

Tableau comparatif des bus de communication :

Bus de Communication	Description	Avantages	Inconvénients
Apache Kafka	Plateforme de streaming de données	Haute performance, persistance des données	Complexité pour les débutants, manque d'UI
Apache ActiveMQ	Broker de messages open-source	Robuste, divers protocoles de communication	Configuration parfois complexe, faible scalabilité
RabbitMQ	Broker de messages open-source	Facile à utiliser, supporte divers protocoles	Faible performance pour les très gros volumes
IBM MQ	Plateforme de messagerie d'entreprise	Grande fiabilité, sécurité avancée	Coût élevé, configuration initiale complexe
MuleSoft Anypoint	Plateforme d'intégration d'applications	Interface utilisateur intuitive, support étendu	Coût élevé, courbe d'apprentissage abrupte
Red Hat Fuse	Plateforme d'intégration open-source	Flexibilité, support de différents protocoles	Documentation parfois limitée, communauté moins étendue

2.1 :

Framework	Avantages	Inconvénients
ReactJS	Virtual DOM pour des mises à jour efficaces	Courbe d'apprentissage initiale pour certains développeurs
Angular	Supporté par Google, structure MVC claire	Courbe d'apprentissage abrupte, couramment plus complexe
Vue.js	Léger, facile à intégrer dans des projets	Moins de support et d'outils que React ou Angular
Svelte	Génération de code extrêmement optimisée	Relativement nouveau, communauté moins étendue

2.2 :

### 1. Connexion de l'utilisateur :

- **LoginForm** : Composant gérant le formulaire de connexion.

- **Authentication** : Gestion de l'authentification, vérification des identifiants.
- 2. **Écran d'accueil** :
  - **Home** : Composant pour afficher l'écran d'accueil avec les informations pertinentes.
- 3. **Achat de cartes depuis une liste** :
  - **CardList** : Affichage de la liste des cartes disponibles à l'achat.
  - **CardItem** : Composant individuel représentant chaque carte avec les détails et les options d'achat.
  - **PurchaseModal** : Modal pour confirmer l'achat de la carte sélectionnée.
- 4. **Vente de cartes depuis une liste** :
  - **UserCardList** : Affichage de la liste des cartes que l'utilisateur souhaite vendre.
  - **UserCardItem** : Composant individuel représentant chaque carte à vendre avec les détails et les options de vente.
  - **SellModal** : Modal pour confirmer la vente de la carte sélectionnée.
- 5. **Composants communs** :
  - **Header** : Barre de navigation, en-tête de l'application.
  - **Footer** : Pied de page de l'application.
  - **Loader** : Indicateur de chargement pour les opérations asynchrones.

Ce découpage vous permettrait d'organiser efficacement votre code en composants réutilisables, chaque composant étant responsable d'une partie spécifique de l'interface utilisateur. Vous pourrez également utiliser des bibliothèques tierces pour gérer l'état global, la gestion des routes (si nécessaire) et d'autres fonctionnalités avancées en fonction des besoins spécifiques de votre application.

## QUESTIONS :

Qu'est ce que le CROSS ORIGIN ? En quoi cela est-il dangereux ?

Le Cross-Origin Resource Sharing (CORS) est une politique de sécurité pour les navigateurs web qui contrôle les requêtes entre différentes origines (domains). Cela permet de déterminer si une ressource sur un serveur peut être légitimement demandée à partir d'une autre origine que celle qui l'a servie initialement.

Pourquoi c'est important :

- **Sécurité** : Évite les attaques telles que les Cross-Site Request Forgery (CSRF) en empêchant l'accès non autorisé à certaines ressources.
- **Confidentialité** : Protège les données sensibles en limitant l'accès à des domaines spécifiques.
- **Intégrité des données** : Préviend la modification non autorisée de ressources par des requêtes provenant de sources non autorisées.

Comment REACTJS fait-il pour afficher rapidement les modifications sur les composants ?

React utilise le Virtual DOM pour rendre rapidement les modifications sur les composants. Plutôt que de mettre à jour directement le DOM, React crée une version virtuelle légère de

celui-ci. Lorsqu'un composant change, React compare cette version virtuelle avec la précédente pour déterminer les différences. En se basant sur ces différences, React met à jour sélectivement les parties spécifiques du DOM réel qui ont changé, minimisant ainsi les manipulations coûteuses du DOM et rendant le processus de rendu plus rapide et efficace.

Quelle est la fonction essentielle de REACTJS ?

ReactJS simplifie la création d'interfaces utilisateur interactives en utilisant des composants réutilisables. Il optimise les mises à jour de l'interface en utilisant le Virtual DOM, gère efficacement l'état des composants et adopte une approche déclarative pour décrire comment l'interface doit se comporter en fonction de l'état.

Quelle est la fonction essentielle de FLUX ?

Flux est un modèle de conception de données unidirectionnel où les données ont un seul flux de modification. Il aide à maintenir un état global cohérent pour toute l'application en utilisant un cycle de données unidirectionnel, ce qui simplifie la gestion de l'état et facilite le suivi des modifications apportées aux données.

Qu'est ce que REDUX ?

Redux fournit un conteneur global pour l'état de l'application, rendant la gestion de l'état plus prévisible et plus facile à suivre. Il utilise un modèle de données unidirectionnel, où l'état global est stocké dans un seul "store". Les modifications de l'état se font par des actions, des objets qui décrivent ce qui doit être modifié. Ces actions sont gérées par des "reducers", des fonctions pures qui spécifient comment l'état change en réponse à une action. Redux facilite ainsi la gestion de l'état d'une application, en améliorant la prévisibilité, la traçabilité et la cohérence de l'état global.

Qu'est ce que JMS ? Est-ce spécifique à Springboot ?

JMS est une API Java pour la messagerie asynchrone entre applications, et bien que ce ne soit pas spécifique à Spring Boot, Spring Boot facilite l'intégration et l'utilisation de JMS dans des applications Java en fournissant des outils et des abstractions pour travailler avec des systèmes de messagerie.

Quelles sont les différents modes de transmissions de JMS ?

- **Point-to-Point (P2P)** : Messages envoyés à une file d'attente, consommés par un seul destinataire.
- **Publish/Subscribe (Pub/Sub)** : Messages envoyés à un sujet, reçus par tous les abonnés connectés au sujet.

Quel est le mode de transmission activé par défaut dans activeMq ?

Apache ActiveMQ utilise le mode de transmission Point-to-Point (P2P) avec des files d'attente (Queues). Ce mode assure qu'un seul consommateur reçoit chaque message envoyé à la file d'attente, garantissant ainsi une livraison unique des messages à un destinataire à la fois. ActiveMQ prend également en charge le mode Publish/Subscribe

(Pub/Sub) en utilisant des sujets (Topics), mais le mode P2P est activé par défaut lors de la création de destinations sans spécification explicite du type.

Qu'est ce que activeMq ?

Apache ActiveMQ est un broker de messagerie open-source qui facilite la communication entre applications, systèmes ou composants via des messages asynchrones. Il offre différents modes de transmission, comme le Point-to-Point (P2P) et le Publish/Subscribe (Pub/Sub), garantissant la fiabilité et la persistance des messages tout en offrant une évolutivité pour répondre aux besoins de communication des applications distribuées.

Quels avantages proposent un bus de communication vis-à-vis de requêtes http classiques ?

1. **Communication décorrélée** : Permet une communication asynchrone entre les composants sans dépendances directes.
2. **Flexibilité et évolutivité** : Facilite l'évolution et la modification des services sans affecter l'ensemble du système.
3. **Gestion centralisée** : Fournit un point central de gestion des messages et des communications.
4. **Fiabilité accrue** : Offre des mécanismes de fiabilité et de persistance pour assurer la livraison des messages, même en cas de pannes temporaires.
5. **Support de divers protocoles** : Permet l'utilisation de différents protocoles de communication, pas seulement HTTP, pour répondre aux besoins spécifiques des services et des applications.

Comment faire pour partager des modules Maven avec un partenaire extérieur ?

Pour partager des modules Maven avec un partenaire extérieur :

1. Utilisez un gestionnaire de référentiel Maven (comme Nexus, Artifactory) pour publier les artefacts dans un référentiel partagé sécurisé.
2. Configurez l'accès distant pour votre partenaire afin qu'il puisse télécharger les artefacts depuis ce référentiel.
3. Si nécessaire, partagez directement les artefacts via des méthodes sécurisées comme FTP, SFTP ou un autre canal sécurisé.

Comment faire pour exporter un composant REACTJS ?

Pour exporter un composant ReactJS :

1. Créez votre composant en React.
2. Utilisez l'instruction `export` pour rendre votre composant disponible pour être importé ailleurs dans votre application ou dans d'autres projets.
3. Utilisez `import` dans d'autres fichiers pour utiliser ce composant exporté.

Quel est le pré-requis pour Springboot afin de pouvoir convertir automatiquement le message reçu dans un bus de communication en objet ?

Pour que Spring Boot convertisse automatiquement un message reçu d'un bus de communication en objet, vous devez :

1. Ajouter les dépendances appropriées dans votre fichier `pom.xml` pour la gestion de la conversion des messages.
2. Utiliser des convertisseurs adaptés pour la sérialisation et la désérialisation des messages en objets Java.

Comment est réalisée la traduction des messages reçus (bus de communication ou request http) en objet Java ? Quelles sont les prérequis ? Est-ce toujours possible ?

La traduction des messages reçus en objets Java est réalisée à l'aide de convertisseurs de données comme Jackson pour JSON ou JAXB pour XML. Cela nécessite des formats de données clairs et compatibles avec les convertisseurs disponibles. La faisabilité dépend du format et de la structure du message, ainsi que des outils de conversion utilisés dans l'application.

Quelles sont les fonctionnalités des différentes annotations en Springboot ?:

- `@EnableJpaRepositories`
- `@EntityScan`
- `@ComponentScan`

Les annotations de configuration en Spring Boot ont les fonctions suivantes :

- **`@EnableJpaRepositories`** : Active et configure les interfaces de dépôt JPA automatiquement.
- **`@EntityScan`** : Indique à Spring où trouver les classes d'entités JPA pour la persistance.
- **`@ComponentScan`** : Recherche et enregistre les composants annotés tels que les beans et les services dans le contexte de l'application, selon le package spécifié.