Okta React Js Web App Integration

Below are the Steps for Integrating Okta Server Authentication with ReactJs Web Application

# OKTA Server Configurations

## Create Okta Dev Account

You can create Okta developer Account on [Okta Developer](https://developer.okta.com/signup/) portal.

Below is an example of Okta Developer Account created (for this Demo )

Developer Okta Account portal created : <http://dev-65193610.okta.com>

Server Admin : [prakhour@microsoft.com](mailto:prakhour@microsoft.com)

Password : \*\*\*\*\*\*\*\*\*

## Create New App integration for React App

Once the Okta Account is created , You can login to the new Okta account portal ( <http://dev-65193610.okta.com> as shown in this case ) using account credentials ( email and password ) as created in step 1.

Graphical user interface, text, application, email

Description automatically generated

Once you are logged in , You can navigate to Applications link and create a **App integration Token** using button “**Create App Integration**” button as shown in below screenshot.

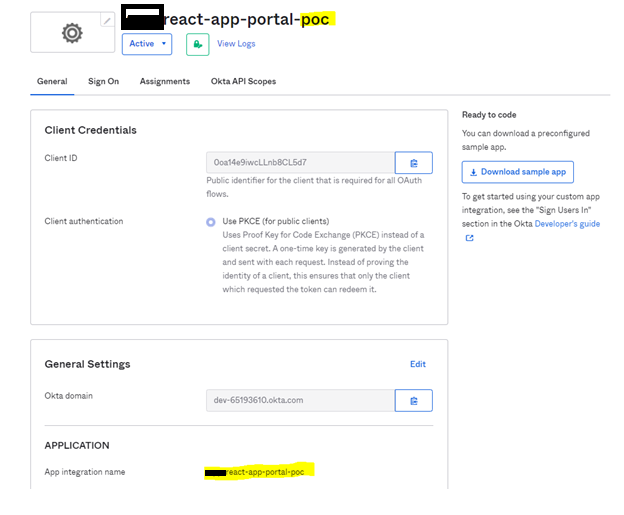
Give any Name to the App registration ( for Eg . react-app-poc , as in this example ) Graphical user interface, application, Word

Description automatically generated

Since we are going to use this App integration in React Web Application (with a Okta Sign In page) , so we will be selecting a **OIDC** ( Token based OAuth 2.0 Sign in method ) and ApplicationType as **Single Page Application** as shown below .

Graphical user interface, text, application

Description automatically generated

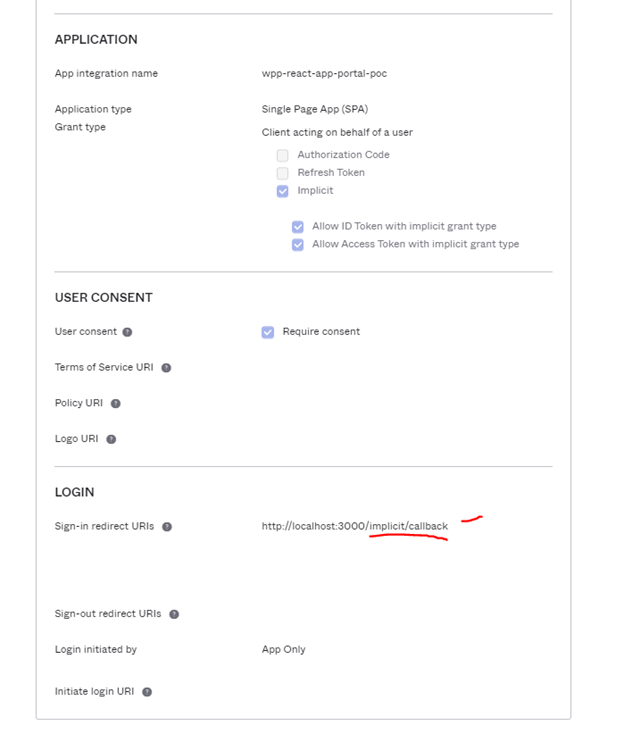


Note the Client ID as it will be used in ReactJs app Okta sign in parameter section.

Select Grant Type : **Implicit**

andIn SignIn redirect Url as **http://localhost:3000/implicit /callback**

Note : We will integrate implicit callback url in our React Js application signin parameters section



Additionally , You can also specify Sign out redirect Url same as login page url.

## Enable CORS on Okta Developer portal for the react app login url

To enable CORS for our web app , Go to **Security > API > Trusted Origins** tab

Click **Add Origin** button

You can provide a **Name** for this origin (say localhost:3000)

Type **Origin Url** as <http://localhost:3000>

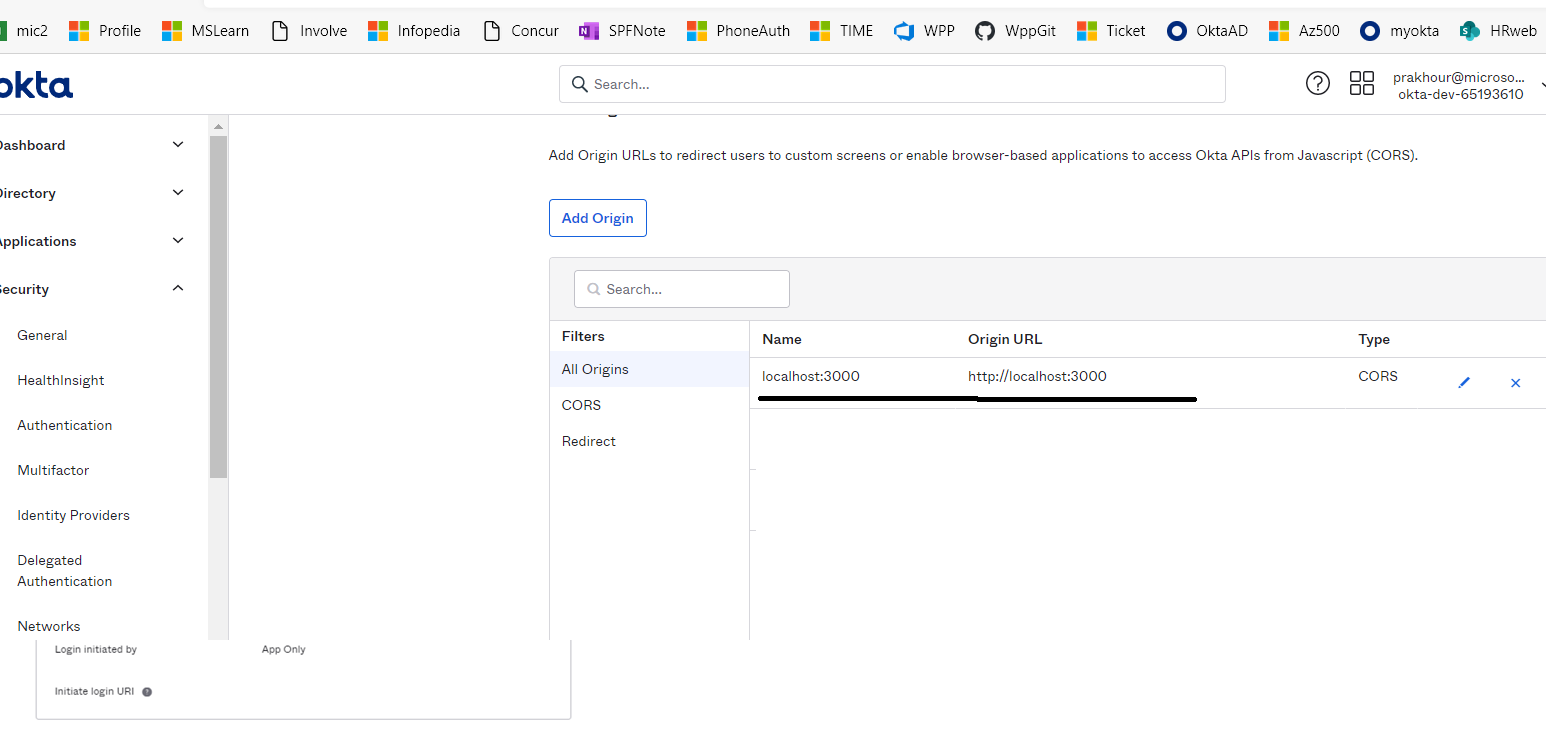
(note we are using 3000 port number as ReactJs app runs on port 3000 by default in local debug mode ) Port can be changed to local debug react app port number.

Select the **2 checkboxes** and **Save**

Graphical user interface, text, application, email, Teams

Description automatically generated

New *Localhost:3000* Origin will get added to the list



**NOTE** : Under Authorization Servers , there must be a default authorization server settings already created for the okta developer portal . We will be using this authorization server in react app for pour demo

Graphical user interface, text, application, chat or text message, email

Description automatically generated

Graphical user interface, application, email

Description automatically generated



## Add other Okta users and assign to App integrations

You can add login user to your react App in Okta porta under the below section

Goto **Directory > People** > Click **Add Person**

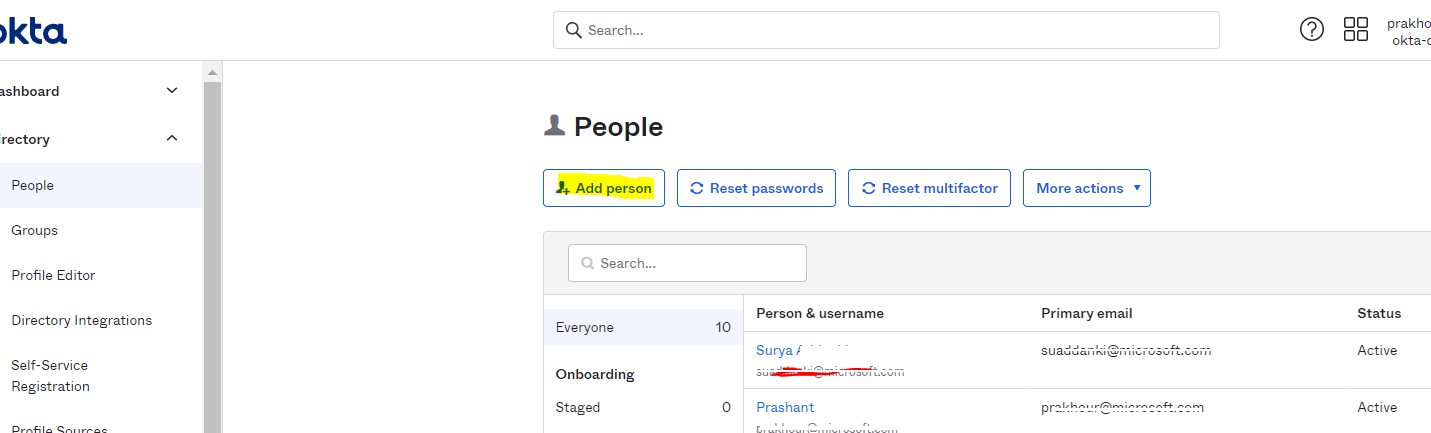
Search for the **First name , Last name , Email id** of the user to be added . Then click **Save**.

NOTE : A mail will be sent to the recipient user to accept the Okta login invitation and will be asked to reset password. After this activity the recipient user will be able to login to the Okta server using the set credentials.

Graphical user interface, application, email

Description automatically generated

New user will get added to the list .



Now Confirm/Add User Assigment

Goto existing **App integrations** (react-app-portal-poc) > **Assignment** tab

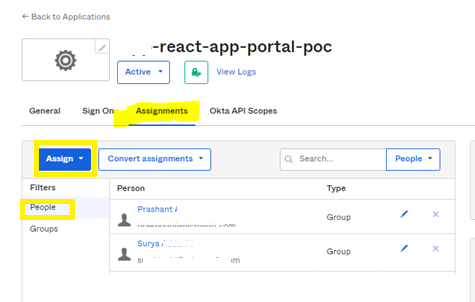
Select **Assign > Assign to People**

Graphical user interface, text, application, chat or text message

Description automatically generated

In the new pop up window **,Search** the new user email id ,you want to add to the list and click **Assign** .

Click **Done** to add to the list of assigned users .



# Build React Js Application with Okta Server

Refer/Download the Okta Sign In ReactApp code attachment provided with this document.

( OktaReactAppIP.zip )

Refer package.json file has below react and okta package dependencies added to it :

react-router-dom

okta/okta-auth-js

okta/okta-react

okta/okta-signin-widget

Graphical user interface, text

Description automatically generated

Refer the Configuration sections in config.Js file where the Okta developer server url is mentioned with App integration Client ID

Text

Description automatically generated

Open the Code in VS Code

Install the okta packages using below command in VS Code terminal window

* **npm install @okta/okta-signin-widget**
* **npm install @okta/okta-auth-js @okta/okta-react react-router-dom**

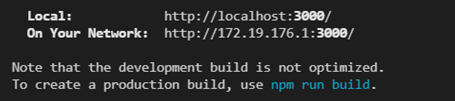
Run the Application using below commands in terminal

* **npm install**

Text

Description automatically generated

* Run command **npm start**



You should get the Okta Signin page hosted on <http://localhost:3000/>

Type in your oktas user credentials (email id & password ) as registered with Okta earlier.

You should be redirected to home page on Successful login.

# Okta React Portal Demo Screenshots

Okta Sign-In page : Enter your Okta user credentials

Graphical user interface, application

Description automatically generated

Home Page : **Home** is the page displayed on successful login to Okta React portal

Graphical user interface, text

Description automatically generated

SignOut : Click Sign Out link to logout from Okta react demo application and move back to Login page

Graphical user interface, application

Description automatically generated