

## Integración con PAEC desde una aplicación Angular

### Import de librerías

Para securizar la app, se usa la librería `@ebondu/angular2-keycloak`. Después de agregarla, se importa el módulo `Ng2KeycloakModule` con la función `forRoot()` en la app principal:

```
import { Ng2KeycloakModule } from '@ebondu/angular2-keycloak';  
  
...  
  
@NgModule({  
  
  imports: [  
  
    Ng2KeycloakModule.forRoot()  
  
  ],  
  
})
```

También se requiere importar el `HttpModule` de `@angular/http`:

```
import { HttpModule } from '@angular/http';  
  
...  
  
@NgModule({  
  
  imports: [  
  
    HttpModule()  
  
  ],  
  
})
```

## Archivo keycloak.json

La app necesita saber de dónde obtener los datos del servidor PAEC. Esto sirve para obtener datos del usuario logueado, y para consultar web services securizados. Para aportar estos datos, se agrega el archivo `keycloak.json` a los assets de un despliegue Angular.

Los assets son todos los archivos que no son de código (datos JSON, imágenes, estilos, etc.), y que se exponen en una URL.

## Usuario autenticado y datos

Para saber si hay un usuario autenticado, basta con ver si `Keycloak.accessToken` tiene valor. La librería provee un campo `Keycloak.authenticatedBehaviourSubject`, al que se pueden suscribir funciones. Cuando la función suscripta también valida que `Keycloak.accessToken` tiene valor, se puede considerar al usuario autenticado. Se asigna a una variable booleana para mostrar la interfaz solamente en este caso. Los roles del usuario se pueden sacar de `Keycloak.tokenParsed`.

Una vez autenticado el usuario, se puede obtener sus datos consultando a Keycloak, con `Keycloak.loadUserProfile()`. Esta función también se resuelve asincrónicamente (ya que va al servidor), y al terminar carga el valor `Keycloak.idTokenParsed`. De este token se puede sacar la información del usuario.