

GLOBETEAM



Danmarks Miljøportal (DMP)

Vejledning til fagsystemejere
omkring tilkobling af Angular web
applikation via OAuth 2.0 OpenID
Connect

Version 1.0

Indledning

Denne vejledning beskriver hvordan man claims-enabler en Angular web applikation til brug med DMP's føderale brugerstyringsløsning ved brug af OAuth 2.0 OpenID Connect profil.

Vejledningen skal læses i konteksten af den overordnede vejledning "Vejledning til fagsystemejere omkring forløbet for tilkobling af en applikation".

Vejledningen forudsætter at man anvender Angular som basic for sin Web applikation. Vejledningen vil give et Steb-by-Step på hvordan man ved brug af Visual Studio Code kan oprette en føderationsforbindelse til et Angular web site.

Indhold

Indledning	2
Indhold	3
Tilkoblingen af en Angular web applikation	4
Sådan kobles Angular web applikationen op mod DMP	5

Tilkoblingen af en Angular web applikation

Sikkerheden for Web Applikationer der bruger WS-Federation passive på DMPs Safewhere*Identify STS-løsning er baseret på følgende standarder:

- OAuth 2.0 – OpenID Connect (OIDC) profil

End points for DMP's STS er angivet nedenfor:

Passive log-in via WS-Federation passive

- <https://log-in.miljoportal.dk/runtime/> (produktionsmiljø)
- <https://log-in.test.miljoportal.dk/runtime/> (testmiljø)

Bemærk at samples'ne som udgangspunkt er bygget til brug med test miljøet. De vil dog naturligvis kunne tilpasses til at bruge produktionsystemet blot ved at udskifte URL'erne på end points'ne.

Det anbefales så vidt muligt, at udviklerne tager udgangspunkt i de medfølgende samples, når der oprettes nye web applikationer. De mange opsætningsmuligheder vil ellers hurtigt kunne risikere at gøre det til en uoverskuelig opgave at konfigurere det ovenstående i hånden.

Sådan kobles Angular web applikationen op mod DMP

1. Installer Node JS med standard indstillinger for alt: <https://nodejs.org/en/download/>
2. Download og installer Visual Studio Code: <https://code.visualstudio.com/download>
3. Åben en PowerShell i admin mode. Installer Angular CLI:

```
npm install -g @angular/cli
```

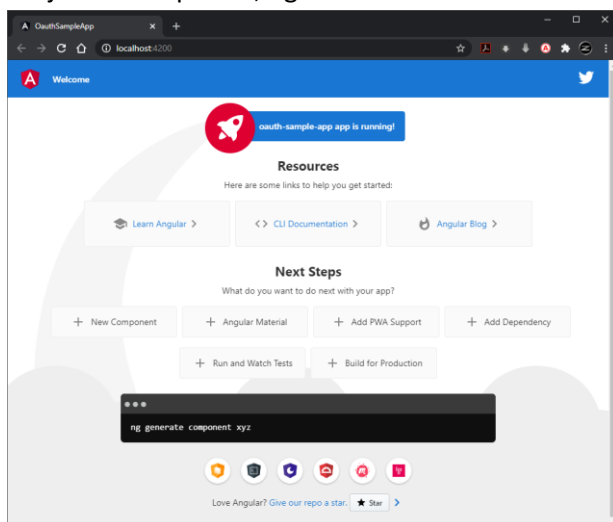
4. Navigér til det bibliotek hvor applikationen skal oprettes – feks. c:/Projects/

```
ng new oauth-sample-app
```

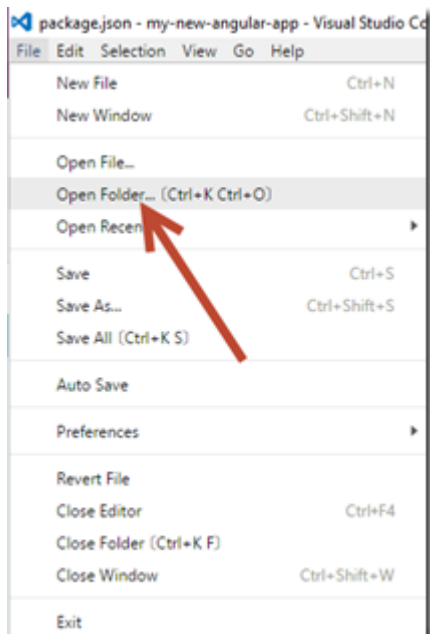
5. Navigér ind i projektet 'cd oauth-sample-app'
6. Verificér at applikationen kan køre

```
ng serve
```

7. Projektet kompilér, og du kan derefter verificere at sitet virker på <http://localhost/4200>



8. Åben Visual Studio Code
9. Klik på Open Folder og vælg det oprettede projekt 'c:/Projects/oauth-sample-app':



10. Åben Terminal og indtast følgende:

```
npm install angular-auth-oidc-client
```

11. Opret komponenterne 'Home' og 'Unauthorized' ved at køre følgende kommandoer i Terminal:

```
ng generate component home  
ng generate component Unauthorized
```

12. Indsæt følgende kodestump i app.module.ts:

```
import { APP_INITIALIZER, NgModule } from '@angular/core';  
import { BrowserModule } from '@angular/platform-browser';  
import { RouterModule } from '@angular/router';  
import { filter } from 'rxjs/operators';  
import { AppComponent } from './app.component';  
import { HomeComponent } from './home/home.component';  
import { UnauthorizedComponent } from './unauthorized/unauthorized.component';  
import { AuthModule, EventTypes, LogLevel, OidcConfigService, PublicEventsService }  
  from 'angular-auth-oidc-client';  
export function configureAuth(oidcConfigService: OidcConfigService) {  
  return () =>  
    oidcConfigService.withConfig({  
      stsServer: 'https://log-in.test.miljoeportal.dk/runtime/oauth2',  
      redirectUrl: window.location.origin,  
      postLogoutRedirectUri: window.location.origin + '/unauthorized/',  
      clientId: 'oauth-sampleapp-localhost',
```

```

    scope: 'openid offline_access http://www.miljoeportal.dk/roles',
    responseType: 'code',
    silentRenew: true,
    useRefreshToken: true,
    logLevel: LogLevel.Debug,
  });
}

```

13. Erstat @NgModule declarations delen med følgende kode:

```

@NgModule({
  declarations: [AppComponent, HomeComponent, UnauthorizedComponent],
  imports: [
    BrowserModule,
    RouterModule.forRoot([
      { path: '', redirectTo: 'home', pathMatch: 'full' },
      { path: 'home', component: HomeComponent },
      { path: 'forbidden', component: UnauthorizedComponent },
      { path: 'unauthorized', component: UnauthorizedComponent },
    ]),
    AuthModule.forRoot(),
  ],
  providers: [
    OidcConfigService,
    {
      provide: APP_INITIALIZER,
      useFactory: configureAuth,
      deps: [OidcConfigService],
      multi: true,
    },
  ],
  bootstrap: [AppComponent],
})

```

14. Erstat AppModule sektionen med følgende:

```

export class AppModule {
  constructor(private readonly eventService: PublicEventsService) {
    this.eventService
      .registerForEvents()
      .pipe(filter((notification) => notification.type === EventTypes.ConfigLoaded))
      .subscribe((config) => {
        console.log('ConfigLoaded', config);
      });
  }
}

```

```
}  
}
```

15. Åben filen app.component.html og overskriv den med følgende:

```
<h2>Sample Code Flow with refresh tokens</h2>  
<p>  
  <button *ngIf="isAnonymous" #loginButton (click)="login()">Login</button>  
  <button *ngIf="isAuthenticated" #logoutButton (click)="logout()">Log off</button>  
</p>  
<div *ngIf="isAuthenticated">  
  <span>User Data:</span>  
  <textarea class="text-area" [innerHTML]="user_data"></textarea>  
</div>  
<div *ngIf="isAuthenticated">  
  <span>Access Token:</span>  
  <textarea class="text-area" [innerHTML]="token"></textarea>  
</div>  
<router-outlet></router-outlet>
```

16. Åben filen app.component.ts og overskriv den med følgende:

```
import { Component, OnInit } from '@angular/core';  
import { OidcSecurityService } from 'angular-auth-oidc-client';  
@Component({  
  selector: 'app-root',  
  templateUrl: 'app.component.html',  
})  
export class AppComponent implements OnInit {  
  public isAnonymous = false;  
  public isAuthenticated = false;  
  public token = null;  
  public user_data = null;  
  
  constructor(public oidcSecurityService: OidcSecurityService) {}  
  
  ngOnInit() {  
    this.oidcSecurityService.checkAuth().subscribe((isAuthenticated) =>  
      {  
        console.log('app authenticated', isAuthenticated);  
        this.isAnonymous = !isAuthenticated;  
        this.isAuthenticated = isAuthenticated;  
        if(isAuthenticated)  
        {  
          this.token = this.oidcSecurityService.getToken();  
        }  
      }  
    );  
  }  
}
```



```

        this.user_data = JSON.stringify(this.oidcSecurityService.userData$.source["_value"], null, 1);
    }
}
);
}

public login(): void {
    this.oidcSecurityService.authorize();
}

public logoff(): void {
    this.oidcSecurityService.logoff();
}
}

```

17. Åben filen styles.css og erstat dens indhold med:

```

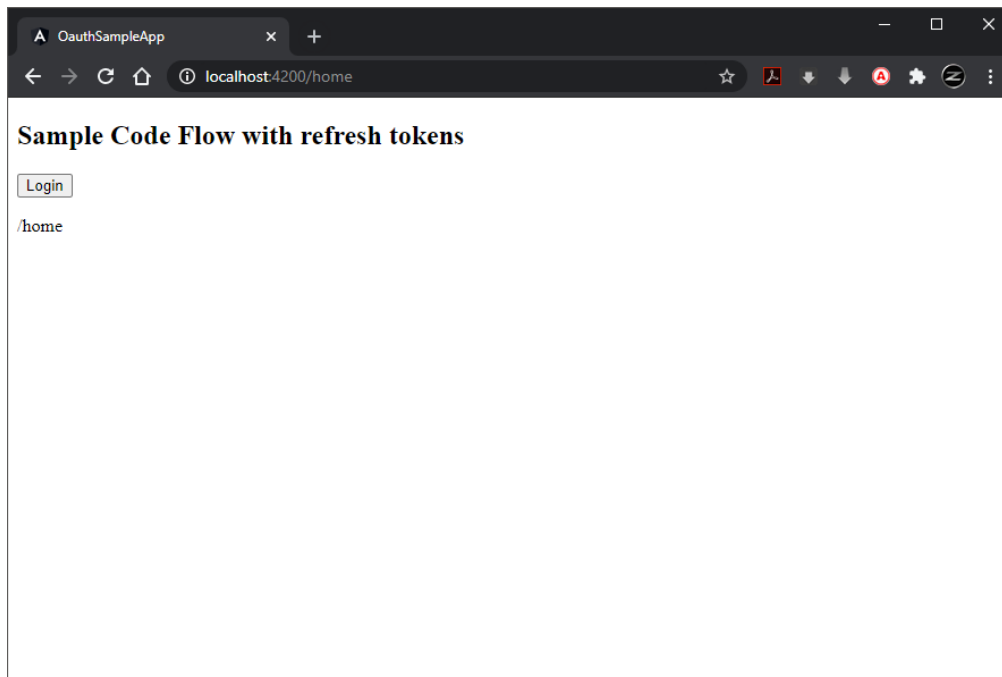
/* You can add global styles to this file, and also import other style files */
.text-area{
    width:100%;
    height:300px;
}

```

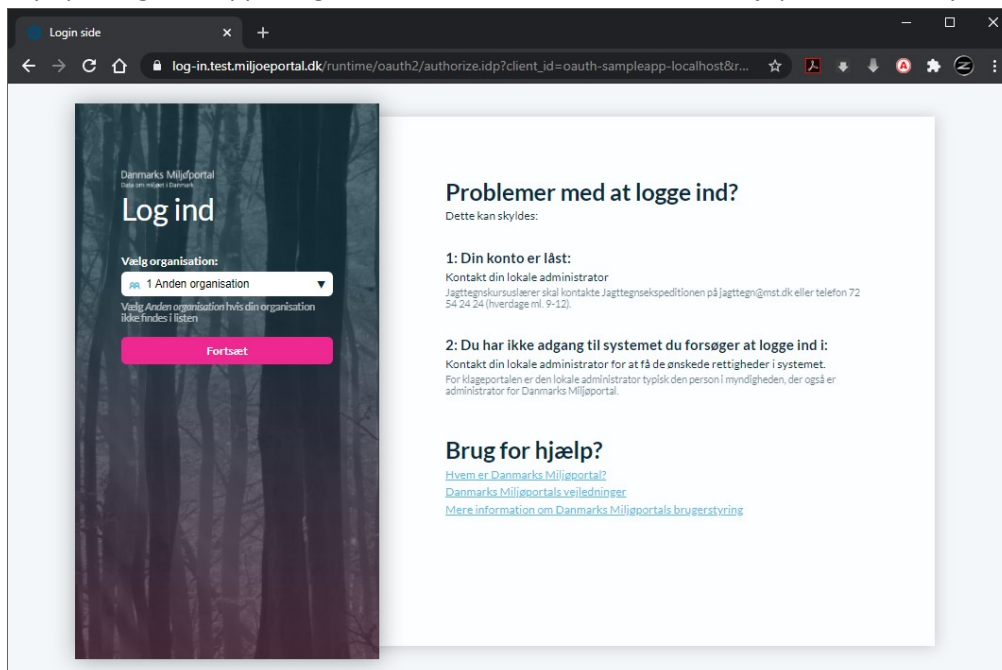
18. Kør sitet hvis det ikke allerede kører ved at gå eksekvere følgende i Terminal:

```
ng serve
```

19. Du vil nu få vist et mere eller mindre tomt site med en enkelt 'Login' knap.

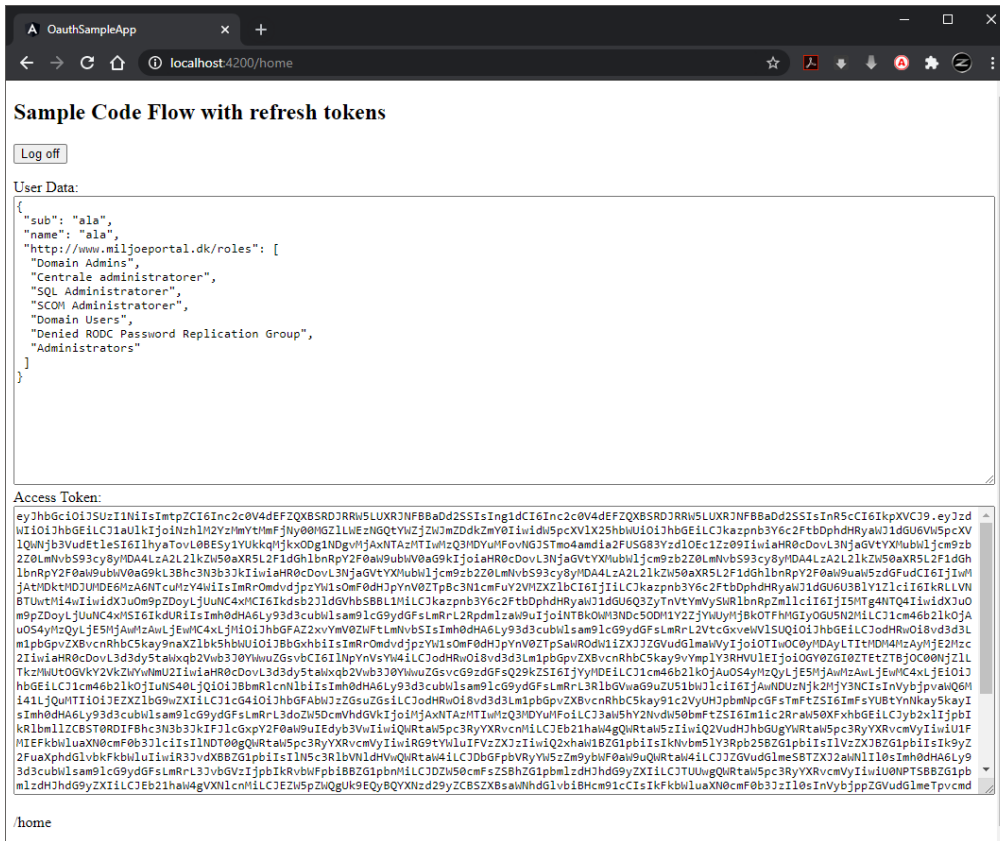


20. Tryk på 'Login' knappen og du vil blive ført over i Danmarks Miljøportals Identity Provider

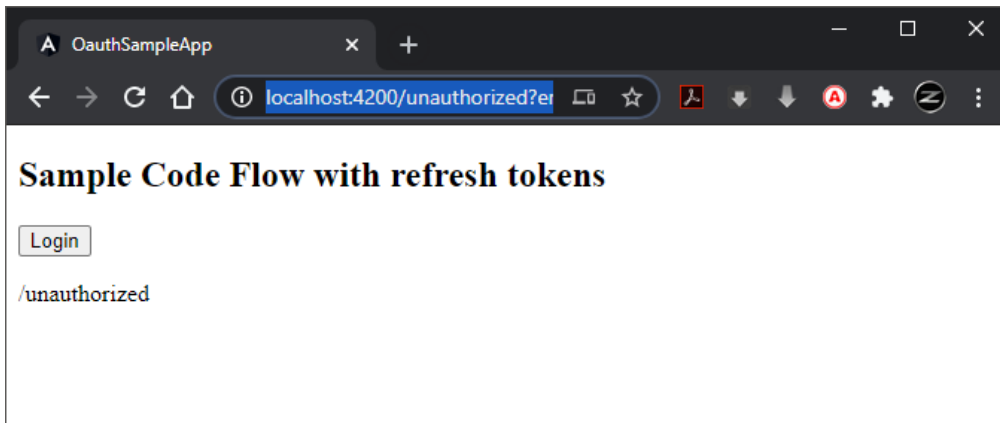


21. Log på med din DMP bruger eller vælg NemLog-in i organisations vælgeren.

22. Efter login vil du blive ført tilbage til dit projekt site som angiver de oplysninger der findes i OIDC login'et – og en mulighed for at logge ud igen:.



23. Ved tryk på Log off – havner du på denne side efter et genbesøg med 'log ud forespørgslen' på Danmarks Miljøportals Identity Provider:



Det AccessToken som udstedes ved identifikations kaldet, kan bruge til at kalde de backend services som godtager denne slags tokens. Har et fagsystem et "Swagger" UI interface, er det med garanti muligt at benytte AccessToken'et til at foretage kald mod dette API.

```
setHeaders: {
  Authorization: 'Bearer ${this.auth.getToken()}`
}
```