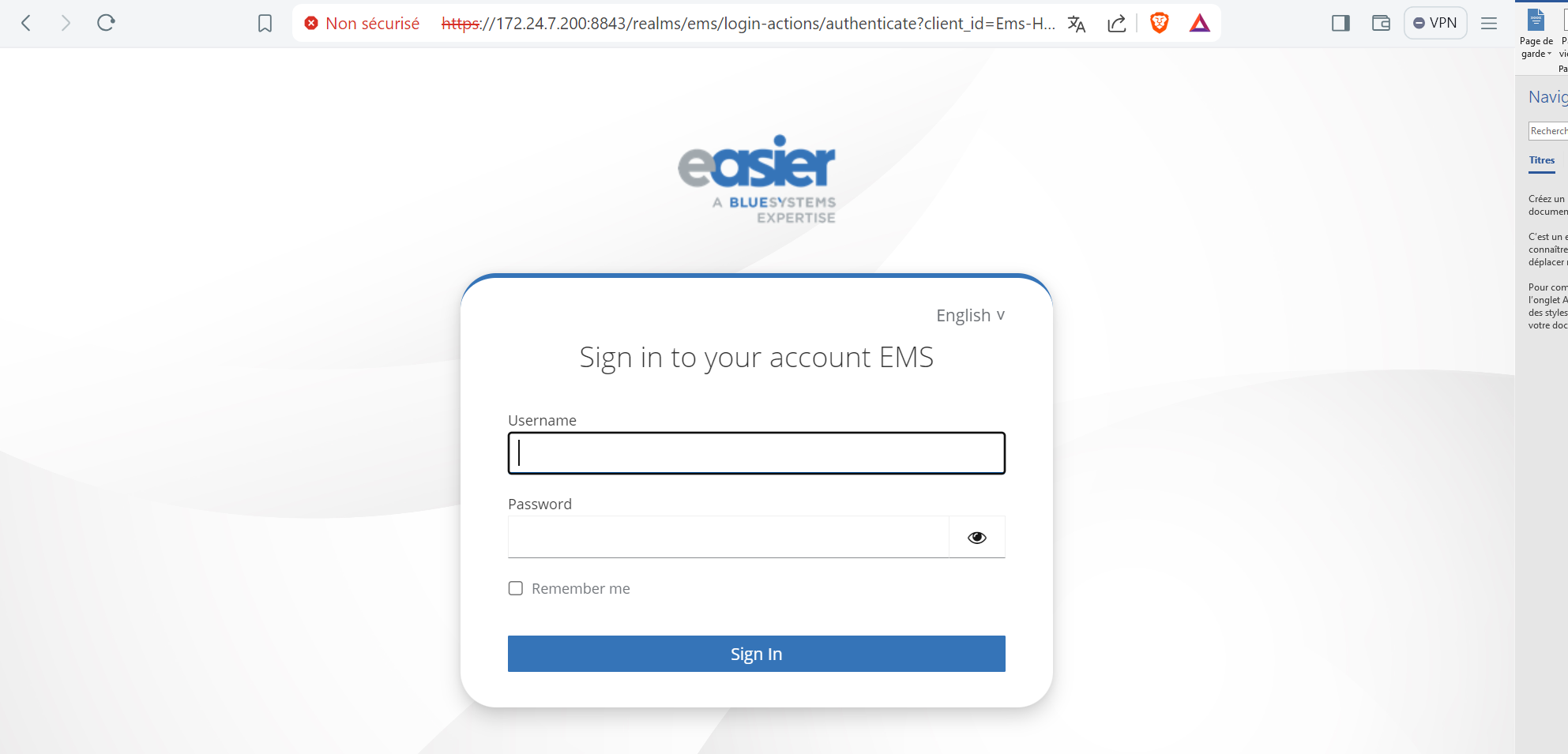
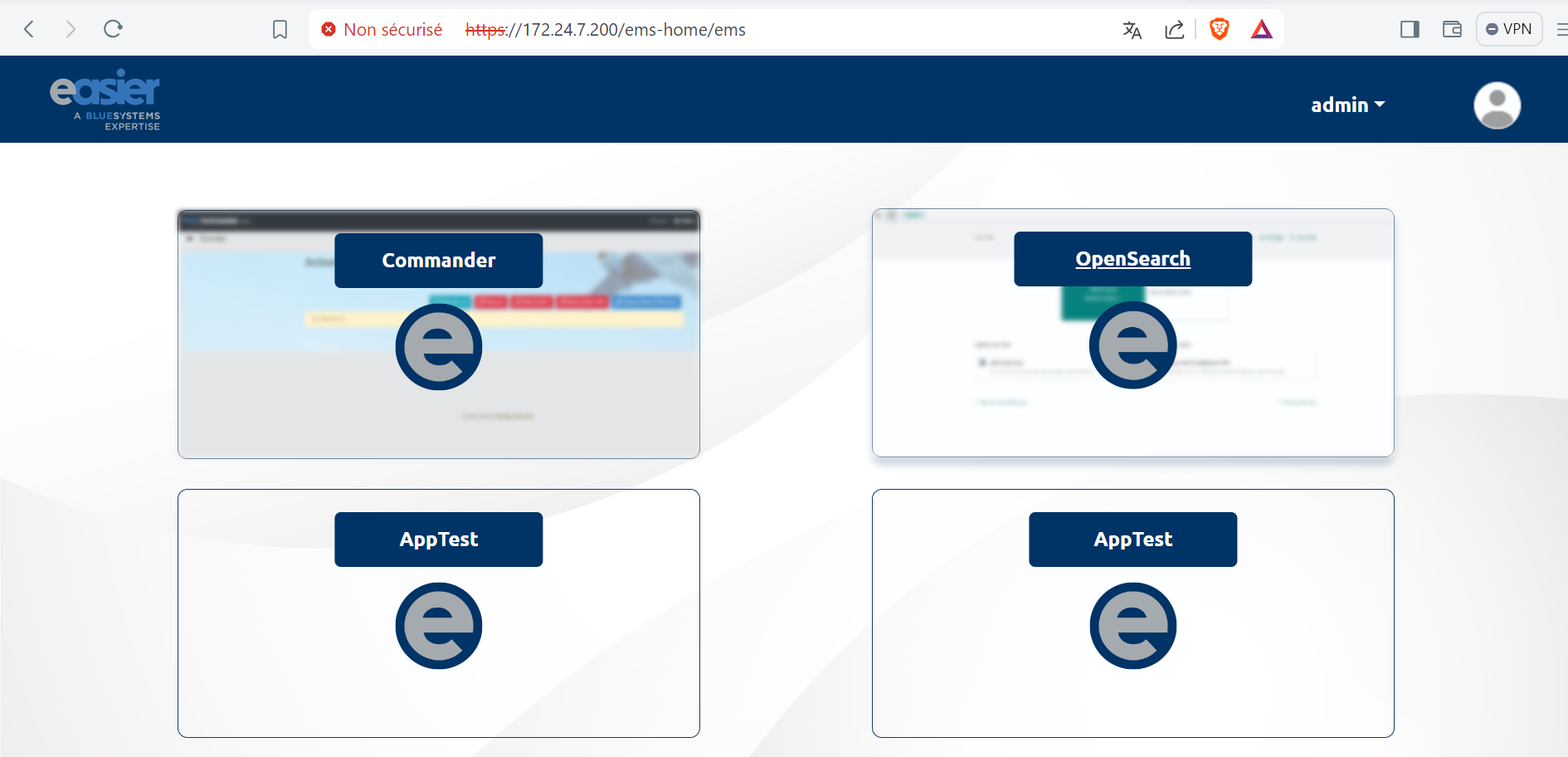
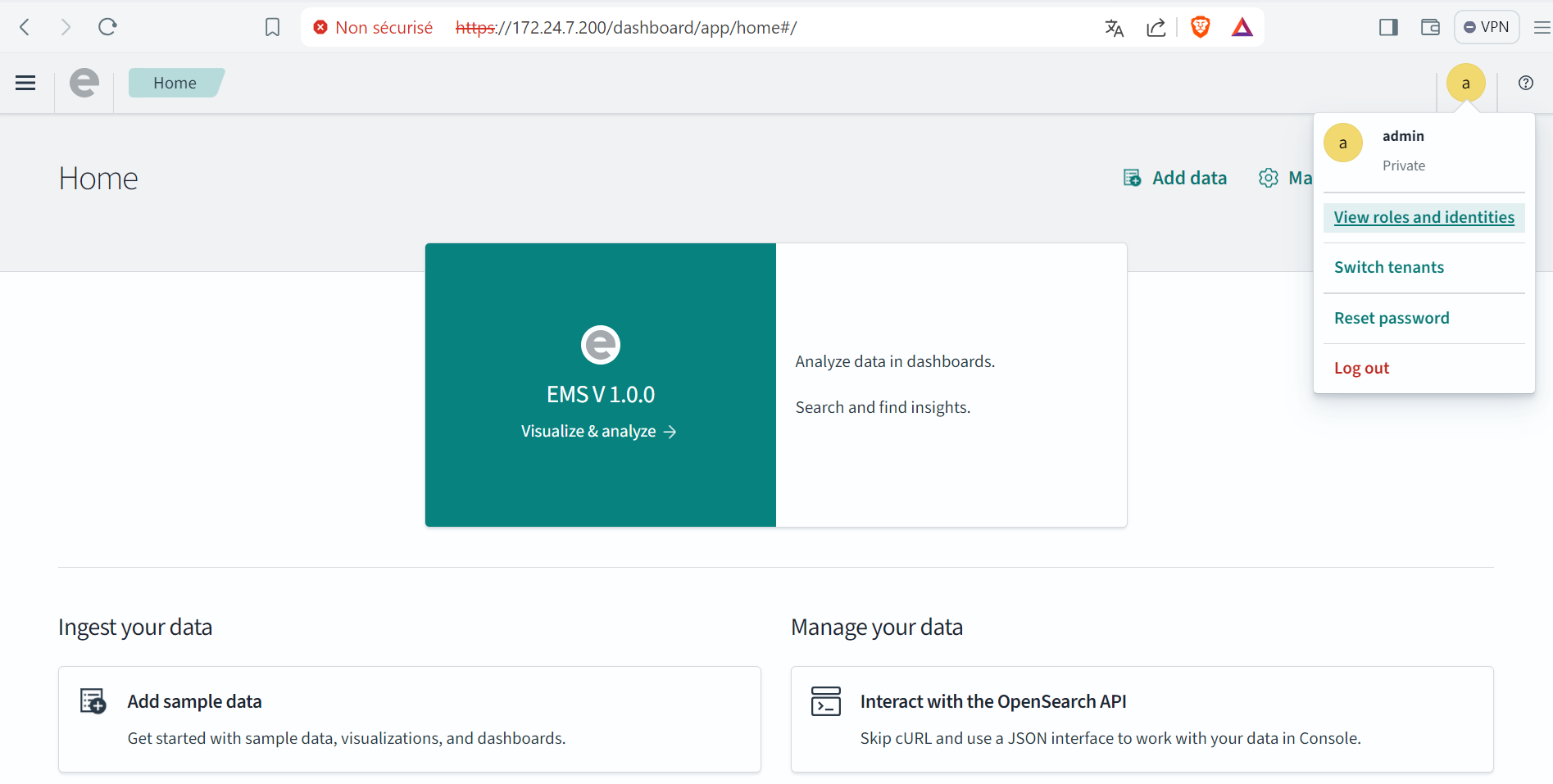
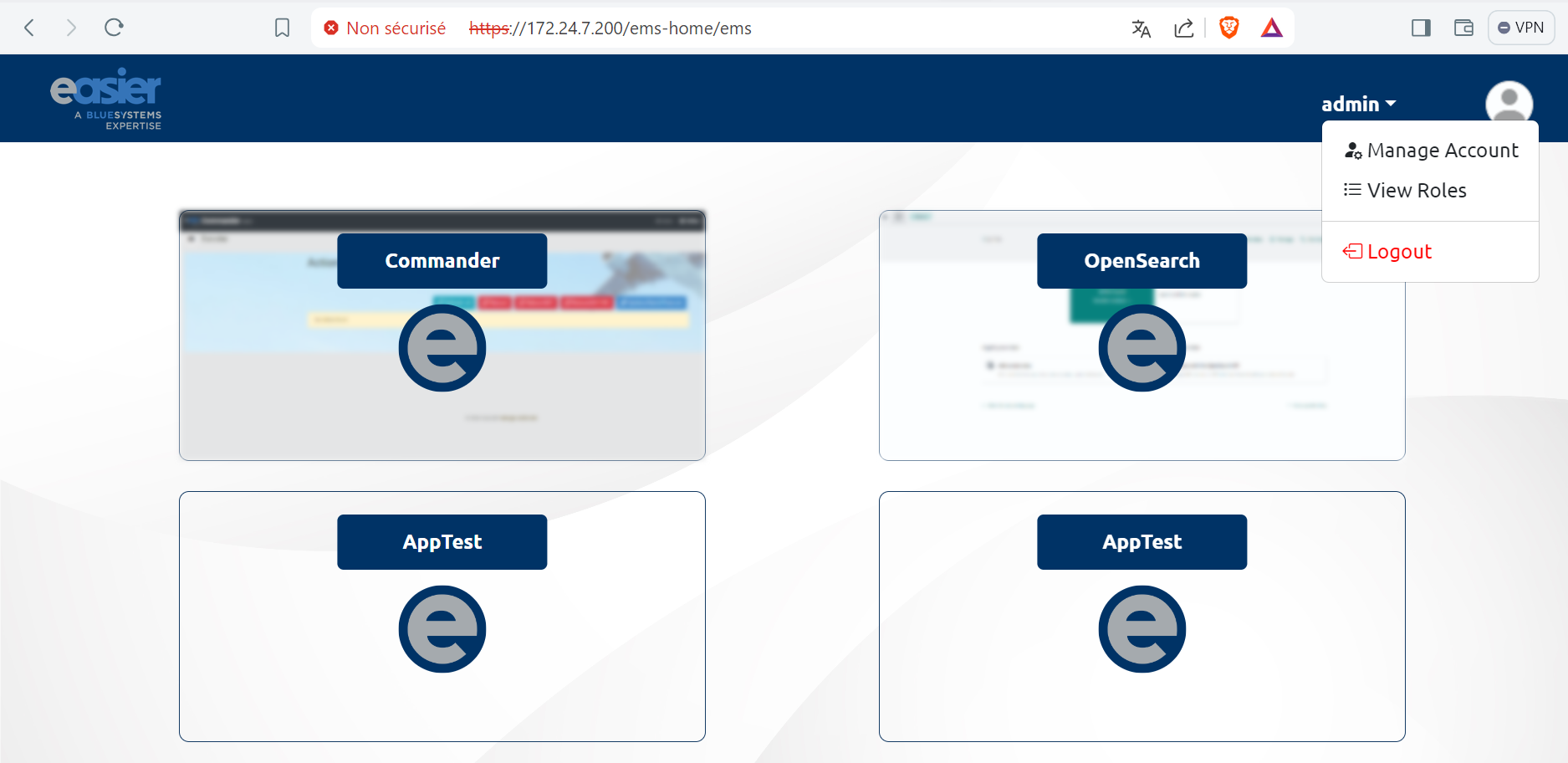
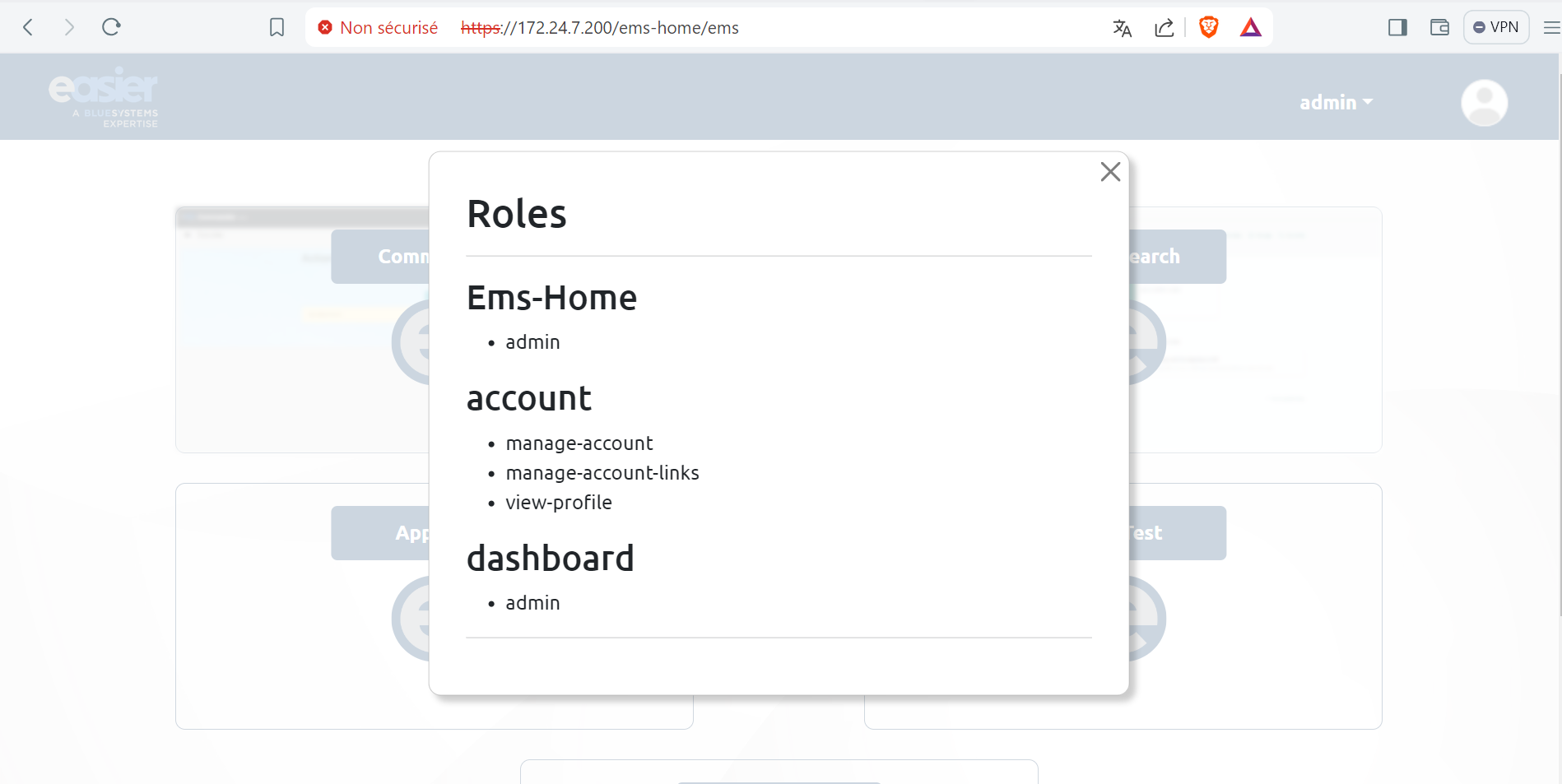
**Comment ça marche ?**

1. L’application ems-home accessible sur https :172.24.7.200/ems-home
2. on y’accèdent on est invité a ce connecter avec un compte EMS via keycloak
3. Apres une connexion réussie on est redirigé vers la page d’accueil de l’application ems-home ou on retrouve les différents application disponible (commander, opensearch,…selon notre configuration) :
4. On clickent sur commander on est redirigé vers commander sans besoin de se reconnceter une autre fois :

Image

1. On clickent sur opensearch on est redirige vers opensearch sans besoin de se reconnecter une autre fois :
2. Sur la nav bar ya un menu deroulant contenant les option suivante : manage account,veiw roles et logout :
3. On peut voir les rôles de l’utilisateur courant sur les différentes applications disponible comme suit :
4. Et pour se deconnecter de tout les app il suffit juste de se deconnceter de ems-home et tout les autre application seront déconnecter automatiquement.

**Comment l’application est configurer ?**

l’application ems-home est developper avec angular CLI contenant plusieur composant et services permettant le bon foctionnement de l’application.

Parmis tout les service existant le service keycloak le plus important pour permettre la securité de l’application et le bon fonctionnement de single-sign-on :  
  
voici la fonction d’initialisation de keycloak sur le KeycloakService,

init() {  
 this.configService.getConfig().subscribe((data: any) => {  
 this.config = data.keycloak;  
 *console*.log("config:", this.config.clientId);  
  
 this.\_keycloak = new Keycloak({  
 url: this.config.url,  
 realm: this.config.realm,  
 clientId: this.config.clientId,  
 });  
  
 *console*.log('Keycloak initialization started', this.config);  
 this.loaderService.show();  
  
 this.\_keycloak.init({  
 onLoad: 'login-required',  
 }).then(async authenticated => {  
 if (authenticated) {  
 *console*.log('User authenticated successfully');  
 await this.loadProfile();  
 this.\_details = this.\_keycloak?.resourceAccess;  
 *console*.log('details:', this.\_details);  
 } else {  
 *console*.warn('User is not authenticated');  
 // Gérer le cas où l'utilisateur n'est pas authentifié  
 }  
 this.\_keycloakInitialized.next(true);  
 }).catch(initError => {  
 this.\_keycloakInitialized.next(true);  
 *console*.error('Keycloak initialization failed:', initError);  
 this.router.navigate(['/errorkeycloakinit']);  
 }).finally(() => {  
 *console*.log("Finalizing Keycloak initialization");  
 this.loaderService.hide();  
 this.\_keycloakInitialized.next(true);  
 });  
 });  
}

qui est en suite appellez dans le fichier app.config.ts pour s’appliquer a tout l’application ce qui fait que chaque access a l’application necessite une authentification de service keycloak.

import {*APP\_INITIALIZER*, ApplicationConfig, importProvidersFrom} from '@angular/core';  
import { provideRouter } from '@angular/router';  
  
import { *routes* } from './app.routes';  
import {HttpClientJsonpModule, HttpClientModule} from "@angular/common/http";  
import {KeycloakService} from "./services/keycloak/keycloak.service";  
import {ConfigService} from "./services/config/config.service";  
  
export function kcFactory(kcService : KeycloakService){  
 return ()=> kcService.init();  
}  
  
  
  
export const *appConfig*: ApplicationConfig = {  
 providers: [  
 // ConfigService  
 ConfigService,  
 // KeycloakService  
 KeycloakService,  
 {  
 provide: *APP\_INITIALIZER*,  
 useFactory: kcFactory,  
 deps: [KeycloakService],  
 multi: true  
 },  
 // Other providers  
 provideRouter(*routes*),  
 importProvidersFrom(HttpClientModule),  
 importProvidersFrom(HttpClientJsonpModule),  
 ]  
};

lors de l’appelle au service keycloak le client est redirige automatiquement vers l’url de keycloak specifie avec le realm et le client id déjà crée sur keycloak lui-même comme suit :

step1: creation d'un client avec des config specifique pour l'application ems-home:

![alt text](image.png)

step 2: on ajoute deux roles pour le client ems-home comme suit:

![alt text](image-1.png)

step 3: pour appliquer un theme personalisé il faut changé la config suivant sur le realm ems:

![alt text](image-3.png)

step 4: pour avoir une interface login multilangues il suffit juste d'ajouter la langue qu'on souhaite sur le realm ems comme suit:

![alt text](image-4.png)

Important : tout l’application repose sur un fichier de config nommé config.json ou tout les configuration de demarage, url et application sont montionné .