Google Authentication with ReactJS and Typescript

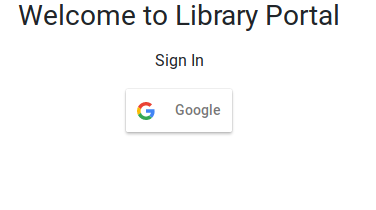
2021-01-05 09:39:00 +0530

# Introduction

Today, we will see how to integrate *Google Authentication* with a front-end application written in ReactJS, Typescript combination.

# Application flow

We would be building a mock Library software, where people can login to add and browse books. This would be secured with Google Authentication. This is how the login screen would look like:



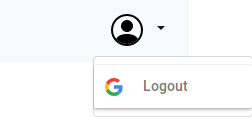
Google login screen

When the user clicks on the *Google button*, he would be re-directed to login into his Google account. On successful login, the user would be able to see the application screen:



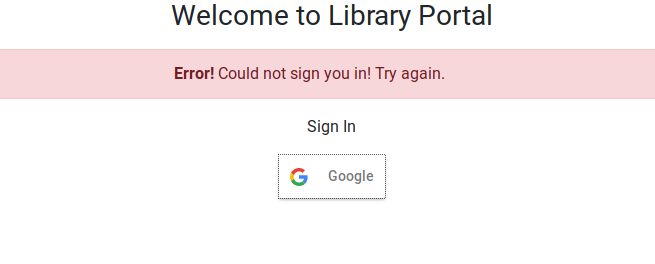
Library application screen

In addition to that, the user would also see a logout button on the Menu Bar as shown below:



Google logout button

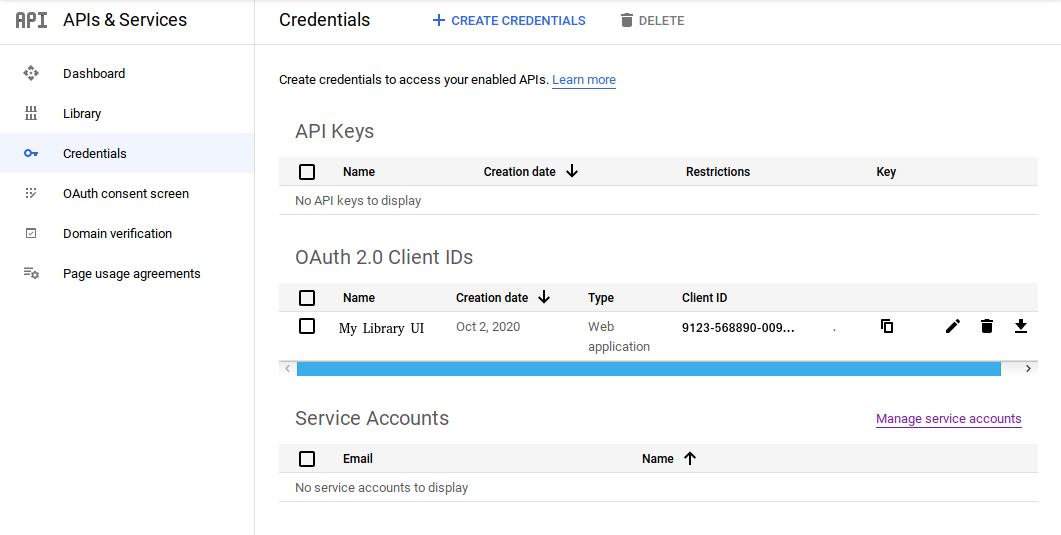
However, if the Google login fails, an error message is displayed as shown below:



Google login failure

# Before we start

We would first, need to create an account in Google and create the authorization credentials. Here is a post that can guide you do that: <https://developers.google.com/identity/sign-in/web/sign-in>. Follow the steps, at the end of it, you should see something like this on your [Credential Console](https://console.developers.google.com):



Google Credential Console

Note down the *Client ID*, as we would need that to authenticate users into our App.

# Implementation Details

I took a short-cut and used found the [React Google Login](https://www.npmjs.com/package/react-google-login) very handy. This has ready-made buttons and hooks that facilitate login and logout.

## Installation

npm install react-google-login

## Login Screen

This is how my *GoogleSignInComponent* looks like:

import React, { FunctionComponent, useState } from 'react'  
import { GoogleLogin, GoogleLoginResponse, GoogleLoginResponseOffline } from 'react-google-login';  
import Alert from './Alert';  
import AlertType from './AlertType';  
  
interface GoogleSignInComponentProps {  
 loginSuccess: (response: GoogleLoginResponse | GoogleLoginResponseOffline) => void;  
}  
  
export const GoogleSignInComponent: FunctionComponent<GoogleSignInComponentProps> = ({ loginSuccess }) => {  
  
 const [loginFailed, setLoginFailed] = useState<boolean>();  
  
 return (  
 <div className="text-center mb-4">  
 <h1 className="h3 mb-3 font-weight-normal">Welcome to Library Portal</h1>  
 {loginFailed &&  
 <Alert message="Could not sign you in! Try again." type={AlertType.ERROR} />  
 }  
 <p>Sign In</p>  
 <GoogleLogin  
 clientId={`${process.env.REACT\_APP\_GOOGLE\_AUTH\_CLIENT\_ID}`}  
 buttonText='Google'  
 onSuccess={loginSuccess}  
 onFailure={(response: any) => {  
 setLoginFailed(true);  
 }}  
 cookiePolicy={'single\_host\_origin'}  
 responseType='code,token'  
 />  
 </div>  
 )  
};

## Logout Button

I create the logout button in the *NavBar* class, as a *MenuItem*.

<ul className="navbar-nav">  
 <li className="nav-item dropdown">  
 <a className="nav-link dropdown-toggle" href="#" id="userProfile" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">  
 <img src="/person-circle.svg" alt="" width="52" height="32" title="UserProfile"></img>  
 </a>  
 <div className="dropdown-menu" aria-labelledby="userProfile">  
 <GoogleLogout  
 className="dropdown-item"  
 clientId={`${process.env.REACT\_APP\_GOOGLE\_AUTH\_CLIENT\_ID}`}  
 buttonText="Logout"  
 onLogoutSuccess={() => menuActionListener(MenuAction.LOGOUT)}  
 >  
 </GoogleLogout>  
 </div>  
 </li>  
</ul>

# Sources

A fully working code can be found here: <https://github.com/paawak/blog/tree/master/code/reactjs/library-ui-secured-with-google>