1. **Okta Angular + Okta Hosted Login Example**

This shows you how to use the [Okta Angular Library] to login a user to a Angular application.  The login is achieved through the [PKCE Flow], where the user is redirected to the Okta-Hosted login page.  After the user authenticates they are redirected back to the application with an ID Token and Access Token.

This example is built with [Angular CLI].

**## Prerequisites**

Before running this sample, you will need the following:

\* An Okta Developer Account, you can sign up for one at https://developer.okta.com/signup/.

\* An Okta Application, configured for Singe-Page App (SPA) mode. This is done from the Okta Developer Console and you can find instructions [OIDC SPA Setup Instructions].  When following the wizard, use the default properties.  They are designed to work with our sample applications.

**## Running This Example**

To run this application, you first need to clone this repo and then enter into this directory:

```bash

git clone https://github.com/okta/samples-js-angular.git

cd samples-js-angular/okta-hosted-login

```

Then install dependencies:

```bash

npm install

```

Now you need to gather the following information from the Okta Developer Console:

- **\*\*Client Id\*\*** - The client ID of the SPA application that you created earlier. This can be found on the "General" tab of an application, or the list of applications.  This identifies the application that tokens will be minted for.

- **\*\*Issuer\*\*** - This is the URL of the authorization server that will perform authentication.  All Developer Accounts have a "default" authorization server.  The issuer is a combination of your Org URL (found in the upper right of the console home page) and `/oauth2/default`. For example, `https://dev-1234.oktapreview.com/oauth2/default`.

* Example:

ISSUER=https://yourOktaDomain.com/oauth2/default

CLIENT\_ID=123xxxxx123

Now start the app server:

```

ng serve

```

Now navigate to http://localhost:8080 in your browser.

If you see a home page that prompts you to login, then things are working!  Clicking the **\*\*Log in\*\*** button will redirect you to the Okta hosted sign-in page.

You can login with the same account that you created when signing up for your Developer Org, or you can use a known username and password from your Okta Directory.

**\*\*Note:\*\*** If you are currently using your Developer Console, you already have a Single Sign-On (SSO) session for your Org.  You will be automatically logged into your application as the same user that is using the Developer Console.  You may want to use an incognito tab to test the flow from a blank slate.

**## Integrating The Resource Server**

If you were able to successfully login in the previous section you can continue with the resource server example.  Please download and run one of these sample applications in another terminal:

\* [Node/Express Resource Server Example](https://github.com/okta/samples-nodejs-express-4/tree/master/resource-server)

\* [Java/Spring MVC Resource Server Example](https://github.com/okta/samples-java-spring/tree/master/resource-server)

\* [ASP.NET](https://github.com/okta/samples-aspnet/tree/master/resource-server) and [ASP.NET Core](https://github.com/okta/samples-aspnetcore/tree/master/samples-aspnetcore-3x/resource-server) Resource Server Examples

Once you have the resource server running (it will run on port 8000) you can visit the `/messages` page within the Angular application to see the authentication flow.  The Angular application will use its stored access token to authenticate itself with the resource server, you will see this as the `Authorization: Bearer <access\_token>` header on the request if you inspect the network traffic in the browser.

[Angular CLI]: https://cli.angular.io/

[Okta Angular Library]: https://github.com/okta/okta-oidc-js/tree/master/packages/okta-angular

[PKCE Flow]: https://developer.okta.com/docs/guides/implement-auth-code-pkce

[OIDC SPA Setup Instructions]: https://developer.okta.com/docs/guides/sign-into-spa/angular/before-you-begin

[Okta Sign In Widget]: https://github.com/okta/okta-signin-widget

**Result:**

* As a result when including client id and issuer url in ”app.component.ts” file authentication is reflecting with 400 Error page of OKTA. There while clicking “Go to Homepage” option, OKTA sign-in widget will appear.