

جامعة الحسن الثاني بالدار البيضاء  
+00101121 | ١٠٠١٢١ | ΤΗΛΕΦΩΝΟ ΟΣΙ | ΕΠΙΧεΙΡΗΣΕΩ  
UNIVERSITÉ HASSAN II DE CASABLANCA



# Rapport Examen Final

Lehcene MOHAMED  
LEMINÉ

Sous la direction de monsieur

Dr. YOUSSEFI Mohamed



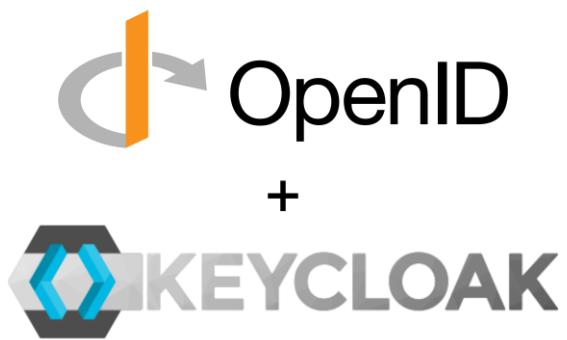
Année universitaire : 2023 – 2024

Master de Recherche SDIA

## 1-Introduction

Ce rapport décrit la conception et la mise en œuvre d'une application de gestion de ressources et de réservations basée sur les microservices. L'application est sécurisée avec Keycloak.

### Les Technologies et les outils utilisés



## 2-Contexte général du projet

On souhaite créer une application basée sur une architecture micro-service qui permet de gérer des réservations concernant des ressources. Chaque réservation concerne une seule ressource. Une ressource est définie par son id, son nom, son type (MATERIEL\_INFO, MATERIEL\_AUDIO\_VUSUEL). Une réservation est définie par son id, son nom, son contexte, sa date, sa durée. Chaque réservation est effectuée par une personne. Une personne est définie par son id, son nom, son email et sa fonction. L'application doit permettre de gérer les ressources et les réservations. Pour faire plus simple, cette application se composera de deux micro-services fonctionnels:

- Un Micro-service qui permet de gérer des « Resources-Service ».
- Un Micro-service qui permet de gérer les réservations effectuées par des personnes.

Les micro-services technique à mettre en place sont :

- Le service Gateway basé sur Spring cloud Gateway
- Le service Discovery base sur Eureka Server ou Consul Discovery (au choix)
- Le service de configuration basé sur Spring cloud config ou Consul Config (au choix)

Pour l'application, nous avons besoin de développer une frontend web, basé sur Angular Framework.

La sécurité de l'application est basée sur Oauth2 et OIDC avec Keycloak comme Provider

Pour les micro-services, il faut générer la documentation des web services Restfull en utilisant la spécification

OpenAPIDoc (Swagger). Prévoir aussi des circuit breakers basés sur Resilience4J comme solution de fault tolerance

Travail de mandé :

Rendre un rapport PDF et le code source des projets au format zip, répondant aux questions suivantes

1. Établir une architecture technique du projet
2. Créer un Projet Maven incluant les micro-services suivants : resources-service, reservation-service, gateway-service, discovery-service, config-service et angular-front-app
3. Développer et tester les micro-services discovery-service et gateway-service et config-service
4. Développer et tester le micro-service resources-service (Entities, DAO, service, DTO, Mapper,

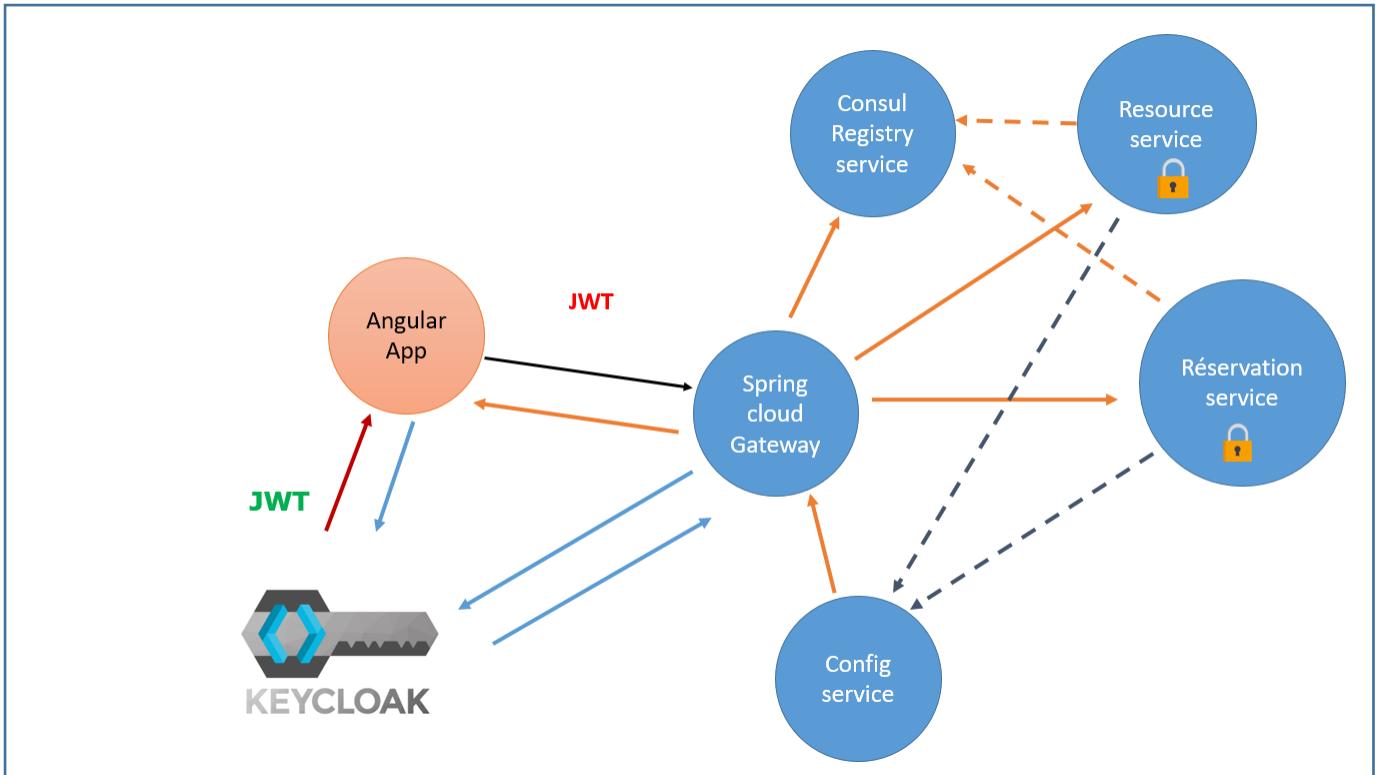
RestController)

5. Développer et tester le micro-service reservation-service (Entities, DAO, service, DTO, Mapper,

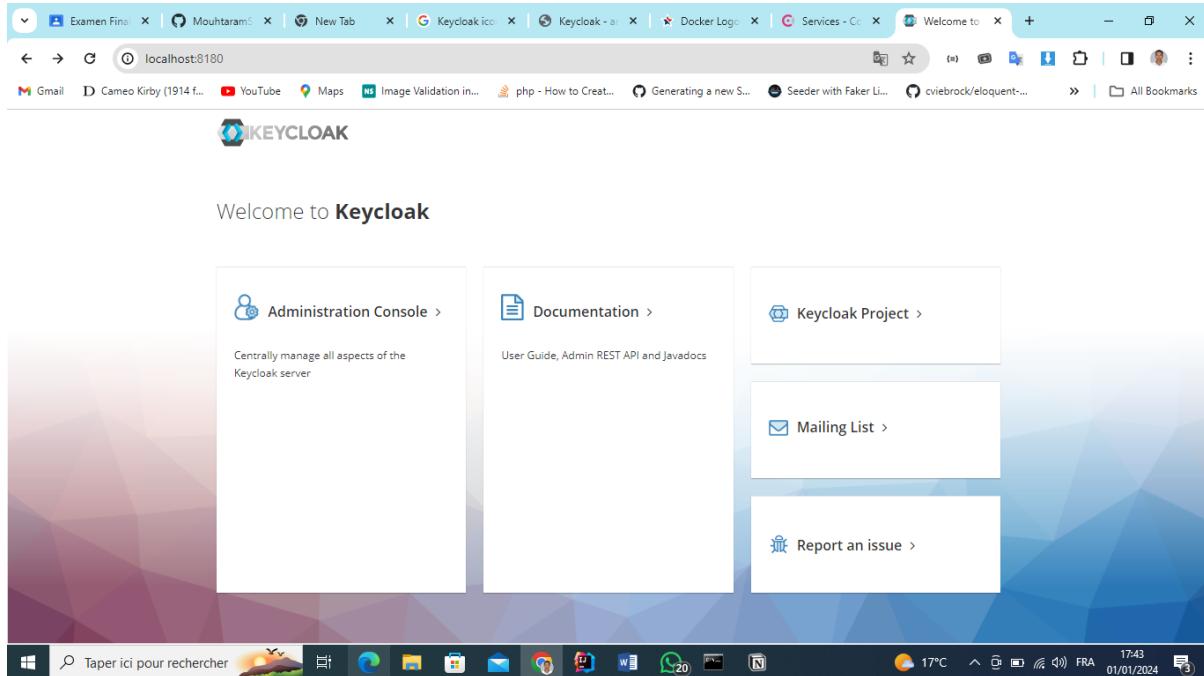
RestController, Client Rest Open Feign)

6. Développer un simple frontend web pour l'application
7. Sécuriser l'application avec une authentification Keycloak
8. Déployer l'application avec Docker et Docker compose

### 3-Architecture



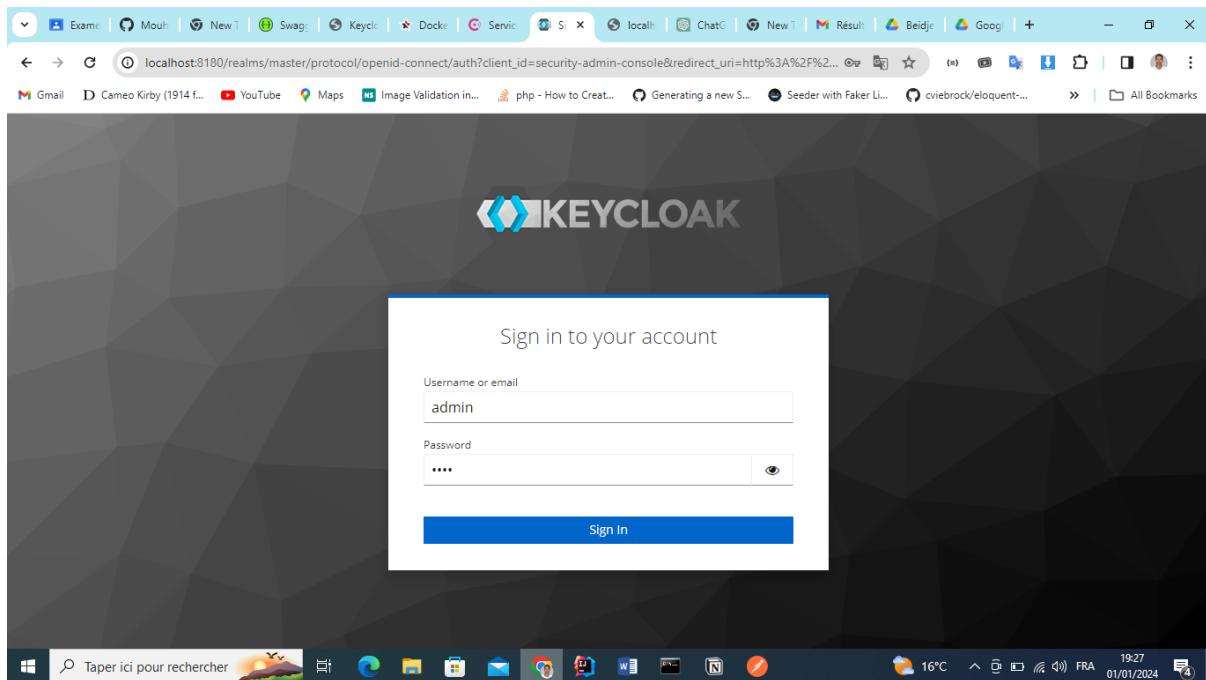
## 4-Keycloak



⇒ Interface de KEYCLOAK

```
C:\Windows\System32\cmd.exe - kc.bat start-dev --http-port=8180
2024-01-01 17:41:53,332 INFO [org.keycloak.quarkus.runtime.hostname.DefaultHostnameProvider] (main) Hostname settings: Base URL: <unset>, Hostname: <request>, Strict HTTPS: false, Path: <request>, Strict BackChannel: false, Admin URL: <unset>, Admin: <request>, Port: -1, Proxied: false
2024-01-01 17:41:56,973 WARN [io.quarkus.agroal.runtime.DataSources] (main) Datasource <default> enables XA but transaction recovery is not enabled. Please enable transaction recovery by setting quarkus.transaction-manager.enable-recovery=true, otherwise data may be lost if the application is terminated abruptly
2024-01-01 17:41:58,496 WARN [org.infinispan.PERSISTENCE] (keycloak-cache-init) ISPN000554: jboss-marshalling is deprecated and planned for removal
2024-01-01 17:41:58,690 WARN [org.infinispan.CONFIG] (keycloak-cache-init) ISPN000569: Unable to persist Infinispan internal caches as no global state enabled
2024-01-01 17:41:58,854 INFO [org.infinispan.CONTAINER] (keycloak-cache-init) ISPN000556: Starting user marshaller 'org.infinispan.jboss.marshalling.core.JBossUserMarshaller'
2024-01-01 17:42:03,612 INFO [org.keycloak.broker.provider.AbstractIdentityProviderMapper] (main) Registering class org.keycloak.broker.provider.mappersync.ConfigSyncEventListener
2024-01-01 17:42:04,288 INFO [org.keycloak.connections.infinispan.DefaultInfinispanConnectionProviderFactory] (main) Node name: node_805964, Site name: null
2024-01-01 17:42:05,946 INFO [io.quarkus] (main) Keycloak 23.0.0 on JVM (powered by Quarkus 3.2.9.Final) started in 15.674s. Listening on: http://0.0.0.0:8180
2024-01-01 17:42:05,947 INFO [io.quarkus] (main) Profile dev activated.
2024-01-01 17:42:05,950 INFO [io.quarkus] (main) Installed features: [agroal, cdi, hibernate-orm, jdbc-h2, jdbc-mariadb, jdbc-mssql, jdbc-mysql, jdbc-oracle, jdbc-postgresql, keycloak, logging-gelf, micrometer, narayana-jta, reactive-routes, resteasy-reactive, resteasy-reactive-jackson, smallrye-context-propagation, smallrye-health, vertx]
2024-01-01 17:42:05,956 WARN [org.keycloak.quarkus.runtime.KeycloakMain] (main) Running the server in development mode. DO NOT use this configuration in production.
```

⇒ Voici je démarre KEYCLOAK avec ligne de commande sur port 8180



⇒ Connexion avec userName : admin & password : admin

Client ID	Name	Type	Description	Home URL
account	#{client_account}	OpenID Connect	–	http://localhost:8180/realmms/test_realm/account/
account-console	#{client_account-conso...}	OpenID Connect	–	http://localhost:8180/realmms/test_realm/account/
admin-cli	#{client_admin-cl...}	OpenID Connect	–	–
broker	#{client_broker}	OpenID Connect	–	–
frontend	–	OpenID Connect	–	http://localhost:4200
realm-management	#{client_realm-ma...}	OpenID Connect	–	–
security-admin-console	#{client_security-a...}	OpenID Connect	–	http://localhost:8180/admin/test_realm/console/

⇒ Authentification est réussie

The screenshot shows the Keycloak admin interface for a 'test\_realm'. On the left, a sidebar lists 'Manage', 'Clients' (which is selected), 'Client scopes', 'Realm roles', 'Users', 'Groups', 'Sessions', 'Events', 'Configure', and 'Realm settings'. The main area displays 'Clients > Client details' for 'frontend' (OpenID Connect). The 'Enabled' toggle is turned on. The 'General settings' tab is active, showing fields for 'Client ID' (frontend), 'Name' (empty), and 'Description' (empty). Below these are 'Save' and 'Revert' buttons. A 'Jump to section' sidebar on the right lists 'General settings', 'Access settings', 'Capability config', 'Login settings', and 'Logout settings'. The system status bar at the bottom indicates it's 16°C, 19:12, FRA, and the date is 01/01/2024.

This screenshot shows the 'Access settings' tab for the 'frontend' client. It includes fields for 'Root URL' (http://localhost:4200), 'Home URL' (http://localhost:4200), 'Valid redirect URLs' (http://localhost:4200/\*), 'Valid post logout redirect URLs' (http://localhost:4200/\*), and 'Web origins' (http://localhost:4200). A 'Save' button is at the bottom. The 'Jump to section' sidebar remains the same as in the previous screenshot. The system status bar at the bottom indicates it's 16°C, 19:12, FRA, and the date is 01/01/2024.

⇒ Voici je crée un client : frontend pour ANGULAR

The screenshot shows the Keycloak admin interface for the 'test\_realm'. The left sidebar has a 'Users' item selected. The main area displays a table of users with columns: Username, Email, Last name, First name, and Status. Two users are listed: 'user' (Email: mohamedleminlehcene@gmail.com, Last name: Lehcene, First name: Mohamed Lemine) and 'user2' (Email: user2@gmail.com, Last name: Lemine, First name: Mohamed).

⇒ Voici liste des utilisateurs

The screenshot shows the Keycloak admin interface for the 'test\_realm'. The left sidebar has a 'Users' item selected. The main area is titled 'User details' for the user 'user'. The 'Role mapping' tab is active. The table shows the assigned roles: 'USER' (Inherited: False) and 'default-roles-test\_realm' (Inherited: False, Description: \${role\_default-roles}).

⇒ Utilisateur user avec role USER

The screenshot shows the Keycloak administration interface for a realm named "test\_realm". The user "user2" is selected. The "Role mapping" tab is active, showing four assigned roles: "USER", "ADMIN", and two others that are partially visible. The "Enabled" switch is turned on. The interface includes a search bar, a "Hide inherited roles" checkbox, and buttons for "Assign role" and "Unassign". The bottom of the screen shows a Windows taskbar with various icons and system status.

Name	Inherited	Description
USER	False	-
ADMIN	False	-
default-roles-test_realm	False	\${role_default-roles}

⇒ Utilisateur user2 avec roles : USER & ADMIN

## 5-SWAGGER

The screenshot shows a browser window with multiple tabs open. The active tab is 'Swagger' at 'localhost:8082/swagger-ui/index.html#/'. The page displays the Swagger UI for a 'reservation-rest-controller'. It lists several API endpoints:

- PUT /api/reservations/update/{id}
- POST /api/reservations/save
- GET /api/reservations
- GET /api/reservations/personne/{personId}
- GET /api/reservations/ById/{reservationId}
- DELETE /api/reservations/{id}

Below this, there is another section for 'personne-rest-controller' which contains similar endpoint definitions. The browser's taskbar at the bottom shows various pinned icons and the system status bar indicates it's 17°C, 18:02, FRA, and the date is 01/01/2024.

⇒ Voici je consulte interface swagger pour micro service reservation sur port : 8082

The screenshot shows a browser window with multiple tabs open. The active tab is 'Swagger' at 'localhost:8081/swagger-ui/index.html#/'. The page displays the Swagger UI for a 'ressource-rest-controller'. It lists several API endpoints:

- PUT /api/ressources/update/{id}
- POST /api/ressources/save
- GET /api/ressources
- GET /api/ressources/{id}
- DELETE /api/ressources/delete/{id}

Below the endpoints, there is a 'Schemas' section that shows a detailed view of an 'AbstractJsonSchemaPropertyObject' schema, with an 'Item' section expanded. The browser's taskbar at the bottom shows various pinned icons and the system status bar indicates it's 17°C, 18:04, FRA, and the date is 01/01/2024.

⇒ Voici j'ai consulté interface swagger pour micro service  
⇒ ressource sur port : 8081

## 6-POSTMAN

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Home, Workspaces, API Network, Collections, APIs, Environments, and History. The main area displays a request configuration for a GET method to the URL <http://localhost:5555/ressource-service/api/ressources>. The Body tab is selected, showing a JSON payload:

```
1 {  
2   ... "id": 4,  
3   ... "nom": "ressource4",  
4   ... "ressourceType": "MATERIEL_INFO"  
5 }
```

Below the request details, the status bar indicates: Status: 401 Unauthorized Time: 19 ms Size: 541 B. The bottom navigation bar includes icons for Postbot, Runner, Start Proxy, Cookies, Trash, and a search bar labeled "Taper ici pour rechercher".

- ⇒ <http://localhost:5555/ressource-service/api/ressources> : pour afficher liste des ressources mais il m'envoie comme résultat **401 Unauthorized** car je besoin d'authentifier et encore envoyer JWT TOKEN

The screenshot shows the Postman interface with a POST request to `http://localhost:810/realm/test_realm/protocol/openid-connect/token`. The request body contains:

```

{
  "grant_type": "password",
  "client_id": "frontend",
  "username": "user2",
  "password": "1234"
}

```

The response status is 200 OK, and the JSON response is:

```

{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCgoAiSl0iUiwi2lKiia6ICj4Zm4tMTBeCThRNzkjcy90ZXN0X31lyWxtIwlwYXVkjIjoivYnjb3vudCisInN1Yi16Imb0GixM0k8LwM1ZwYtND12kZc1iyTbj1vb19dgdF0ZS16ImwM2E5NjfilwMNGEtNQgw0094Mj2llwI18NDNkMj5c6ZGZ1OC1sImFjci16TjE1LCj2Vzc1y6eyjy2xIcyIwYjvZmzsaw1x2fjy2zc1yisIm1Zm1bHQtcn9szKXtdVzdf9yZwPabs1sik result of the action.
Ons1YhN1b59dUcDZc1y6eyjy2xIcyIwYjvZmzsaw1x2fjy2zc1yisIm1Zm1bHQtcn9szKXtdVzdf9yZwPabs1sik
zYTk2MwItzyYSS00DA4LTgyNm1tVj0M0Q2yNz1k2m141i1w1Zw1haxwfdmVyaZwQ10mZbhHn1lCuYw11joiTw9oYw11ZCmZw1pbmUl1LJwcmVmZXJyZwRfdXN1cmBhwU01j1c1j2vhy1s1m
dpdnVw2hbwU101N2hbwVwK1wZm1ftawksX2k6hmu1013Mw1bmUl1C1l1wPbc161nVZ2IyQd0tYw1s1mVbS39,
BMwKV1EiiYu11IGJUBNt80m0C819agZPkj3C_TtxvCoUv56Au91mba43pvt7eoKMNxI469GbaohLX6sKw1s1xNbIASQDoyz1PGGYjigouZa3wZLPwVxhGX_EvGbW09qKUOyqkk9f4aA
zRYza1wqXkw4mD0H8mWzg3Zp5Up1a0-m24LQmH7xyss3MXxxz7Le6b1lVYKywp102EPiV6-Tbn-YoIfz2n0299XQ6gMxU68_FmOpL4eShfxhyFWKe7Q-v6ce4io_EEBfxAaS39
eyHazzoV1JdZ-UfifgjPHBK81dy3zHlUna1189",
}

```

⇒ Voici je fais une authentification avec POSTMAN sur KEYLCOAK pour obtenir JWT TOKEN

The screenshot shows the Postman interface with a GET request to `http://localhost:5555/resource-service/api/resources`. The Authorization header is set to `Bearer Token` with the token `eyJhbGciOiJSUzI1NiIsInR5cCgoAiSl0iUiwi...`.

The response status is 200 OK, and the JSON response is:

```

[
  {
    "id": 1,
    "nom": "ressource1",
    "ressourceType": "MATERIEL_INFO"
  },
  {
    "id": 2,
    "nom": "ressource2",
    "ressourceType": "MATERIEL_INFO"
  },
  {
    "id": 3,
    "nom": "ressource3",
    "ressourceType": "MATERIEL_AUDIO_VUSUEL"
  }
]

```

⇒ Voici je affiche liste des ressources car j'envoie JWT TOKEN que j'obtiens par KEYCLOAK

The screenshot shows the Postman interface with a successful POST request to `http://localhost:5555/ressource-service/api/ressources/save`. The request body is a JSON object:

```
1 {  
2   "id": 5,  
3   "nom": "ressource5",  
4   "ressourceType": "MATERIEL_INFO"  
5 }
```

The response status is 200 OK, time 54 ms, size 411 B. The response body is:

```
1 {  
2   "id": 5,  
3   "nom": "ressource5",  
4   "ressourceType": "MATERIEL_INFO"  
5 }
```



<http://localhost:5555/ressource-service/api/ressources/save>  
pour une nouvelle ressource et encore j'envoie JWT TOKEN

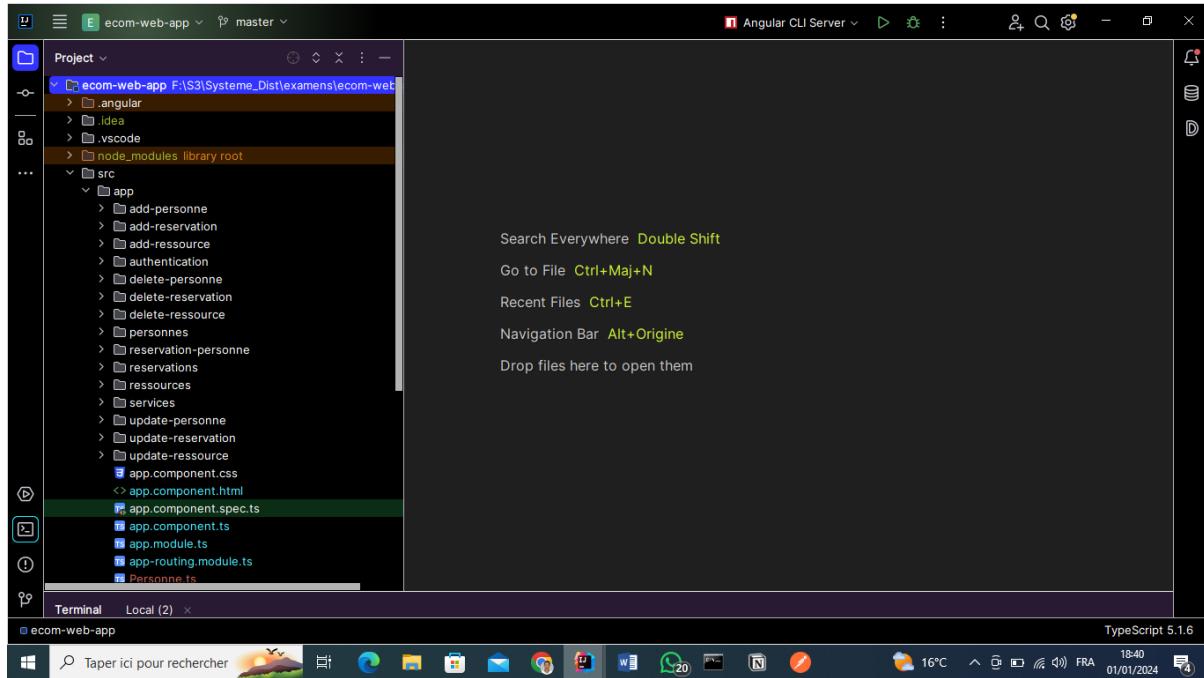
The screenshot shows the Postman interface with a successful GET request to `http://localhost:5555/ressource-service/api/ressources`. The Authorization header is set to `Bearer eyJhbGciOiJSUzI1NiIsInR5cC1gOjAISldUliwi...`.

The response status is 200 OK, time 28 ms, size 661 B. The response body is a list of resources:

```
12 [ {  
13   "id": 3,  
14   "nom": "ressource3",  
15   "ressourceType": "MATERIEL_AUDIO_VSUEL"  
16 },  
17   {  
18     "id": 4,  
19     "nom": "ressource4",  
20     "ressourceType": "MATERIEL_INFO"  
21 },  
22   {  
23     "id": 5,  
24     "nom": "ressource5",  
25     "ressourceType": "MATERIEL_INFO"  
26   }  
27 ]
```

⇒ Voici l'ajoute du nouvelle ressource est réussi

## 7-Structure du projet ANGULAR



## 8-CONSUL DISCOVERY

Interface CONSUL qui contient l'ensemble des microservices qui enregistre sur eux

The screenshot shows the Consul UI interface running in a browser window. The URL is `localhost:8500/ui/dc1/services`. The left sidebar has a dark theme with the following navigation items:

- Overview
- Services** (selected)
- Nodes
- Key/Value
- Intentions
- ACCESS CONTROLS
- Tokens
- Policies
- Roles
- Auth Methods
- ORGANIZATION
- Peers

Below the sidebar, it says "Consul v1.16.2". The main content area is titled "Services 5 total". It lists five services with their status and instance count:

- reservation-service** (red X icon) - 1 instance
- ressource-service** (red X icon) - 1 instance
- consul** (green checkmark icon) - 1 instance
- config-service** (green checkmark icon) - 1 instance
- gateway-service** (green checkmark icon) - 1 instance

At the top of the main content area, there is a warning message from the command line:

```
C:\Windows\System32\cmd.exe - consul agent -server -bootstrap-expect=1 -data-dir=consul-data -ui -bind=1...  
2024-01-01T17:38:15.652+0100 [WARN]  agent.http: This request used the token query parameter  
which is deprecated and will be removed in Consul 1.17: logUrl="/v1/catalog/services?token=<hidden>  
index=12624&token=<hidden>"  
2024-01-01T17:38:15.964+0100 [WARN]  agent.http: This request used the token query parameter  
which is deprecated and will be removed in Consul 1.17: logUrl="/v1/catalog/services?token=<hidden>  
dden"  
2024-01-01T17:38:15.964+0100 [WARN]  agent.http: This request used the token query parameter  
which is deprecated and will be removed in Consul 1.17: logUrl="/v1/catalog/services?token=<hidden>  
dden">
```

The browser taskbar at the bottom shows various open tabs and system icons.

## 9-SECURITE DES MICROSERVICES

### Classe JwtAuthConverter:

```
@Component
public class JwtAuthConverter implements Converter<Jwt,
AbstractAuthenticationToken> {
    private final JwtGrantedAuthoritiesConverter
jwtGrantedAuthoritiesConverter=new JwtGrantedAuthoritiesConverter();

    @Override
    public AbstractAuthenticationToken convert(Jwt jwt) {
        Collection<GrantedAuthority> authorities = Stream.concat(
            jwtGrantedAuthoritiesConverter.convert(jwt).stream(),
            extractResourceRoles(jwt).stream()
        ).collect(Collectors.toSet());
        return new JwtAuthenticationToken(jwt,
authorities,jwt.getClaim("preferred_username"));
    }

    private Collection<GrantedAuthority> extractResourceRoles(Jwt jwt) {
        Map<String , Object> realmAccess;
        Collection<String> roles;
        if(jwt.getClaim("realm_access")==null){
            return Set.of();
        }
        realmAccess = jwt.getClaim("realm_access");
        roles = (Collection<String>) realmAccess.get("roles");
        return roles.stream().map(role->new
SimpleGrantedAuthority(role)).collect(Collectors.toSet());
    }
}
```

## Classe SecurityConfig

```
@Configuration
@EnableWebSecurity
@EnableMethodSecurity(prePostEnabled = true)
public class SecurityConfig {

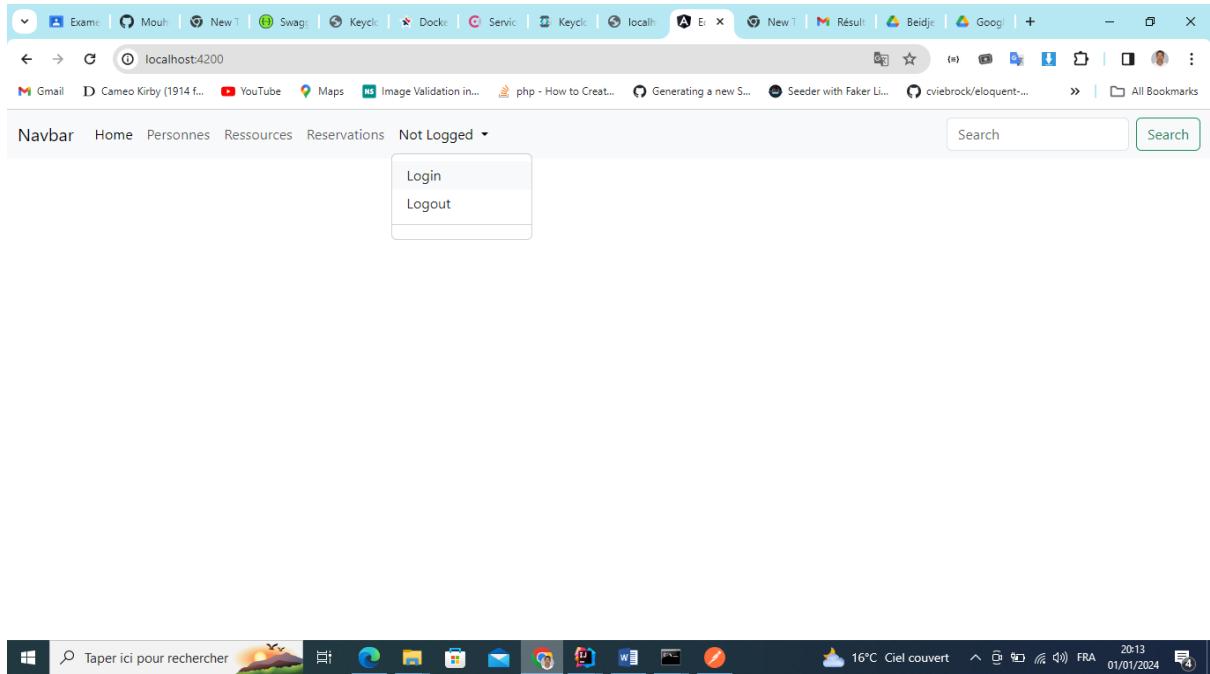
    private JwtAuthConverter jwtAuthConverter;

    public SecurityConfig(JwtAuthConverter jwtAuthConverter) {
        this.jwtAuthConverter = jwtAuthConverter;
    }

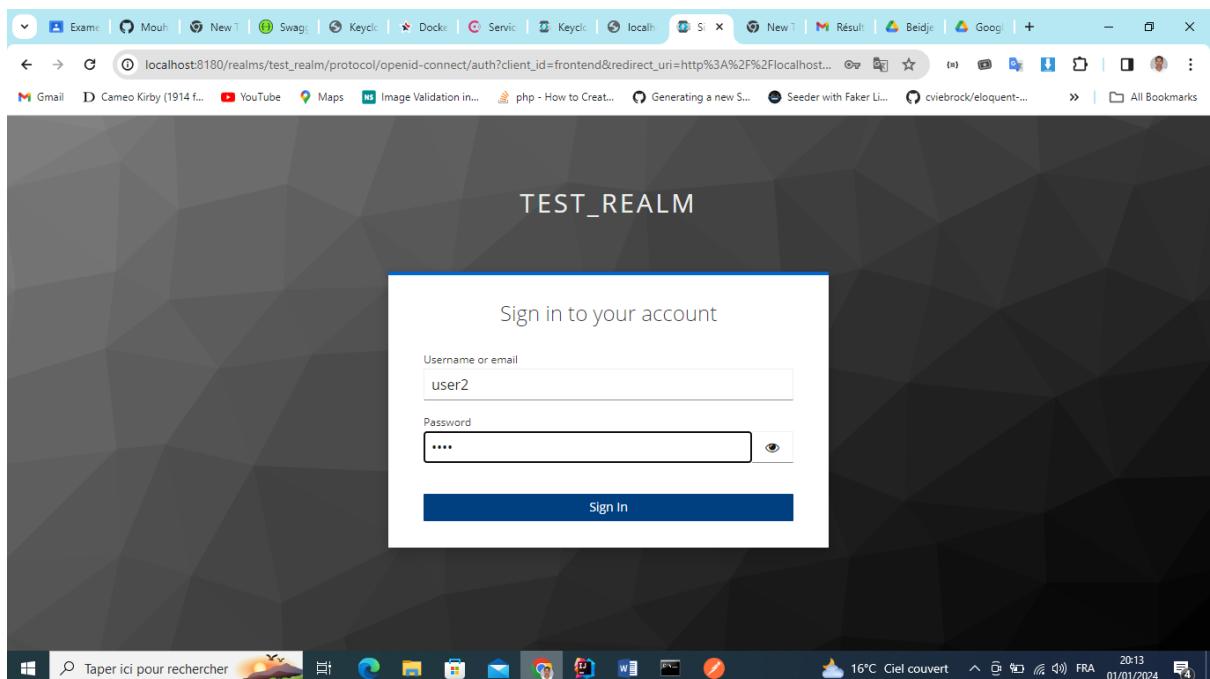
    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http)
throws Exception {
        return http
            .csrf(csrf->csrf.disable())
            .authorizeHttpRequests(ar -> ar.requestMatchers(
                "/v3/api-docs/**",
                "/swagger-ui/**",
                "/swagger-ui.html",
                "health/**",
                "/h2-console/**").permitAll())
            .authorizeHttpRequests(ar->ar.anyRequest().authenticated())
            .oauth2ResourceServer(o2->o2.jwt(jwt-
>jwt.jwtAuthenticationConverter(jwtAuthConverter)))
            .headers(h->h.frameOptions(fo->fo.disable()))
            .csrf(csrf->csrf.ignoringRequestMatchers("/h2-console/**"))
            .build();
    }

    @Bean
    CorsConfigurationSource corsConfigurationSource() {
        CorsConfiguration configuration = new CorsConfiguration();
        configuration.setAllowedOrigins(Arrays.asList("*"));
        configuration.setAllowedMethods(Arrays.asList("*"));
        configuration.setAllowedHeaders(Arrays.asList("*"));
        UrlBasedCorsConfigurationSource source = new
        UrlBasedCorsConfigurationSource();
        source.registerCorsConfiguration("/**", configuration);
        return source;
    }
}
```

## 10- ANGULAR



⇒ Page home de mon application : <http://localhost:4200/>



⇒ Voici lors ce que j'ai cliqué sur personnes il m'a redirigé vers login qui propose par keycloak cette root est protégé par guards qui demande authentification

localhost:4200/personnes

Navbar Home Personnes Ressources Reservations user2

Add Personne

#	Name	Email	Operations
---	------	-------	------------

Search

## List Personnes



⇒ Voici liste des personnes après l'authentification avec user2 qui a mot de passe : 1234 qui est à ce moment ne contient aucun personne

localhost:4200/ressources

Navbar Home Personnes Ressources Reservations user2

Add Ressource

#	Nom	RessourceType	Operations
1	ressource1	MATERIEL_INFO	<button>DELETE</button> <button>UPDATE</button>
2	ressource2	MATERIEL_INFO	<button>DELETE</button> <button>UPDATE</button>
3	ressource3	MATERIEL_AUDIO_VUSUEL	<button>DELETE</button> <button>UPDATE</button>

Search

## List Ressources



⇒ Voici liste des ressources

localhost:4200/addPersonne

Nom: Mohamed Lemine

Email: mohamedleminlehcene@gmail.com

Ok



⇒ Pour ajouter nouvelle personne

localhost:4200/personnes

#	Name	Email	Operations
1	Mohamed Lemine	mohamedleminlehcene@gmail.com	<a href="#">DELETE</a> <a href="#">UPDATE</a> <a href="#">Reservations</a>



⇒ Voici ajoutassions du nouvelle personne est réussie

A screenshot of a web browser window titled "localhost:4200/updatePersonne/1". The page displays a modal dialog box for updating a person's information. The modal has a red "Cancel" button at the top left. Inside, there are two input fields: one for "Nom" containing "Mohamed Lemine" and another for "Email" containing "mohamedleminlehcene@gmail.com". At the bottom left of the modal is a blue "Ok" button.

## Update Infos Personne

Cancel

Nom

Mohamed Lemine

Email

mohamedleminlehcene@gmail.com

Ok



⇒ Pour modifier l'information de personne par son ID : 1

A screenshot of a web browser window titled "localhost:4200/personnes". The page displays a table titled "List Personnes". The table has columns for "#", "Name", "Email", and "Operations". There is one row shown, with values: #1, Mohamed, mohamedleminlehcene@gmail.com, and three buttons in the "Operations" column: "DELETE" (red), "UPDATE" (green), and "Reservations" (blue).

## List Personnes

Add Personne

#

Name

Email

Operations

1 Mohamed

mohamedleminlehcene@gmail.com

DELETE

UPDATE

Reservations



⇒ Voici modification de l'information de personne de son ID égal 1 est réussie

The screenshot shows a web browser window with the URL `localhost:4200/reservationPersonne/1`. The page title is "List des Reservations pour Personne : Mohamed". A yellow button labeled "Add Reservation" is visible. Below it is a table with columns: ID, Nom, Contexte, Date, Duree, and Nom de Ressource. The table currently has one row with the following data:

ID	Nom	Contexte	Date	Duree	Nom de Ressource
	Mohamed	sadia	12/12/23		



⇒ Liste des réservations pour personne qui a ID : 1

The screenshot shows a web browser window with the URL `localhost:4200/addReservation/1`. The page title is "Add Reservation". The form fields are as follows:

- Nom: reservation new
- Contexte: sdia
- Durée: 12/12/23
- Ressource Id: 1
- Personne: 1

A red "Cancel" button is at the top left, and a blue "Ok" button is at the bottom left.



⇒ Pour faire une réservation pour personne qui a ID : 1

localhost:4200/reservations

Navbar Home Personnes Ressources Reservations user2 ▾

List des Reservations

ID	Nom	Contexte	Date	Durée	Operations
1	reservation new	sdia	01/01/24	12/12/23	<button>DELETE</button> <button>UPDATE</button> <button>Details</button>



⇒ Voici la réservation est réussie

localhost:4200/reservations

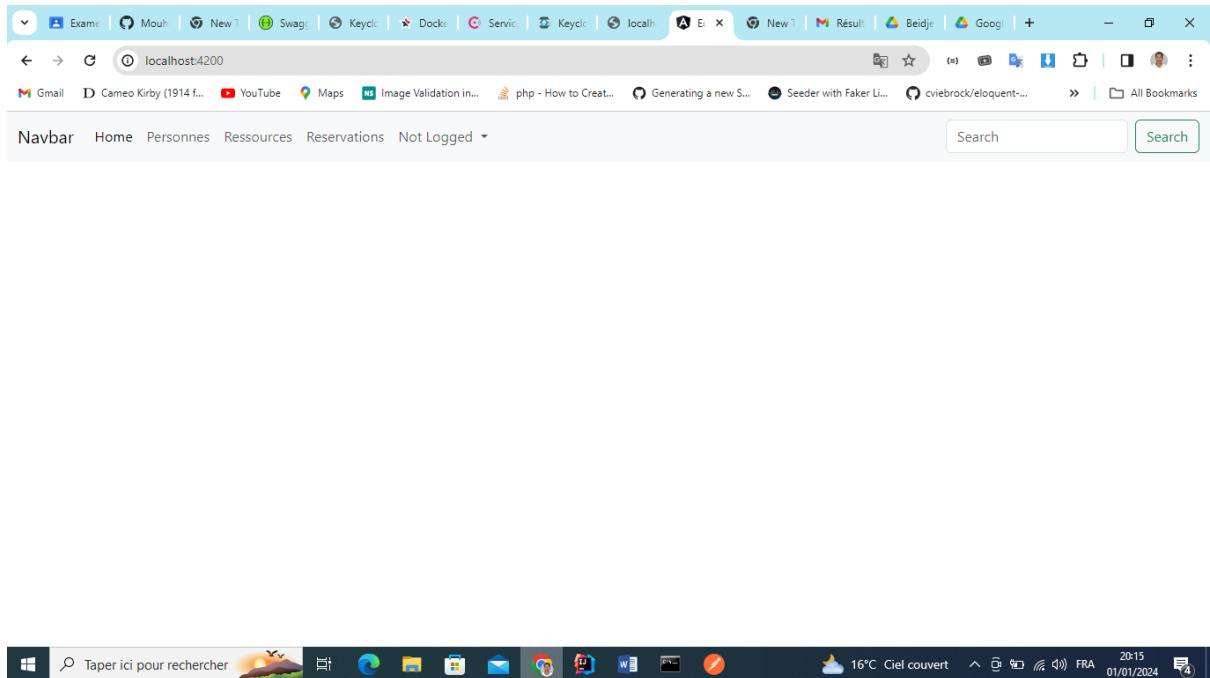
Navbar Home Personnes Ressources Reservations user2 ▾

List des Reservations

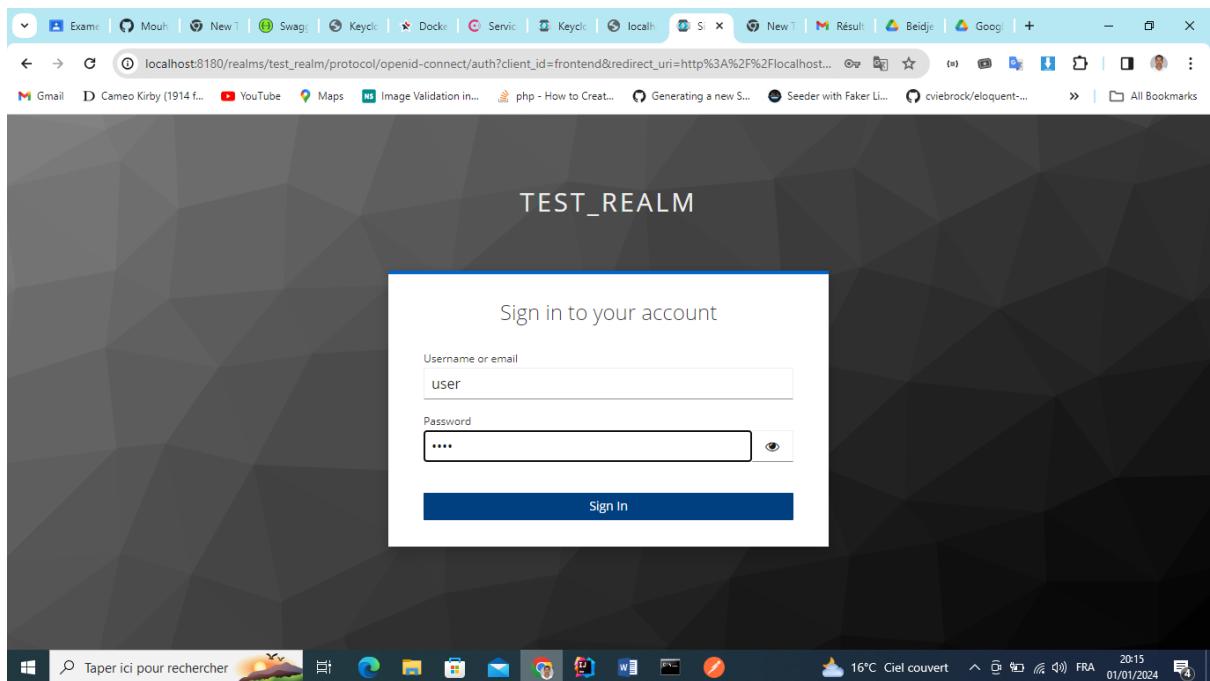
ID	Nom	Contexte	Date	Durée	Operations
1	reservation new	sdia	01/01/24	12/12/23	<button>DELETE</button> <button>UPDATE</button> <button>Details</button>



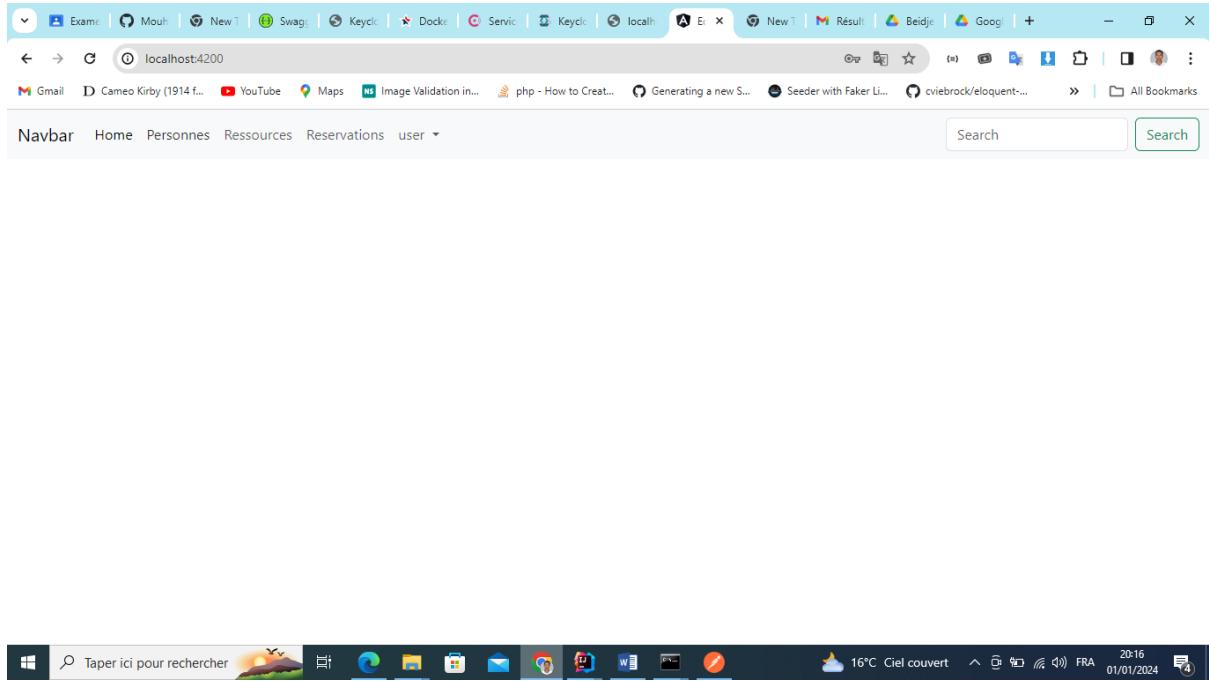
⇒ Pour faire LOGOUT lors ce qu'on clique sur bouton Logout



⇒ Voici LOGOUT est réussie



⇒ Authentification avec utilisateur username : user & password : 1234



⇒ Voici lors ce que j'ai authentifié avec utilisateur de username user ne peut pas d'accéder au liste des personnes car je protégé par rôle ADMIN

MERCI DE VOTRE ATTENTION