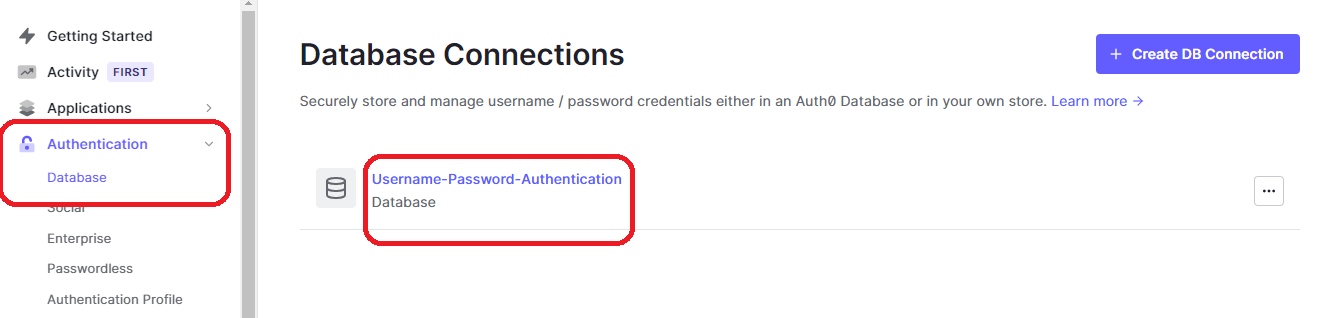
Go to the Permissions tab and create the following permissions

* read:quiz
* create:quiz
* update:quiz
* delete:quiz
* create:question
* read:question
* update:question
* delete:question
* read:answer

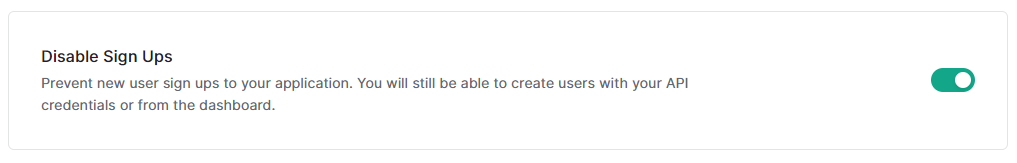


### Configure Authentication

Go to Authentication > Database then click on Username-Password-Authentication

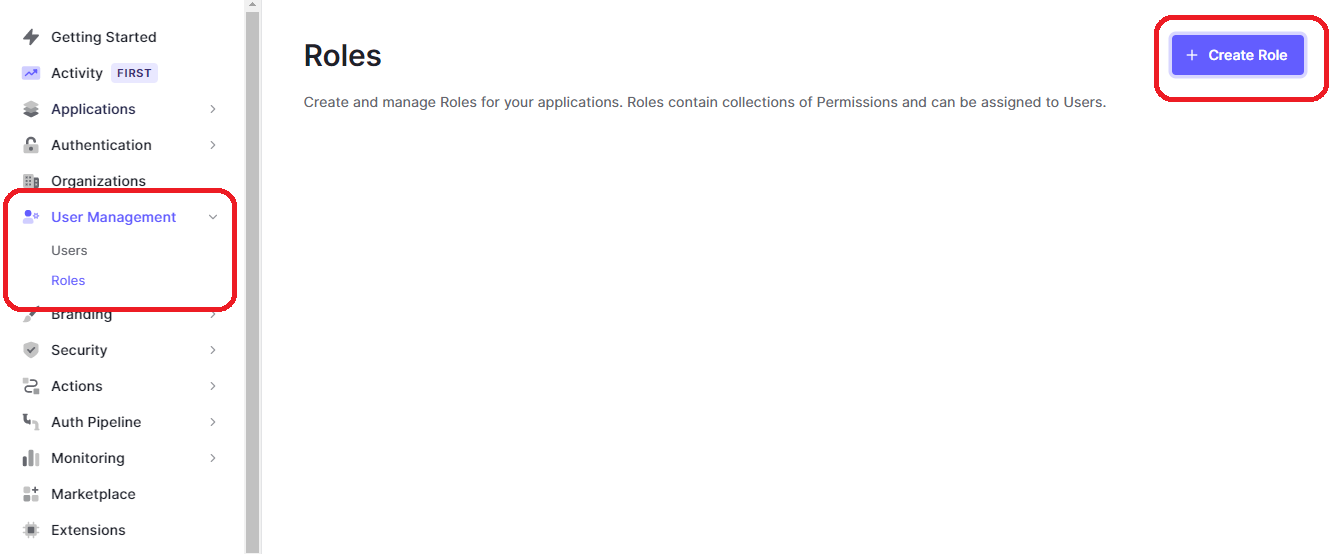


Enable the Disable Sign Ups setting

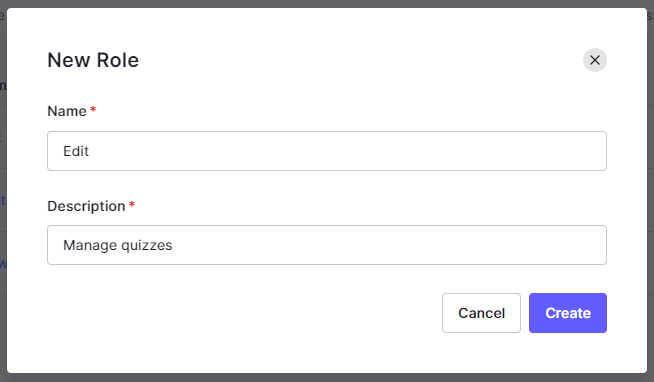


### Create Roles

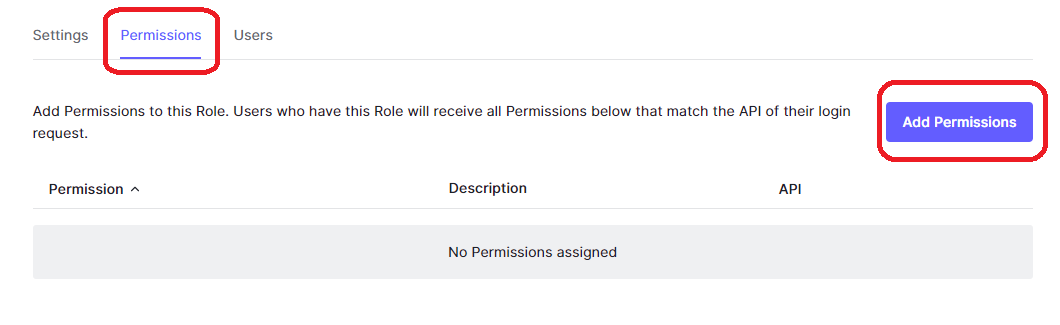
Go to User Management > Roles then click Create Role.



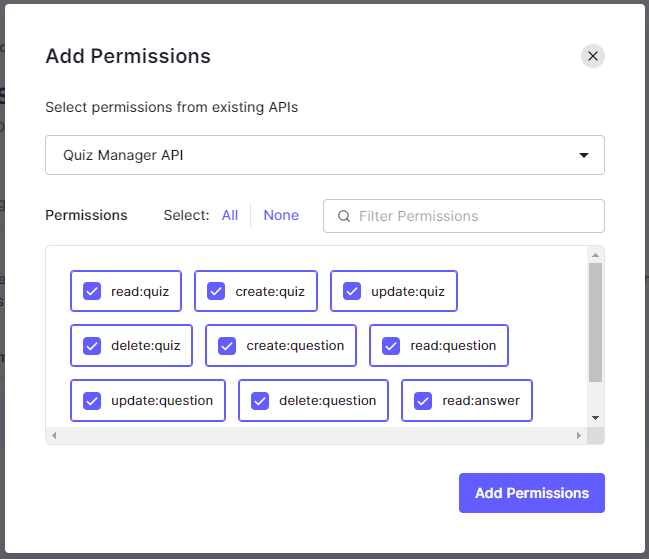
Enter “Edit” into the Name field and enter some text into the Description field.



Go to the Permissions tab and click Add Permissions



Select your API from the dropdown then select all permissions.



Click Add Permissions.

Go to User Management > Roles then click Create Role.

Enter “View” into the Name field and enter some text into the Description field.

Go to the Permissions tab and click Add Permissions

Select your API from the dropdown then select the following permissions

* read:answer
* read:question
* read:quiz

Click Add Permissions.

Go to User Management > Roles then click Create Role.

Enter “Restricted” into the Name field and enter some text into the Description field.

Go to the Permissions tab and click Add Permissions

Select your API from the dropdown then select the following permissions

* read:question
* read:quiz

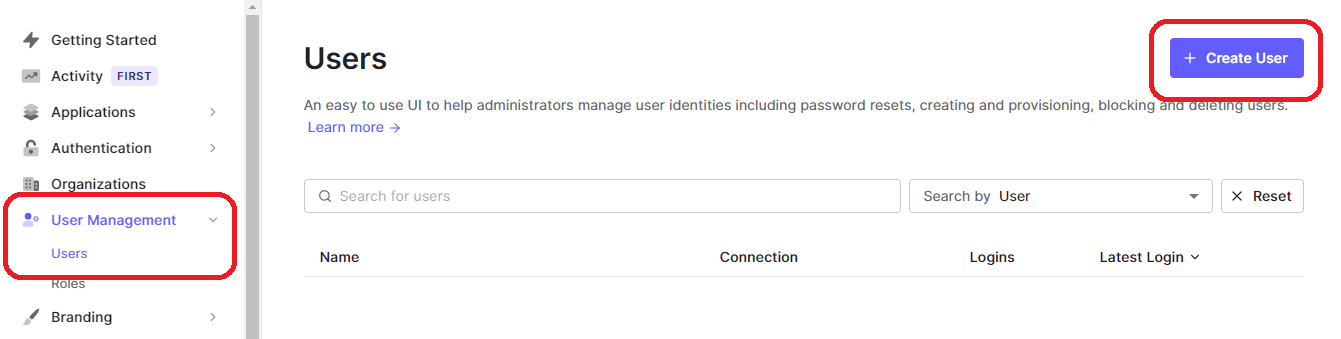
Click Add Permissions.

### Create Users

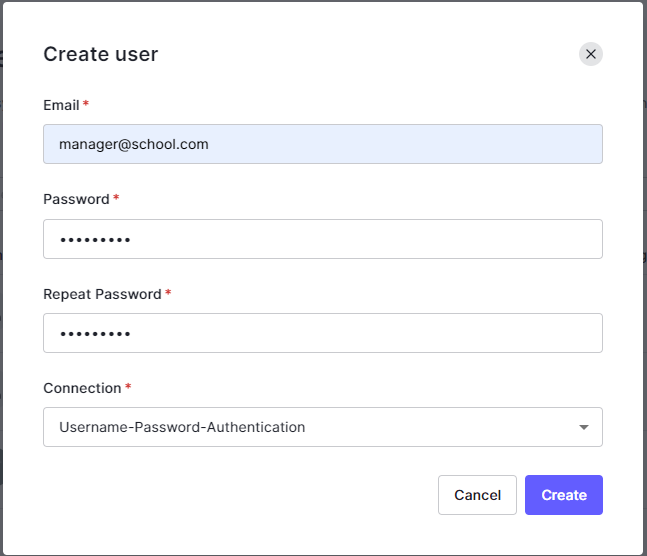
The following instructions explain a simple method for creating users in Auth0.

There are many methods available, such as importing from a CSV, importing from a database, linking to Active Directory or allowing users to self-register. These processes are documented thoroughly in the Auth0 Docs and a beyond the scope of this document.

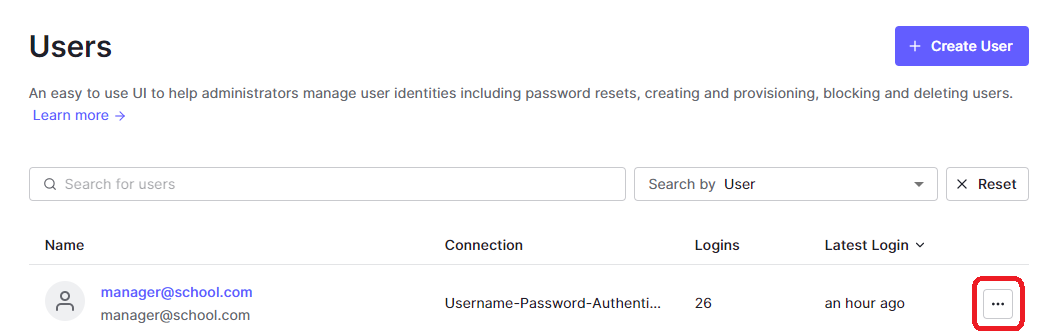
Go to User Management > Users and click Create User.



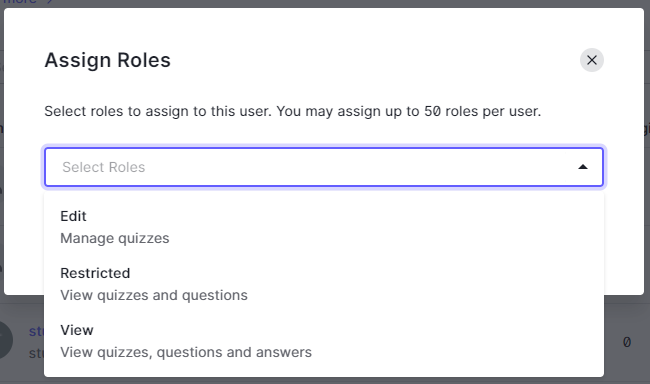
Enter credentials for the user and select Username-Password-Authentication in the Connection field.



Click the ellipsis button for the user then click Assign Roles



Assign whichever roles you like to the user



*These roles are additive, which means the user will be granted the level of access of all applied roles combined.*

*For example, if a user has both the Restricted and View roles, they will be able to view answers. A user with only the Restricted role would not be able to view answers.*

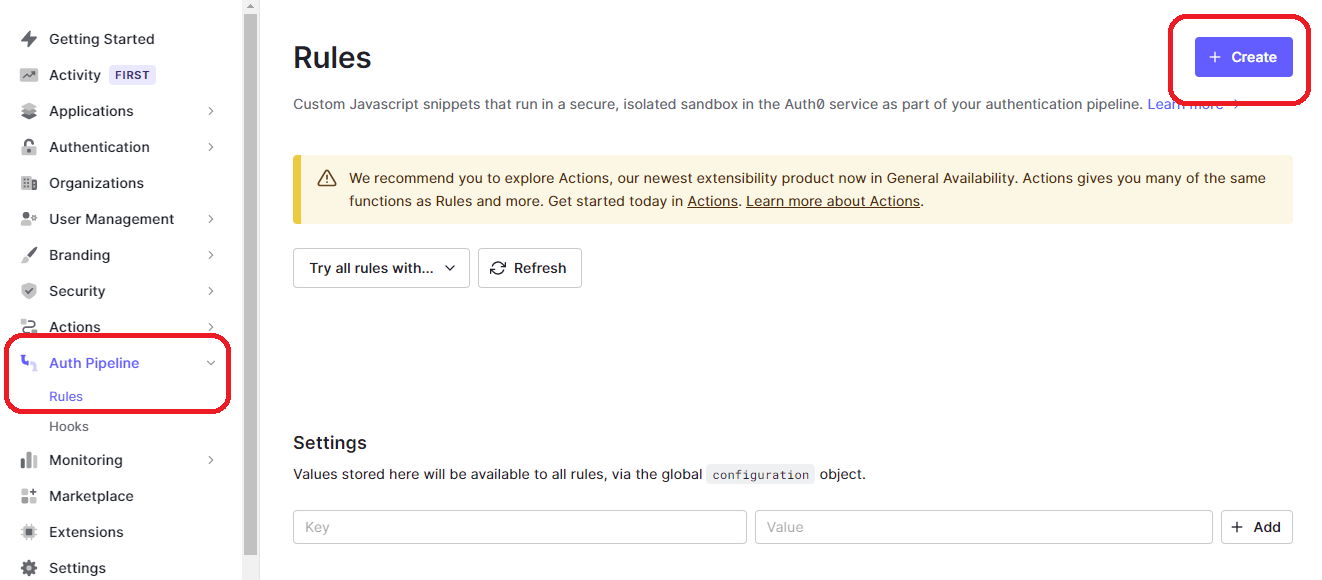
### Branding (optional)

Go to Branding > Universal Login.

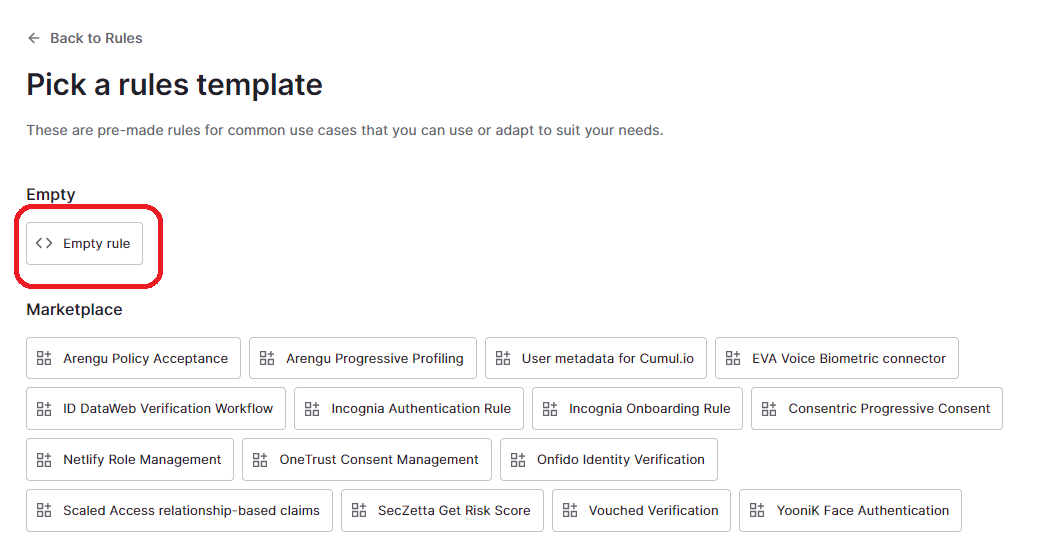
Enter the URL to the logo, the primary brand colour and the background colour here.

### Auth Pipeline

Go to Auth Pipeline > Rules and click the Create button.



Select the <> Empty rule option



Enter Add roles to access token into the Name field

Enter the following script into the Script field

function (user, context, callback) {

const namespace = 'http://localhost:3000';

const assignedRoles = (context.authorization || {}).roles;

let idTokenClaims = context.idToken || {};

let accessTokenClaims = context.accessToken || {};

idTokenClaims[`${namespace}/roles`] = assignedRoles;

accessTokenClaims[`${namespace}/roles`] = assignedRoles;

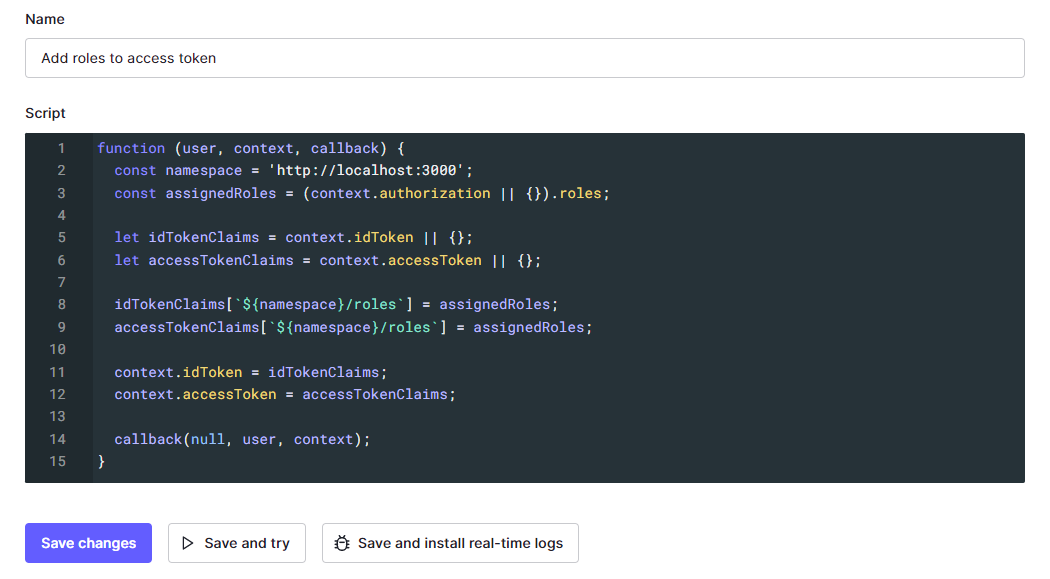
context.idToken = idTokenClaims;

context.accessToken = accessTokenClaims;

callback(null, user, context);

}

Click Save Changes



## Quiz Manager Client

This project is stored in the quiz-manager-client folder, within the Applications folder.

### Install Dependencies

Using the command line, navigate to the quiz-manager-client folder then enter the following command:

npm i

### Environment Variables

Edit the .env file which is found via:

quiz-manager-client\src\.env

This file contains a number of settings. The name of each setting appears in capital letters, followed by an equal sign, followed by the value of the setting.

Set the values for the following settings

|  |  |
| --- | --- |
| **Setting Name** | **Value** |
| REACT\_APP\_AUTH0\_DOMAIN | The domain of your Auth0 application |
| REACT\_APP\_AUTH0\_CLIENTID | The client ID of your Auth0 application |
| REACT\_APP\_AUTH0\_ROLES\_KEY | http://localhost:3000/roles  *(unless you changed this value in the script in the Auth Pipeline section above)* |
| REACT\_APP\_AUTH0\_AUDIENCE | The API Audience of your Auth0 API |
| REACT\_APP\_ORGANISATION | The name of the organisation that is using the Quiz Manager app. This is displayed on the login page. |

### Logo (optional)

The logo used by the application is found via:

quiz-manager-client\src\logo.svg

Replace this with your logo. The new file must be called logo.svg

### Branding / Colours (optional)

The colours used by the application can be modified as needed. The colours are stored in the following file:

quiz-manager-client\src\colors.scss

Edit these colours as desired

Notes

$light is the background of the navigation bar

$body-bg is the background of the application

## Quiz Manager API

This project is stored in the QuizManager folder, within the Applications folder.

### Application Settings

The settings for the API are stored in the appsettings.json file which is accessed via:

QuizManager\QuizManager.Api\appsettings.json

This file contains several settings. The name of the setting appears, followed by a colon, followed by the value of the setting.

Enter the following values

|  |  |
| --- | --- |
| **Setting Name** | **Value** |
| Database | Connection string to your database |
| Domain | The Domain field for your Auth0 API |
| Audience | The Audience field for your Auth0 API |
| ClientOrigin | The URL for the front-end application  *Do not include a trailing slash* |

# Running

## Client

Using the command line, navigate to the quiz-manager-client directory then run the following command

npm start

## API

Open the QuizManager solution in Visual Studio then click the Start button in the IDE.

# Deployment

Giving instructions for deployment is beyond the scope of this document, but the following links may be useful:

## API

<https://docs.microsoft.com/en-us/aspnet/web-api/overview/hosting-aspnet-web-api/>

## Client

<https://create-react-app.dev/docs/deployment/>