**Angular**

**SSO Authentication**

**LOGIN SSO IN ANGULAR**

**Reference Link:** <https://learn.microsoft.com/en-us/azure/active-directory/develop/tutorial-v2-angular-auth-code>

Install in angular for single sign login:

npm install @azure/msal-browser @azure/msal-angular

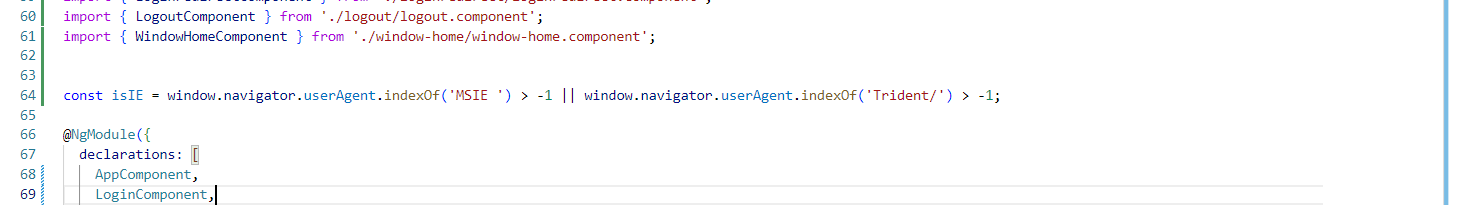
# Above will Install MSAL Browser and MSAL Angular in your application

**1. App-module.ts**

* Add the below content in App.module.ts page after the import link

|  |
| --- |
| const isIE = window.navigator.userAgent.indexOf('MSIE ') > -1 || window.navigator.userAgent.indexOf('Trident/') > -1; |

**Refer Screen shot:**



* Add the below content in app.module page inside the import

Below Application client details will be provided by NOC team.

For DEV :

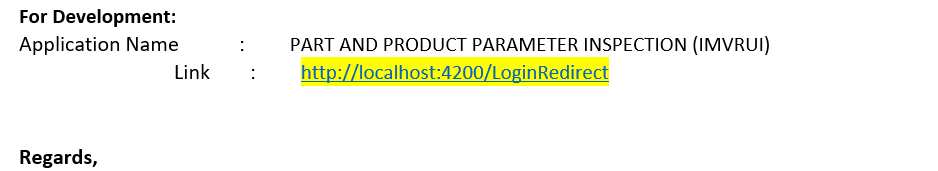
Application ID : d20987a2-ef83-4db7-9563-e68090a43ff8 ---add application id to client id

S.value : tit8Q~1bbmNYXN7elGKUq~7vLtCGicAnrndKJaWg

S.ID : adb709ab-c4da-4120-a6d4-a0e85754acd4

Object ID : 3c207a21-9d04-4e32-88cb-0654eef9112a

Tenant ID : d30feff3-78f9-476a-81e4-c71b80743988 --- add Tenant ID to the authority



For Prod :

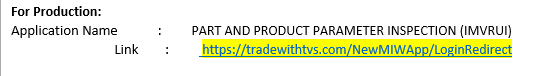
Application ID : 8e642b69-4772-40f8-b92e-e2f025d4e8d5 ---add application id to client id

S.Value : MMJ8Q~X3s6sh50VN~marU0fBt~fbyXmI6fUDccKm

S.ID : a30f0d7b-ec96-4240-a9db-58625ee64845

Object ID :a339f96c-726a-4aaf-8d11-20cb0309e551

Tenant ID :d30feff3-78f9-476a-81e4-c71b80743988 --- add Tenant ID to the authority



**Refer below format:**

**imports: [**

 MsalModule.forRoot(new PublicClientApplication({

      auth: {

        //--Development

        clientId: 'd20987a2-ef83-4db7-9563-e68090a43ff8', //DVS

        authority: 'https://login.microsoftonline.com/d30feff3-78f9-476a-81e4-c71b80743988',

redirectUri: 'http://localhost:4200/LoginRedirect', //DVS

    postLogoutRedirectUri: 'http://localhost:4200/Logout'  //DVS

       // --Production

      //  clientId: '8e642b69-4772-40f8-b92e-e2f025d4e8d5', //PRD

      //  authority: 'https://login.microsoftonline.com/d30feff3-78f9-476a-81e4-c71b80743988',  //PRD

      //  redirectUri: ' https://tradewithtvs.com/NewMIWApp/LoginRedirect',  //PRD

      //  postLogoutRedirectUri: 'https://tradewithtvs.com/NewMIWApp/Logout'  //PRD

},

      cache:

      {

        cacheLocation: 'localStorage',

        storeAuthStateInCookie: isIE,

      }

    }),

      {

        interactionType: InteractionType.Redirect,

        authRequest: { scopes: ['user.read'] }

      },

      {

        interactionType: InteractionType.Redirect,

        // MSAL Interceptor Configuration

        protectedResourceMap: new Map([['https://graph.microsoft.com/v1.0/me', ['user.read']]])

      }),

**],**

  providers: [{ provide: HTTP\_INTERCEPTORS, useClass: MsalInterceptor, multi: true }, MsalGuard],

  bootstrap: [AppComponent, MsalRedirectComponent]

})

**2. app-routing.module.ts**

* Add the below content in app-routing.module.ts page below the import

const isIframe = window !== window.parent && !window.opener;

**Refer Screen shot:**



const routes: Routes = [ {    path:'LoginRedirect',    component:LoginredirectComponent  },  {    path:'WindowHome',    component:WindowHomeComponent  },  {    path:'WindowLogout',   component:LogoutComponent   }, ];

@NgModule({

  imports: [RouterModule.forRoot(routes, {

    // Don't perform initial navigation in iframes or popups

   initialNavigation: !BrowserUtils.isInIframe() && !BrowserUtils.isInPopup() ? 'enabledNonBlocking' : 'disabled'

// Set to enabledBlocking to use Angular Universal

  })],

**3. app.component.ts**

* Add the below code in app.component.ts page

import { MsalBroadcastService, MsalService } from '@azure/msal-angular';

import { InteractionStatus } from '@azure/msal-browser';

import { filter, Subject, takeUntil } from 'rxjs';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'IMVRApp';

  isIframe = false;

  loginDisplay = false;

  private readonly \_destroying$ = new Subject<void>();

  constructor(private router:Router,private broadcastService: MsalBroadcastService, private authService: MsalService){}

  ngOnInit()

  {

    this.isIframe = window !== window.parent && !window.opener;

    this.broadcastService.inProgress$

    .pipe(

      filter((status: InteractionStatus) => status === InteractionStatus.None),

      takeUntil(this.\_destroying$)

    )

    .subscribe(() => {

      this.setLoginDisplay();

    })  }

   login() {

   this.authService.loginRedirect();

  }

  setLoginDisplay() {

    this.loginDisplay = this.authService.instance.getAllAccounts().length > 0;

  }

  ngOnDestroy(): void {

    this.\_destroying$.next(undefined);

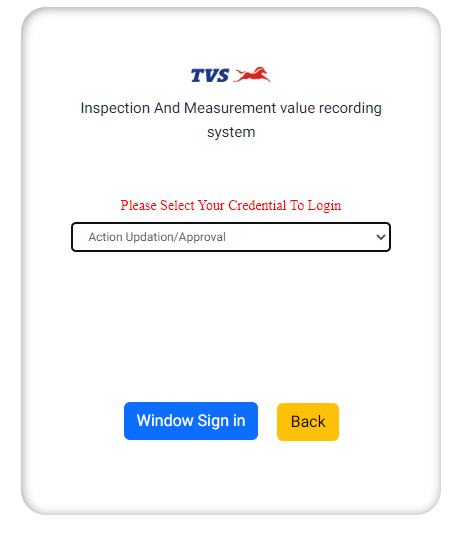
    this.\_destroying$.complete();

  }

 }

}

**4. Login.component.ts**



After clicking Window Sign In button, it will redirect to login Redirect Component

 Window() {

    this.router.navigate(['LoginRedirect'])

  }

**5. Loginredirect.component.ts**

Create a new loginredirect component

LoginRedirect ts component

import { HttpClient } from '@angular/common/http';

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute, Router } from '@angular/router';

import { MsalBroadcastService, MsalService } from '@azure/msal-angular';

import { EventMessage, EventType, InteractionStatus } from '@azure/msal-browser';

import { filter, Subject, takeUntil } from 'rxjs';

import { IMVRAPPService } from '../imvrapp.service';

const GRAPH\_ENDPOINT = '<https://graph.microsoft.com/v1.0/me>';

type ProfileType = { givenName?: string, surname?: string, userPrincipalName?: string, id?: string };

@Component({

  selector: 'app-loginredirect',

  templateUrl: './loginredirect.component.html',

  styleUrls: ['./loginredirect.component.css']

})

export class LoginredirectComponent implements OnInit {

  isIframe = false;

  private readonly \_destroying$ = new Subject<void>();

  profile!: ProfileType;

  loginDisplay = false;

  flagnumchk: any = [];

  constructor(private http: HttpClient, private authService: MsalService,

    private msalBroadcastService: MsalBroadcastService,private imvr\_service: IMVRAPPService,

    private router: Router,  private route: ActivatedRoute,) { }

  ngOnInit() {

    this.msalBroadcastService.msalSubject$.pipe(filter((msg: EventMessage) => msg.eventType === EventType.LOGIN\_SUCCESS),).subscribe(

      (result: EventMessage) => { console.log(result); });

      console.log('testing')

      this.msalBroadcastService.inProgress$.pipe(filter((status: InteractionStatus) => status === InteractionStatus.None)).subscribe(() => {

         this.setLoginDisplay();

         this.getProfile();

         })

         this.router.navigate(['WindowHome'])

  }

setLoginDisplay() {

this.loginDisplay = this.authService.instance.getAllAccounts().length > 0; }

getProfile() {

    this.http.get(GRAPH\_ENDPOINT).subscribe(profile => {

      this.profile = profile;

      console.log('profile', this.profile)

     let option = { Email: this.profile.userPrincipalName}

     console.log('profiledata',this.profile.userPrincipalName)

//Passing employee email id as input to fetch employee details

     this.imvr\_service.getempno(this.profile.userPrincipalName).subscribe(

      data => {

      console.log('empno', data)

      let ref = JSON.parse(data.toString())

      sessionStorage.setItem("WindowEmpno", ref[0].empno);

      sessionStorage.setItem("WindowEmpname",ref[0].empname);

     })

    });

  }

}

**6. Logout.component.ts**

import { Component, OnInit, OnDestroy, Inject } from '@angular/core';

import { MsalService, MsalBroadcastService, MSAL\_GUARD\_CONFIG, MsalGuardConfiguration } from '@azure/msal-angular';

import { InteractionStatus, PopupRequest, RedirectRequest } from '@azure/msal-browser';

import { Subject } from 'rxjs';

import { filter, takeUntil } from 'rxjs/operators';

@Component({

  selector: 'app-logout',

  templateUrl: './logout.component.html',

  styleUrls: ['./logout.component.css']

})

export class LogoutComponent implements OnInit {

  title = 'msal-angular-tutorial';

  isIframe = false;

  loginDisplay = false;

  private readonly \_destroying$ = new Subject<void>();

  constructor(@Inject(MSAL\_GUARD\_CONFIG) private msalGuardConfig: MsalGuardConfiguration, private broadcastService: MsalBroadcastService, private authService: MsalService) { }

  ngOnInit(): void {

    this.isIframe = window !== window.parent && !window.opener;

    this.broadcastService.inProgress$

    .pipe(

      filter((status: InteractionStatus) => status === InteractionStatus.None),

      takeUntil(this.\_destroying$)

    )

    .subscribe(() => {

      this.setLoginDisplay();

    })

   this.logout();

  }

  login() {

    if (this.msalGuardConfig.authRequest){

      this.authService.loginRedirect({...this.msalGuardConfig.authRequest} as RedirectRequest);

    } else {

      this.authService.loginRedirect();

    }

  }

  logout() { // Add log out function here

    this.authService.logoutRedirect({

      postLogoutRedirectUri: '/login'

    });

  }

  setLoginDisplay() {

    this.loginDisplay = this.authService.instance.getAllAccounts().length > 0;

  }

  ngOnDestroy(): void {

    this.\_destroying$.next(undefined);

    this.\_destroying$.complete();

  }

}

