

RÉPUBLIQUE DU CAMEROUN

REPUBLIC OF CAMEROON

Peace – Work – Fatherland

UNIVERSITÉ DE DSCHANG

UNIVERSITY OF DSCHANG

Scholae Thesaurus Dschangensis Ibi Cordum

P 96, Dschang (Cameroun) -Tél. /Fax

(237) 233 45 13 811

Website: <http://www.univ-dschang.org>



FACULTE DES SCIENCES

FACULTY OF SCIENCE

**Département de Mathématiques
et Informatique**

**Department of Mathematics and
Computer Science**

BP 67, Dschang (Cameroun)

E-mail : udsrectorat@univ-dschang.org

Licence III Informatique

INF 312 : Base de Données

Projet

Thème : Création d'un système de gestion des états

Enseignant:

Dr Benoit AZANGUEZET

2023 /2024

LISTE DES PARTICIPANTS

N°	Noms	Prénoms	Matricules
01	KENFACK FONGANG	VICTOR CYNTICHE	CM-UDS-21SCI0555
02	NANSEU NJAKOU	STEVE WILFRIED	CM-UDS-19SCI0892
03	MFENTAM	MOHAMMED SALAM	CM-UDS-21SCI0941
04	LOMOFOUET NDONGMO	HANDREY JAURESS	CM-UDS-21SCI0124
05	FEUTSEU TCHUENKAM	IVAN LIONEL	CM-UDS-20SCI0027
06	JIOTANG DONGMEZA	Rostov Perez	CM-UDS-21SCI1068
07	SOFEUGANG	JUNIOR HERVE	CM-UDS-19SCI0100
09	KEOU TCHEUMEDJIE	NINE-NICAISE	CM-UDS-21SCI0562
10	YETTCHOM TCHIZE	ELODIE ROSITA	CM-UDS-21SCI0691
11	KANE DJOUFACK	FLORIANE JOSEPHINE	CM-UDS-20SCI039
11	DJOUFACK	ERICA	CM-UDS-19SCI0325

Chapitre 1 : Front-end

1. Objectif

Le front-end de notre projet repose sur une architecture basée sur les composants. En fait, ces composants représentent les différentes parties de l'interface utilisateur interactive.

2. Technologies utilisées

Notre plateforme repose sur l'utilisation du framework React.js. Ce framework est de nature JavaScript développé par Facebook et repose sur la notion de composant. Pour son utilisation, elle nécessite un certain nombre d'installations de dépendances à l'instar de :

2.1 Node.js

C'est un environnement d'exécution JavaScript côté serveur, ce qui signifie que les développeurs peuvent utiliser JavaScript à la fois côté client et côté serveur.

2.2 Npm (Node Package Manager)

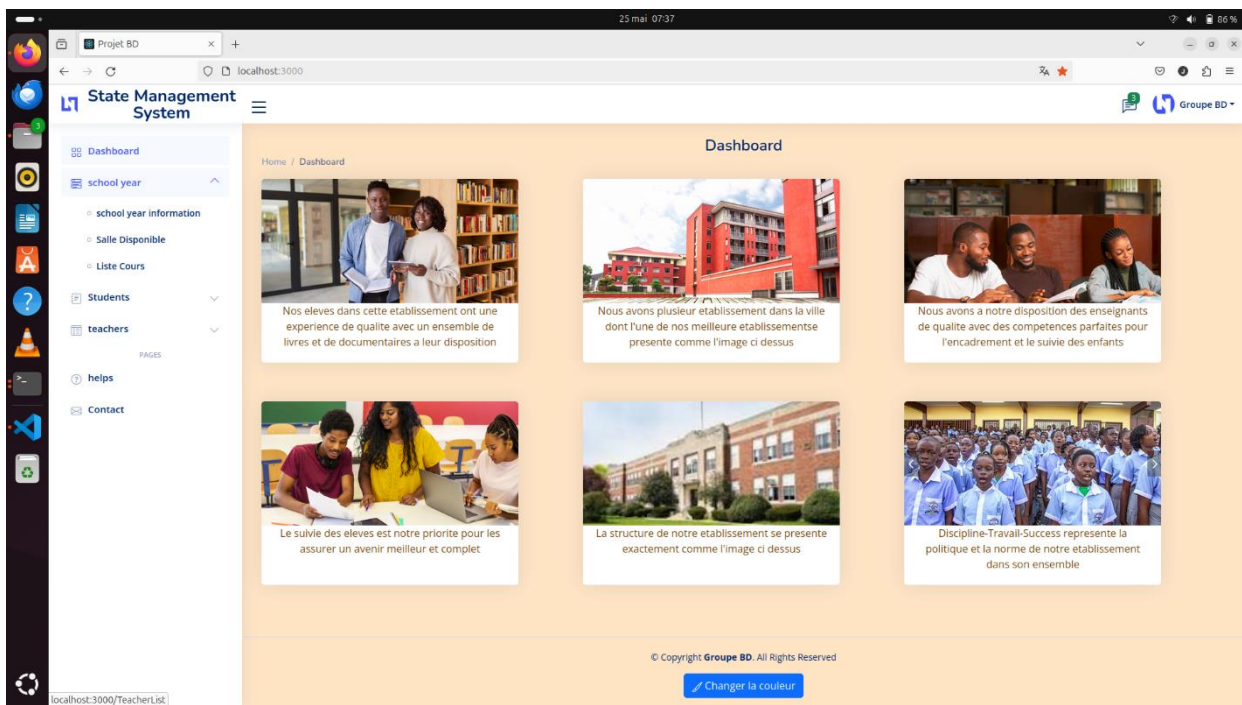
C'est un gestionnaire de paquets par défaut pour Node.js. Il permet d'installer, gérer et de partager des packages JavaScript et des dépendances pour des projets Node.js. Comme l'une des fonctionnalités nous avons :

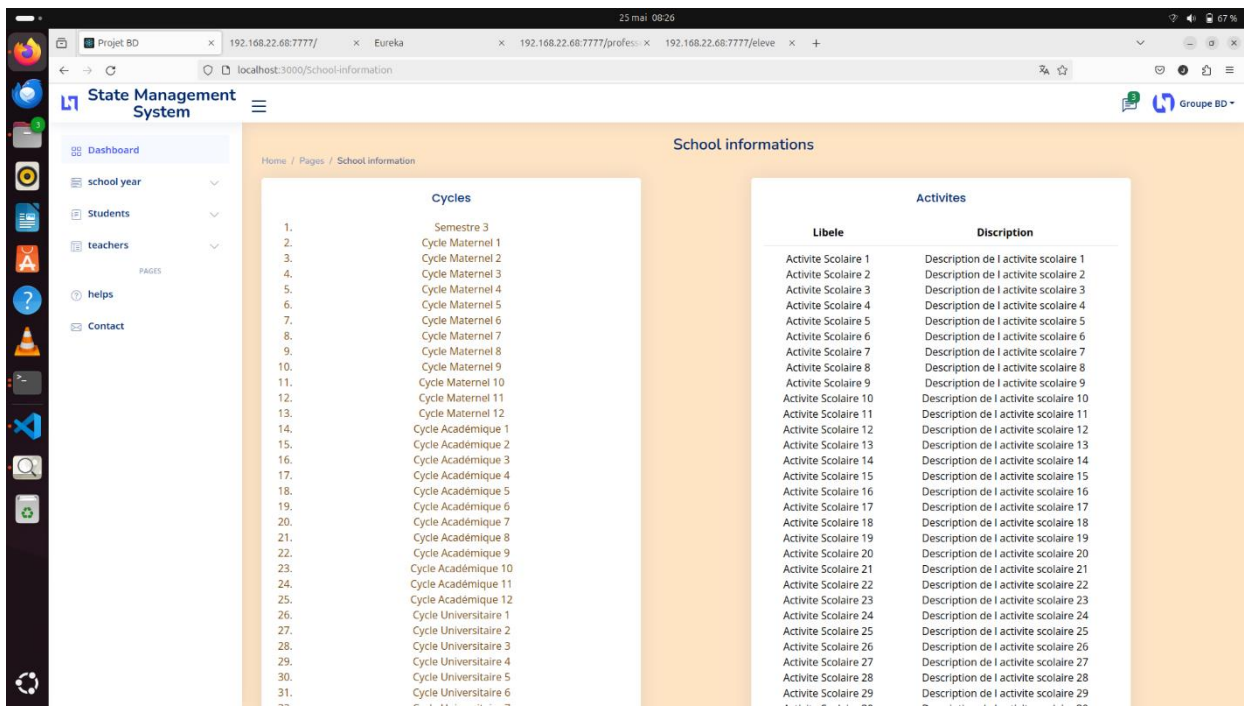
Scripts personnalisés: il permet aux développeurs de définir des scripts personnalisés dans leur fichier package.json, ce qui facilite l'exécution de tâches telles que la compilation, les tests et le déploiement.

2.3 vite

C'est un outil de développement rapide pour les applications react.js. Il est conçu pour améliorer le processus de développement en fournissant une configuration et des fonctionnalités prêtes à l'emploi, ce qui permet aux développeurs de se concentrer sur l'écriture du code plutôt que sur la configuration.

3. Interface de nos différents composants





State Management System

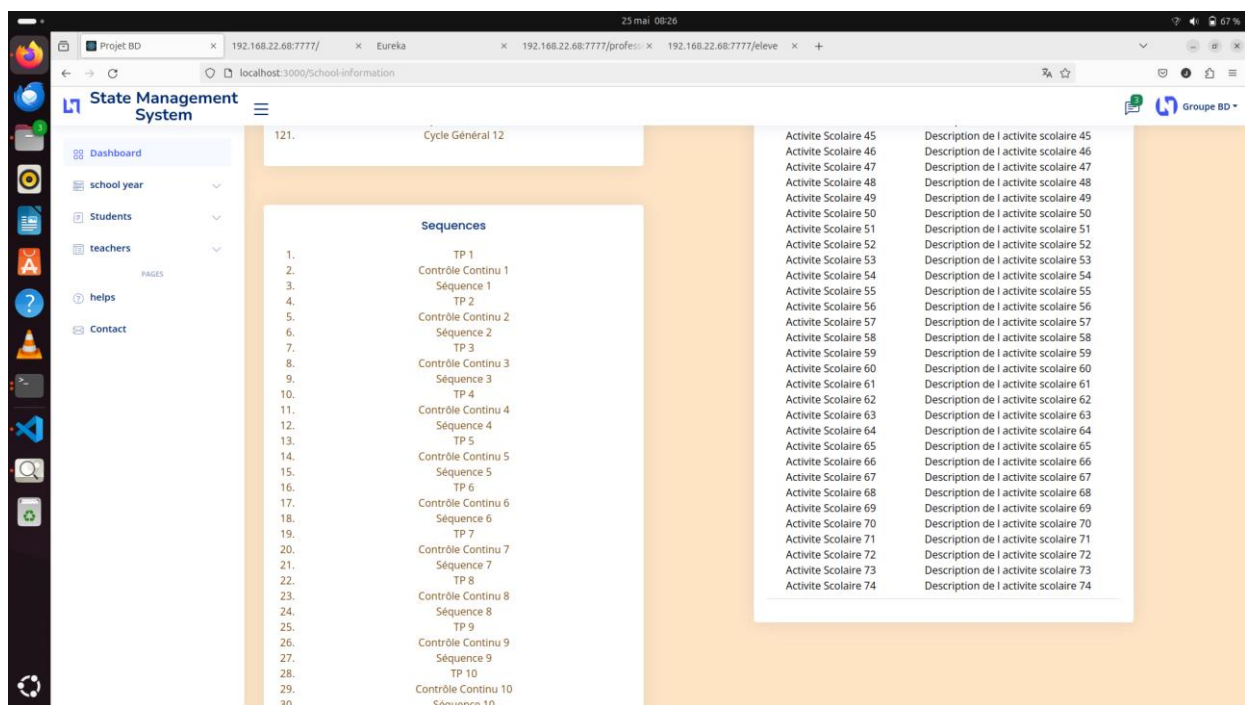
Home / Pages / School information

Cycles

1.	Semestre 3
2.	Cycle Maternel 1
3.	Cycle Maternel 2
4.	Cycle Maternel 3
5.	Cycle Maternel 4
6.	Cycle Maternel 5
7.	Cycle Maternel 6
8.	Cycle Maternel 7
9.	Cycle Maternel 8
10.	Cycle Maternel 9
11.	Cycle Maternel 10
12.	Cycle Maternel 11
13.	Cycle Maternel 12
14.	Cycle Académique 1
15.	Cycle Académique 2
16.	Cycle Académique 3
17.	Cycle Académique 4
18.	Cycle Académique 5
19.	Cycle Académique 6
20.	Cycle Académique 7
21.	Cycle Académique 8
22.	Cycle Académique 9
23.	Cycle Académique 10
24.	Cycle Académique 11
25.	Cycle Académique 12
26.	Cycle Universitaire 1
27.	Cycle Universitaire 2
28.	Cycle Universitaire 3
29.	Cycle Universitaire 4
30.	Cycle Universitaire 5
31.	Cycle Universitaire 6

Activites

Libele	Discription
Activite Scolaire 1	Description de l activite scolaire 1
Activite Scolaire 2	Description de l activite scolaire 2
Activite Scolaire 3	Description de l activite scolaire 3
Activite Scolaire 4	Description de l activite scolaire 4
Activite Scolaire 5	Description de l activite scolaire 5
Activite Scolaire 6	Description de l activite scolaire 6
Activite Scolaire 7	Description de l activite scolaire 7
Activite Scolaire 8	Description de l activite scolaire 8
Activite Scolaire 9	Description de l activite scolaire 9
Activite Scolaire 10	Description de l activite scolaire 10
Activite Scolaire 11	Description de l activite scolaire 11
Activite Scolaire 12	Description de l activite scolaire 12
Activite Scolaire 13	Description de l activite scolaire 13
Activite Scolaire 14	Description de l activite scolaire 14
Activite Scolaire 15	Description de l activite scolaire 15
Activite Scolaire 16	Description de l activite scolaire 16
Activite Scolaire 17	Description de l activite scolaire 17
Activite Scolaire 18	Description de l activite scolaire 18
Activite Scolaire 19	Description de l activite scolaire 19
Activite Scolaire 20	Description de l activite scolaire 20
Activite Scolaire 21	Description de l activite scolaire 21
Activite Scolaire 22	Description de l activite scolaire 22
Activite Scolaire 23	Description de l activite scolaire 23
Activite Scolaire 24	Description de l activite scolaire 24
Activite Scolaire 25	Description de l activite scolaire 25
Activite Scolaire 26	Description de l activite scolaire 26
Activite Scolaire 27	Description de l activite scolaire 27
Activite Scolaire 28	Description de l activite scolaire 28
Activite Scolaire 29	Description de l activite scolaire 29



State Management System

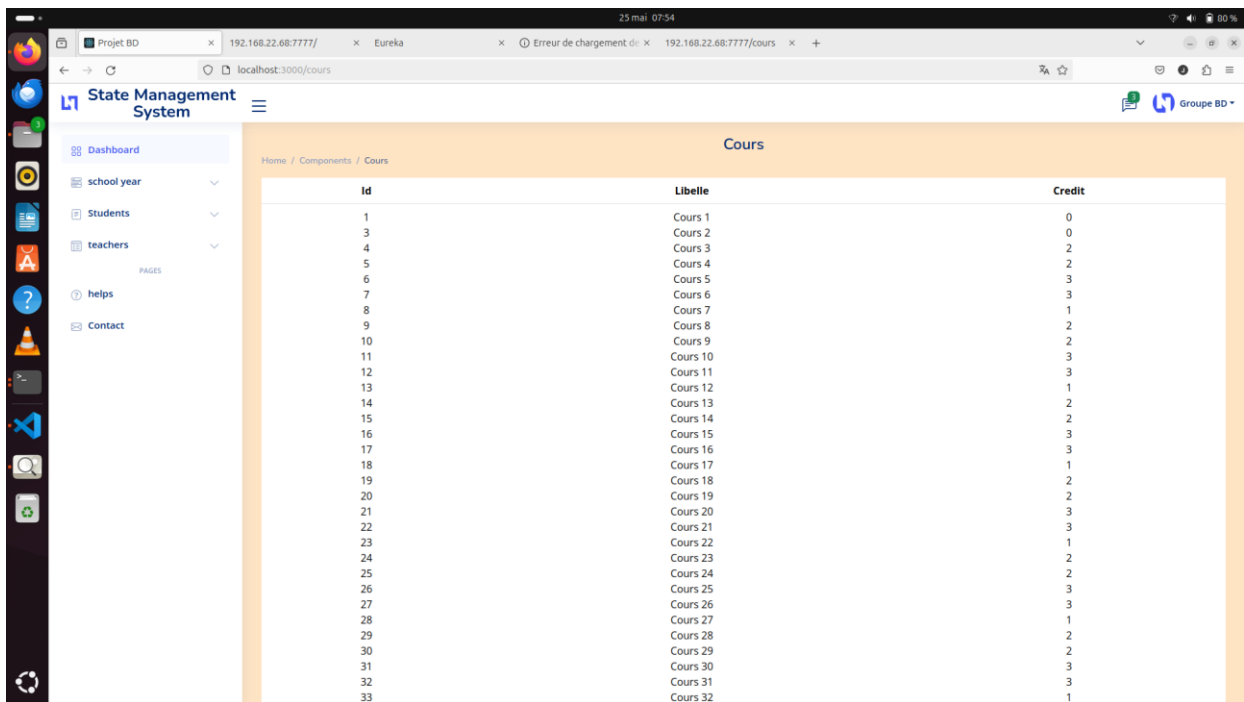
121. Cycle Général 12

Sequences

1.	TP 1
2.	Contrôle Continu 1
3.	Séquence 1
4.	TP 2
5.	Contrôle Continu 2
6.	Séquence 2
7.	TP 3
8.	Contrôle Continu 3
9.	Séquence 3
10.	TP 4
11.	Contrôle Continu 4
12.	Séquence 4
13.	TP 5
14.	Contrôle Continu 5
15.	Séquence 5
16.	TP 6
17.	Contrôle Continu 6
18.	Séquence 6
19.	TP 7
20.	Contrôle Continu 7
21.	Séquence 7
22.	TP 8
23.	Contrôle Continu 8
24.	Séquence 8
25.	TP 9
26.	Contrôle Continu 9
27.	Séquence 9
28.	TP 10
29.	Contrôle Continu 10
30.	Séquence 10

Activites

Activite Scolaire 45	Description de l activite scolaire 45
Activite Scolaire 46	Description de l activite scolaire 46
Activite Scolaire 47	Description de l activite scolaire 47
Activite Scolaire 48	Description de l activite scolaire 48
Activite Scolaire 49	Description de l activite scolaire 49
Activite Scolaire 50	Description de l activite scolaire 50
Activite Scolaire 51	Description de l activite scolaire 51
Activite Scolaire 52	Description de l activite scolaire 52
Activite Scolaire 53	Description de l activite scolaire 53
Activite Scolaire 54	Description de l activite scolaire 54
Activite Scolaire 55	Description de l activite scolaire 55
Activite Scolaire 56	Description de l activite scolaire 56
Activite Scolaire 57	Description de l activite scolaire 57
Activite Scolaire 58	Description de l activite scolaire 58
Activite Scolaire 59	Description de l activite scolaire 59
Activite Scolaire 60	Description de l activite scolaire 60
Activite Scolaire 61	Description de l activite scolaire 61
Activite Scolaire 62	Description de l activite scolaire 62
Activite Scolaire 63	Description de l activite scolaire 63
Activite Scolaire 64	Description de l activite scolaire 64
Activite Scolaire 65	Description de l activite scolaire 65
Activite Scolaire 66	Description de l activite scolaire 66
Activite Scolaire 67	Description de l activite scolaire 67
Activite Scolaire 68	Description de l activite scolaire 68
Activite Scolaire 69	Description de l activite scolaire 69
Activite Scolaire 70	Description de l activite scolaire 70
Activite Scolaire 71	Description de l activite scolaire 71
Activite Scolaire 72	Description de l activite scolaire 72
Activite Scolaire 73	Description de l activite scolaire 73
Activite Scolaire 74	Description de l activite scolaire 74

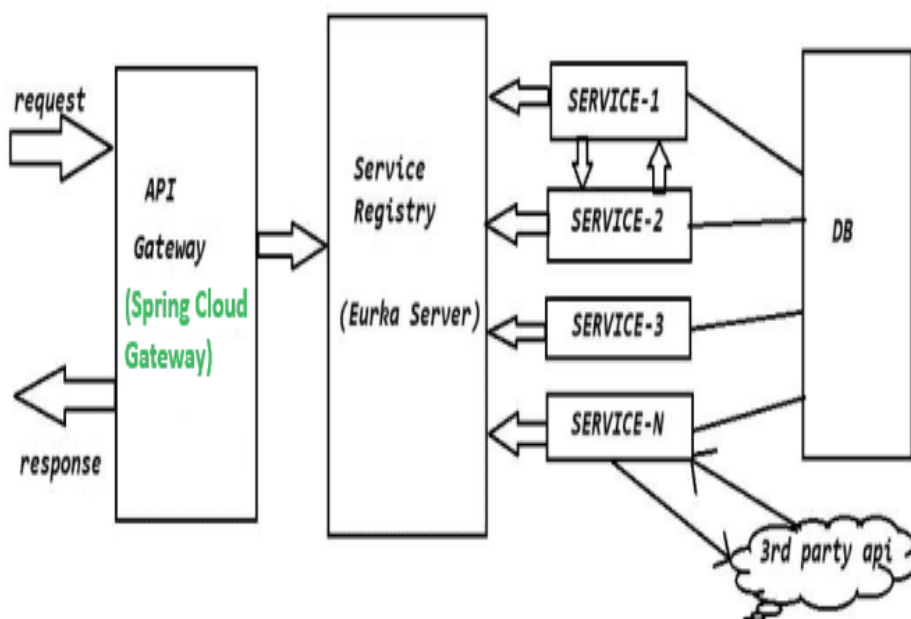


Id	Libelle	Credit
1	Cours 1	0
3	Cours 2	0
4	Cours 3	2
5	Cours 4	2
6	Cours 5	3
7	Cours 6	3
8	Cours 7	1
9	Cours 8	2
10	Cours 9	2
11	Cours 10	3
12	Cours 11	3
13	Cours 12	1
14	Cours 13	2
15	Cours 14	2
16	Cours 15	3
17	Cours 16	3
18	Cours 17	1
19	Cours 18	2
20	Cours 19	2
21	Cours 20	3
22	Cours 21	3
23	Cours 22	1
24	Cours 23	2
25	Cours 24	2
26	Cours 25	3
27	Cours 26	3
28	Cours 27	1
29	Cours 28	2
30	Cours 29	2
31	Cours 30	3
32	Cours 31	3
33	Cours 32	1

Chapitre 2 : Back-end

1. Objectif

Le back-end de notre projet repose sur une architecture basée sur Spring Boot, avec une configuration de Spring Cloud pour la production d'une API de micro-services. Ce back-end utilise à la fois MongoDB et PostgreSQL comme bases de données, associées à des serveurs de découverte de services Eureka, un serveur de traçage Zipkin, une Gateway et un système d'authentification pour accéder aux points de terminaison des micro-services. Ainsi, nous décrirons en détail le fonctionnement des micro-services, le rôle de chaque serveur et la communication entre les micro-services, les serveurs et les bases de données. Notre objectif est de fournir une plateforme flexible et performante pour la gestion et le traitement des données, en utilisant des technologies modernes et des pratiques recommandées qui offre une base solide pour le développement de micro-services en Java. En intégrant Spring Cloud, nous étendons les capacités de Spring Boot pour la création et la gestion de systèmes distribués complexes. Cette approche nous permet de développer et de déployer efficacement des micro-services indépendants, tout en assurant une communication transparente entre eux comme suit :



2. Technologies Utilisées

Spring Boot

Spring Boot est utilisé comme Framework principal pour développer nos micro-services, offrant une configuration simple et une approche conventionnelle pour démarrer rapidement.

Spring Cloud


Spring Cloud facilite la création de systèmes distribués complexes en fournissant des outils pour le déploiement et la gestion de micro-services.

MongoDB et PostgreSQL

Nous utilisons MongoDB et PostgreSQL comme bases de données pour stocker les données nécessaires à notre application, en choisissant MongoDB pour sa flexibilité dans le stockage de données non structurées et PostgreSQL pour sa robustesse dans la gestion des données relationnelles. Cette décision a été prise en fonction des besoins spécifiques de notre application, en utilisant MongoDB pour stocker des données non structurées nécessitant une flexibilité de schéma, tandis que PostgreSQL est utilisé pour des données plus structurées nécessitant une intégrité et une cohérence élevées.

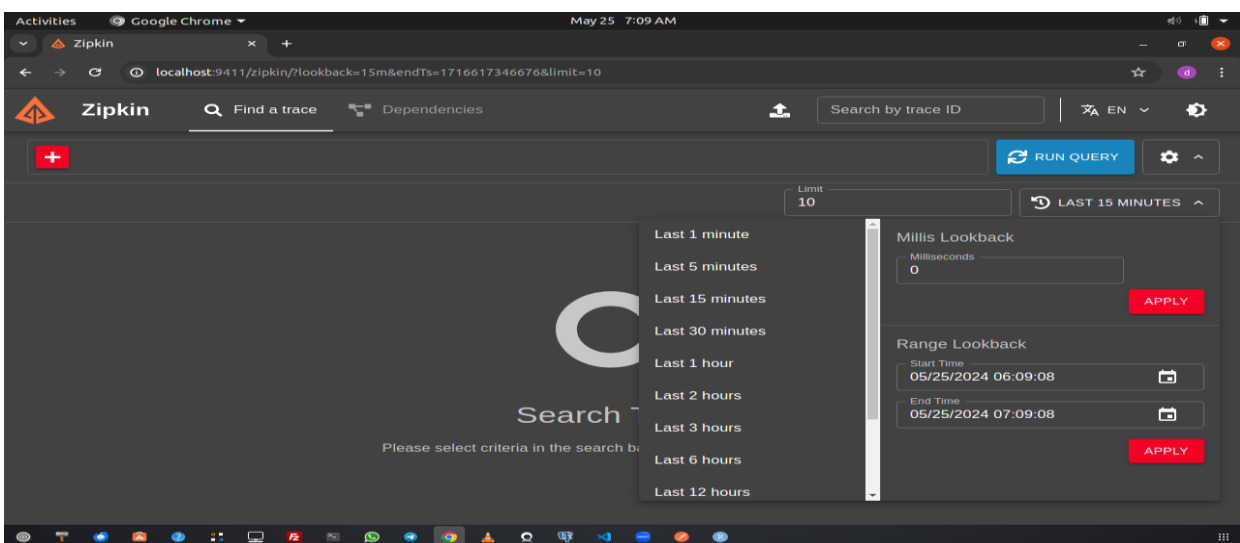
Eureka

Eureka est utilisé comme serveur de découverte de services, permettant à chaque micro-service de s'enregistrer et de découvrir les autres services disponibles dans l'écosystème pour garantir la haute disponibilité et la résilience de notre système, nous avons mis en place un serveur de découverte de services Eureka. Ce serveur permet à chaque micro-service de s'enregistrer et de découvrir dynamiquement les autres services disponibles dans l'écosystème, facilitant ainsi la gestion des dépendances et des mises à jour.

 HOME LAST 1000 SINCE STARTUP			
System Status			
Environment	test	Current time	2024-05-25T07:07:52 +0100
Data center	default	Uptime	00:04
		Lease expiration enabled	false
		Renews threshold	3
		Renews (last min)	2
DS Replicas			
Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
GATEWAY	n/a (1)	(1)	UP (1) - LAPTOP-RC43MN0T.gateway.7777
General Info			
Name	Value		
total-avail-memory	80mb		

Zipkin

Zipkin est un serveur de traçage utilisé pour suivre et visualiser les requêtes à travers différents micro-services, facilitant le débogage et l'optimisation des performances. Pour le suivi et la visualisation des requêtes à travers nos micro-services, nous avons intégré Zipkin, un serveur de traçage distribué. Zipkin nous permet d'identifier les goulots d'étranglement et d'optimiser les performances de notre système en analysant les temps de réponse et les dépendances entre les services.



🚦 Gateway

La Gateway agit comme un point d'entrée unique pour toutes les requêtes client, dirigeant le trafic vers les micro-services appropriés en fonction des URL et des règles prédéfinies. En tant que point d'entrée unique pour toutes les requêtes client, notre Gateway agit comme un proxy inverse, dirigeant le trafic vers les micro-services appropriés en fonction des règles de routage prédéfinies. Cette approche simplifie la gestion du trafic et renforce la sécurité en centralisant l'authentification et l'autorisation des utilisateurs.

🚦 Système d'Authentification

Un système d'authentification est mis en place pour sécuriser les points de terminaison des micro-services, garantissant que seuls les utilisateurs autorisés peuvent accéder aux données sensibles. Pour garantir la sécurité de notre système, nous avons mis en place un système d'authentification qui protège les points de terminaison des micro-services contre les accès non autorisés. Ce système permet de vérifier les identités des utilisateurs et de garantir que seuls les utilisateurs authentifiés et autorisés peuvent accéder aux données sensibles de notre application.

3. Configuration

a. Configuration des Micro-services

Chaque micro-service est configuré pour s'enregistrer auprès du serveur Eureka lors de son démarrage, permettant ainsi une découverte dynamique des services.

b. Configuration des Bases de Données

Les configurations pour se connecter à MongoDB et PostgreSQL sont définies dans les fichiers de propriétés de chaque micro-service, assurant une gestion centralisée des paramètres de connexion.

c. Configuration de la Gateway

La Gateway est configurée pour rediriger les requêtes HTTP entrantes vers les micro-services appropriés en fonction des règles de routage prédéfinies.

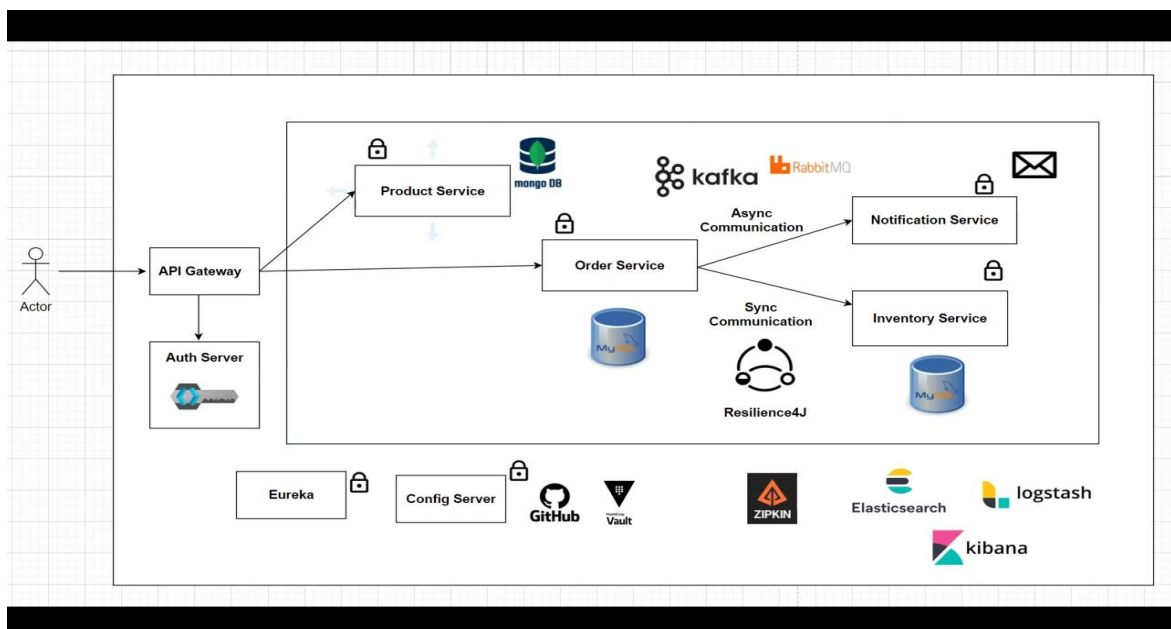
d. Configuration de l'Authentification

Le système d'authentification est configuré pour intercepter les requêtes vers les points de terminaison des micro-services et vérifier les jetons d'authentification avant de permettre l'accès.

4. Communication entre les Composants

a) Communication entre les Micro-services

Les micro-services communiquent entre eux via des requêtes HTTP RESTful, en utilisant les URL fournies par le serveur Eureka pour découvrir et atteindre les services nécessaires.



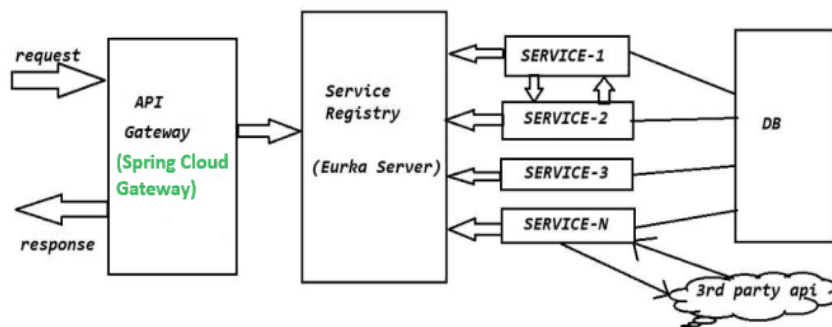
b) Communication avec les Bases de Données

Chaque micro-service communique avec sa base de données respective en utilisant les bibliothèques de connexion appropriées, garantissant l'accès sécurisé et efficace aux données.



c) Communication avec la Gateway

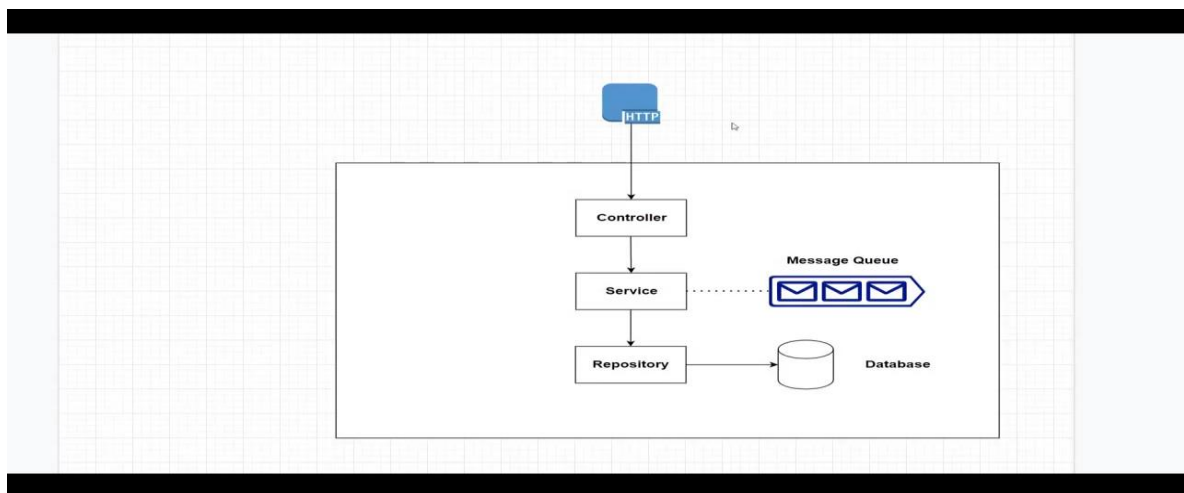
La Gateway communique avec les micro-services en agissant comme un proxy inverse, redirigeant le trafic client vers les micro-services appropriés en fonction des règles de routage définies.



5. Exposition via la Gateway

i.Exposition des Micro-services

Les micro-services sont exposés via la Gateway en définissant des points de terminaison RESTful accessibles aux clients externes via des requêtes http dont le déroulement est le suivant :



ii.Authentification via la Gateway

La Gateway intègre le système d'authentification pour sécuriser les points de terminaison des micro-services, nécessitant une authentification avant l'accès aux données.

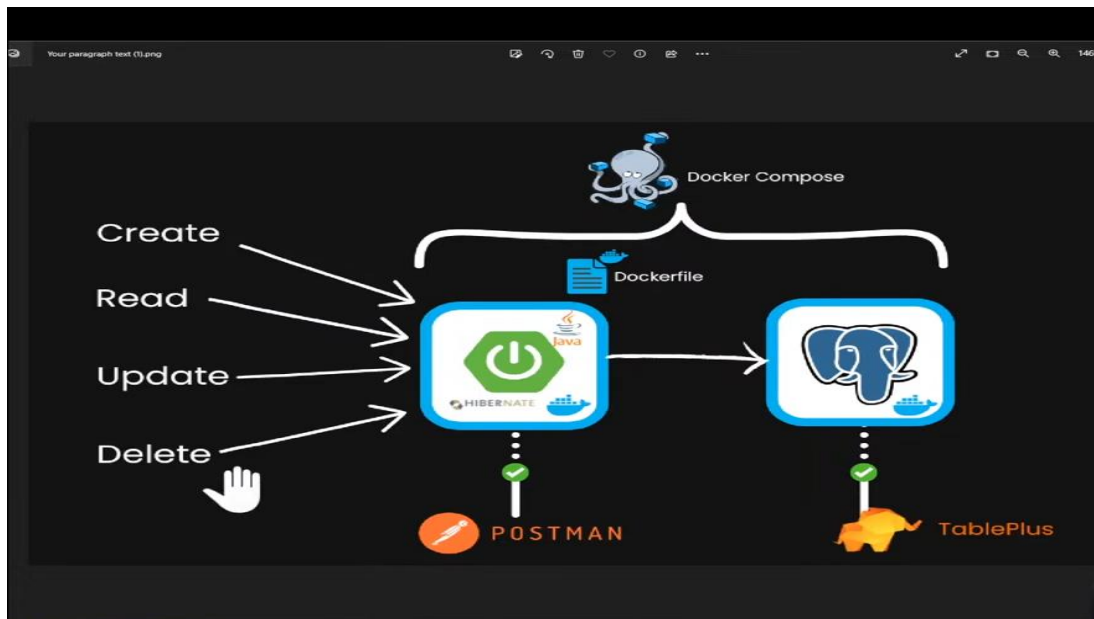
6. Fonctionnement avec Front-end React

➤ Connexion avec la Gateway

Le front-end React se connecte à la Gateway en utilisant les URL exposées pour chaque micro-service, envoyant des requêtes HTTP pour récupérer et manipuler les données nécessaires.

➤ Intégration de l'Authentification

Le front-end React intègre le système d'authentification en envoyant des jetons d'authentification avec chaque requête vers la Gateway, assurant ainsi un accès sécurisé aux données.



Annexe

Lien github: https://github.com/regent-ndongmo/Projet_BD