

TP KAFKA

Professeur référent : Hervé Tondeur



**Vauley Thomas - Pain Anthonin - Kaoukeb Adam -
Rouillard Antoine**

FISA 4A INFO

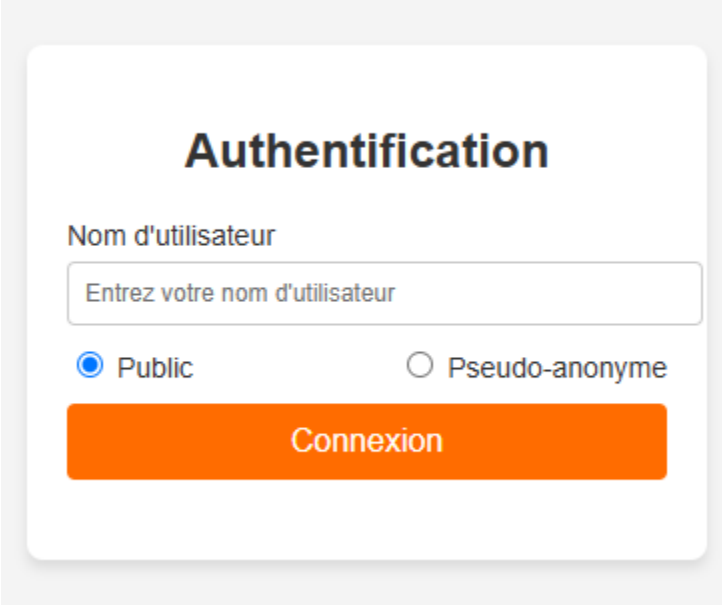
Description du fonctionnement

Les instructions pour exécuter le projet se situent dans le readme. La branche sur laquelle se placer est la branche “main”. En cas d’erreur lors de l’exécution, n’hésitez pas à nous contacter.

1- Connexion:

Lors de l’identification, nous pouvons sélectionner un mode “Public”, qui communiquera le fait que nous sommes connectés à tous les autres utilisateurs en ligne, ou alors un mode “Pseudo-Anonyme”, qui nous permet d’envoyer des messages tout en n’étant pas affiché comme “En ligne” pour les autres utilisateurs.

Ici, on peut utiliser les utilisateurs “Manuel”, “David” ou “Francis” pour se connecter.

A screenshot of a web form titled "Authentication". It features a text input field for the username, two radio buttons for selecting a connection mode ("Public" or "Pseudo-anonyme"), and an orange "Connexion" button.

Authentication

Nom d'utilisateur

Entrez votre nom d'utilisateur

☒ Public ☐ Pseudo-anonyme

Connexion

2- Communication:

Après s'être connecté, on arrive dans l'interface de communication, qui nous permet d'envoyer des messages, soit à un utilisateur précis, soit dans un topic global que tout le monde peut consulter.

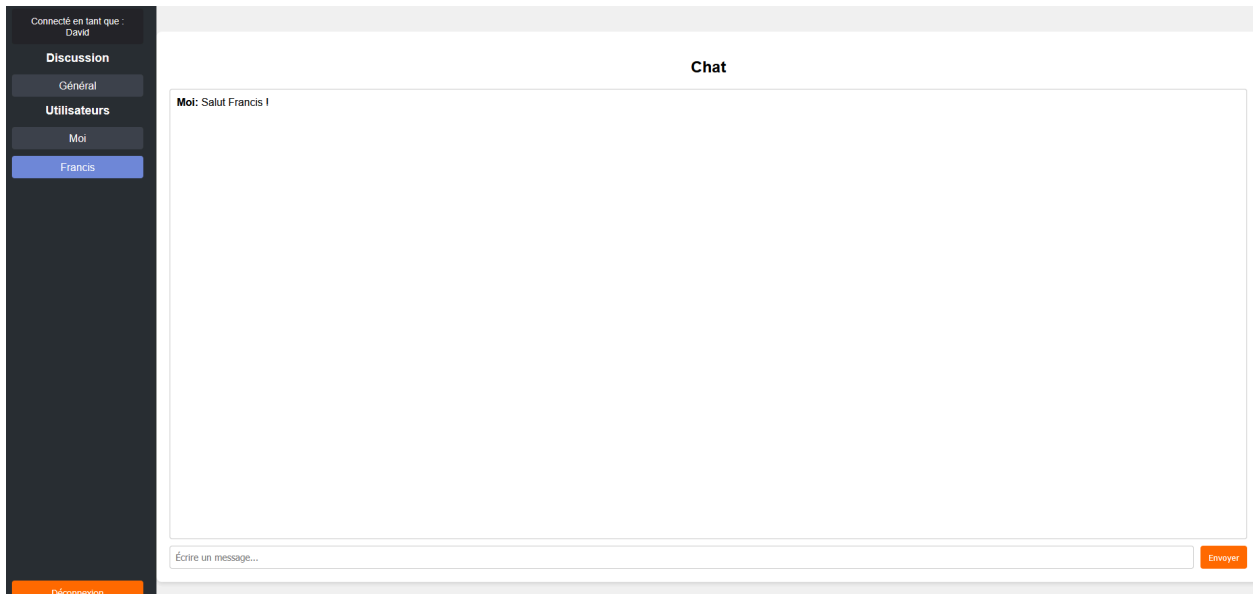
The image shows a web-based chat interface. On the left is a dark sidebar titled "Utilisateurs" (Users) containing two buttons: "David" and "Manuel". The main area is titled "Chat" and displays a list of messages in a scrollable container. The messages are: "David: ping", "Manuel: pong", "David: pong ?", and "David: aa". At the bottom of the chat area is a text input field with the placeholder "Écrire un message..." and an orange "Envoyer" (Send) button.

Utilisateurs
David
Manuel

Chat
David: ping
Manuel: pong
David: pong ?
David: aa

Écrire un message... Envoyer

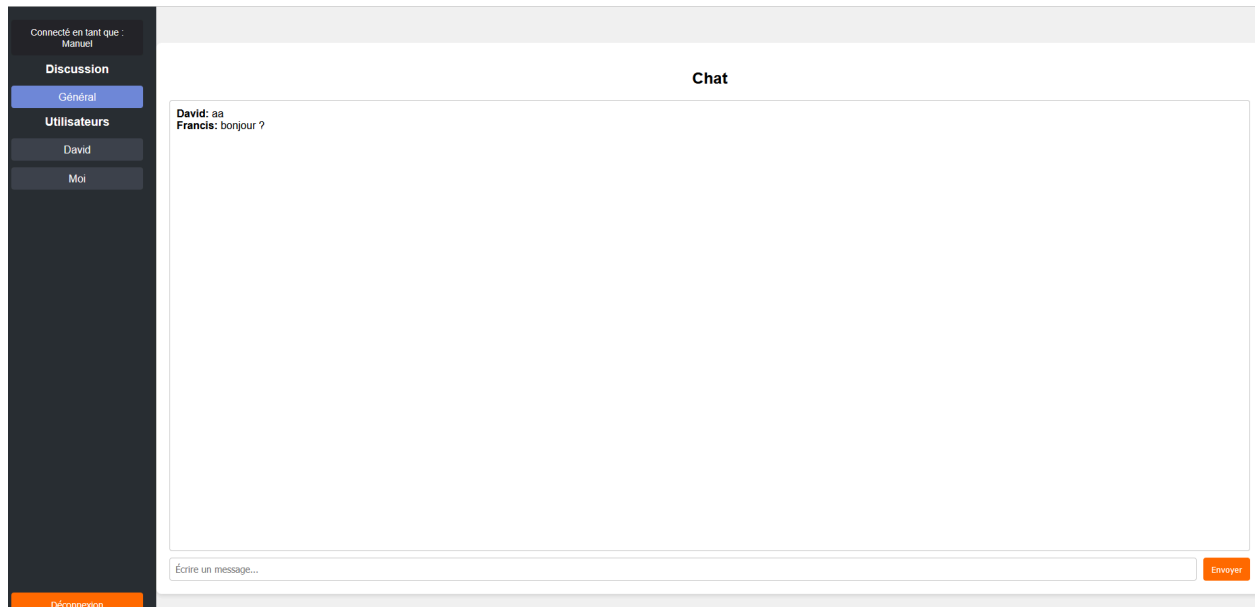
2- 1: Communication privée



En cliquant sur les différents noms qui apparaissent sur le côté gauche de l'écran, il est possible d'ouvrir des chats privés avec les différents utilisateurs connectés. Le chat est enregistré et l'historique reste affiché à l'écran.

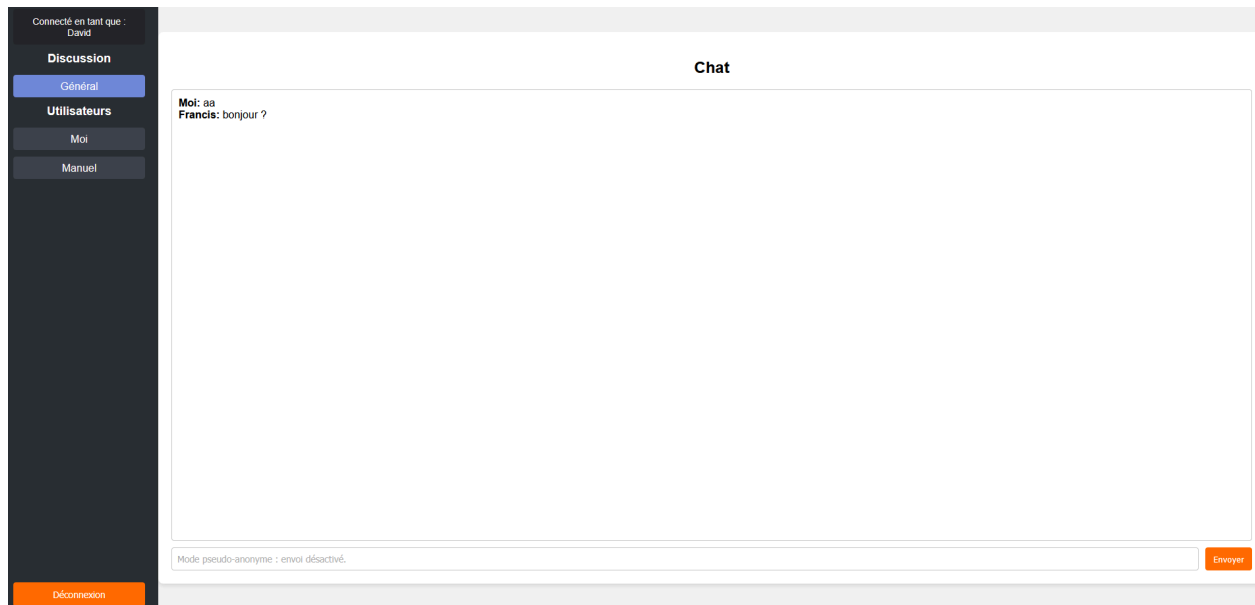
Nous avons utilisé un kafka consumer dynamic, de sorte à ce que chaque utilisateur se voit attribuer un topic privé sur lequel il écoute, dans notre cas chaque topic privé a le nom de l'utilisateur qui l'écoute.

2- 2: Communication globale



Concernant la communication via un topic global, nous avons simplement utilisé l'objet spring `@KafkaListener`, qui nous permet d'écouter sur un topic en particulier. L'objet java contenant ce `@KafkaListener` est simplement instancié lors de la création de chaque utilisateur, ce qui fait que chacun des clients écoute sur ce topic.

3: Mode pseudo-anonyme

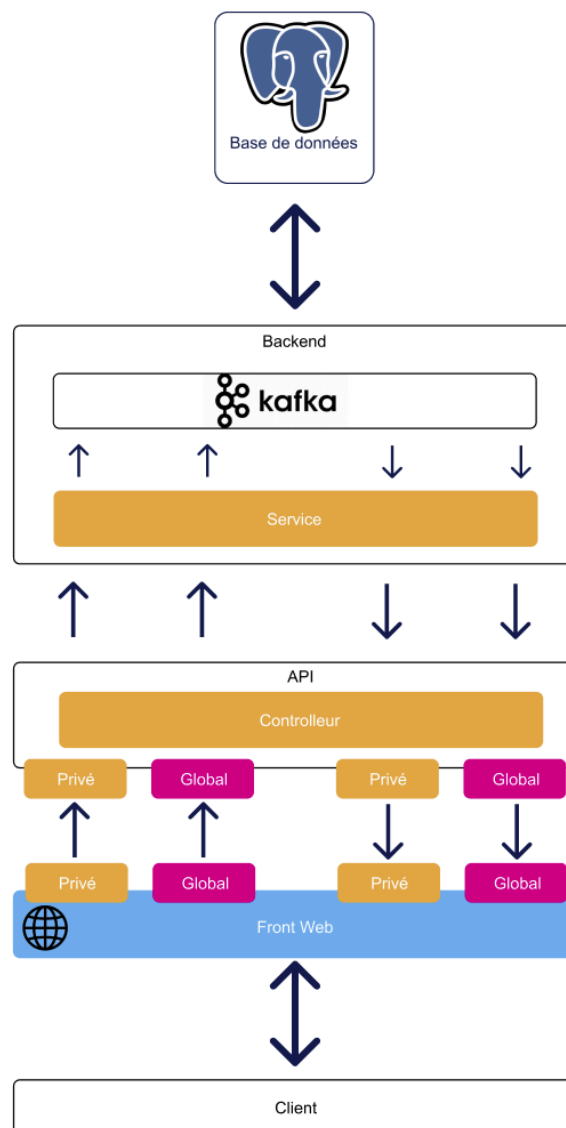


Dans le mode pseudo-anonyme, l'utilisateur a accès aux différents chats en lecture seule, et peut recevoir des messages qui ont été envoyés sur le topic global. Il n'apparaît pas en tant qu'utilisateur connecté pour les autres clients, il ne peut donc pas recevoir de message privé, et l'envoi de message est bloqué de son côté.

Questions

Question N° 1 :

Proposer une architecture de service qui permet de répondre à ce besoin décrit ci-dessus. Faire un schéma de cette architecture. Cette architecture doit montrer clairement l'ingénierie mise en place pour permettre ce mode de fonctionnement décrit ci-dessus. Une explication des points complexes sera la bienvenue.



Question N°3 :

Comment avez-vous résolu la dualité entre l'écoute des channels et la disponibilité du prompt au niveau du client ?

Nous avons fait en sorte de ne pas actualiser toute la page à chaque message reçu. A la place, pour limiter la charge, nous réalisons un fetch permettant la récupération de tous les messages nécessaires sur une partie de la page affichant les messages toutes les deux secondes, cela évite que l' input pour rédiger un message se retrouve actualisée et empêche l'utilisateur de rédiger ses messages .

Question N°4 :

Comment avez-vous résolu le besoin de séparation des flux public / privé ?

Pour résoudre ce problème, nous avons mis en place un topic global (ou public) qui est écouté par tous les clients par défaut. Ensuite, pour chaque utilisateur, nous créons un topic portant leur nom, qui est accessible uniquement par les personnes sélectionnées.