Authentication and Access Control

1 Introduction

Purpose:

Authentication and Access Control is important to ensure secure access to the platform, both on booking and administration. It prevents unwanted tamper on the data.

Description:

The access control happens on multiple levels: UI and backend. Whenever an user wants to access the platform, its request should be authenticated for sensitive data.

The following points should be behind an authentication point:

- Booking system exposed to citizens through a UI
- Data ingestion services/APIs
- Data administration services/APIs
- Data administration exposed to city managers through a UI

While some services may already have an integrated authentication and access control (databases for example), others rely on 3rd parties' integration, for which Keycloak is used.

2 Where this component fits in the ReGreeneration architecture

Cross-cutting layer, this component should be part of the following components:

- Data Management: Keycloak service
- Secure Database: Built-in access control
- Real time data ingestion: ?
- (Map Integration SDK ?)
- OGS Integration: ?
- (Data pipeline libraries ?)
- Citizen's interactions: Keycloak service
- Admin Dashboard for Cities: Keycloak service
- Data Sources of Cities: Check with Emilie

3 Module description

Whenever the access control is not incorporated, this is done through Keycloak.

Keycloak is a standalone service, to which components are either redirecting their requests (UIs) or relying on a token provided by the Keycloak service (direct APIs authentication).

4 Technical Foundations and Background

Subcomponent/Component	Owner	License
Keycloak	?	Open-Source – Apache?

5 Sequence diagram of module components

The following sub-sections describe the sequence diagrams of the module.

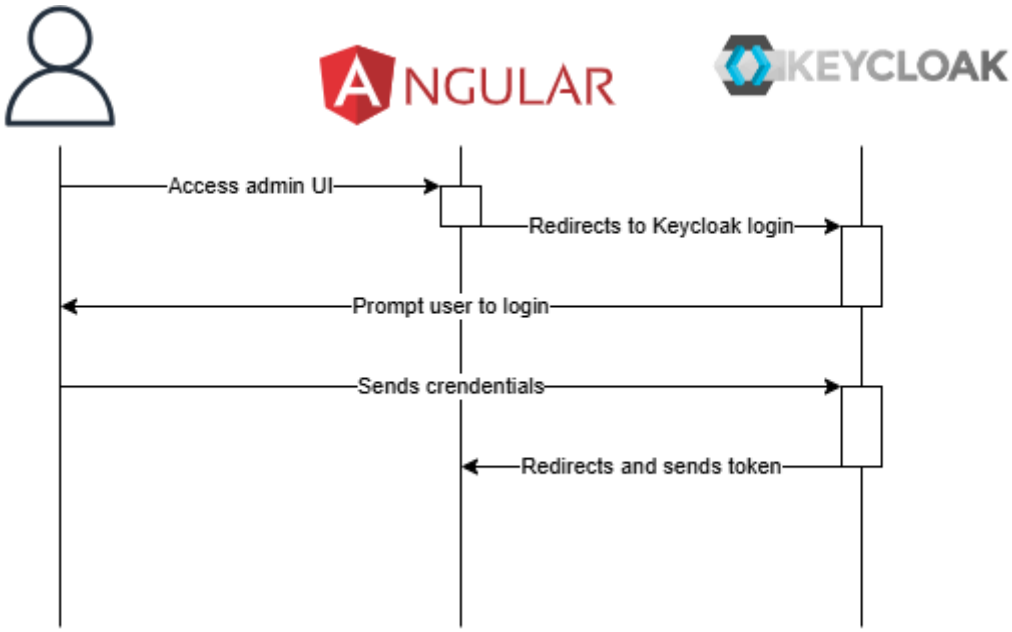
5.1 Frontend

5.1.1 Authentication using Keycloak

This shows the sequence diagram whenever a user accesses the administration UI and authenticates in the system through Keycloak.

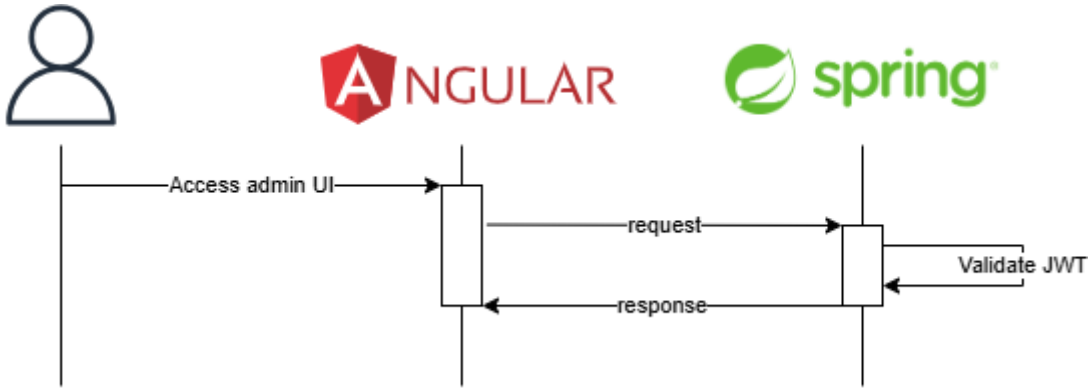
The user accesses the admin UI, which generates an authentication request and redirects to Keycloak.

The user sends its credentials, including automatically its authentication request to Keycloak which authenticates the user and redirects to the admin UI.



5.1.2 Authenticating user requests

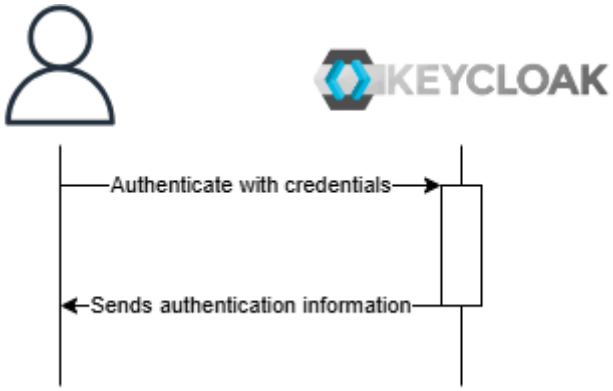
Whenever there is a request from the UI to the backend, the backend checks the validity of the authentication token.



5.2 Backend

5.2.1 Authenticating using Keycloak

Whenever a user wants to reach secured API endpoints, the first step is to authenticate directly to Keycloak. This will return some authentication information (access token) which should be used later in an authorization header.



5.2.2 Authenticating API calls

Whenever a user wants to access secured API endpoints and has an access token (see previous point), the request should contain an authorization header containing that token.