

Document de mi-parcours : Organiz'Asso

Yuxiang ZHANG

Antoine LECOMTE

Graphe des dépendances des composants React

Le graphe suivant montre les relations entre les composants React du projet, mettant en évidence les dépendances mutuelles et la hiérarchie des composants.

```
1 App (index.jsx)
2   MainPage.jsx
3     NavigationPanel.jsx
4       Login.jsx
5       Logout.jsx
6       Signup.jsx
7     Feed.jsx
8       SearchBar.jsx
9       MessageList.jsx
10        Message.jsx
11          ReplyList.jsx
12            Reply.jsx
13              EditMessage.jsx
14              DeleteMessage.jsx
15        AddMessage.jsx
16      UserProfile.jsx
17        UserInfo.jsx
18        UserSettings.jsx
19        UserMessageList.jsx
20          Message.jsx
21            ReplyList.jsx
22              Reply.jsx
23                EditMessage.jsx
24                DeleteMessage.jsx
25      SearchResults.jsx
26        MessageList.jsx
27          Message.jsx
28            ReplyList.jsx
29              Reply.jsx
30                EditMessage.jsx
31                DeleteMessage.jsx
32      AdminPanel.jsx
33        AdminMenu.jsx
34        UserValidation.jsx
35          ValidateUser.jsx
```

```

36      AdminFeed.jsx
37      MessageList.jsx
38      Message.jsx
39      ReplyList.jsx
40      Reply.jsx
41      EditMessage.jsx
42      DeleteMessage.jsx
43      AddMessage.jsx
44      api/messagesAPI.jsx
45      api/userAPI.jsx
46      api/serverConfig.jsx
47      context/UserContext.jsx

```

Listing 1: Structure des composants React

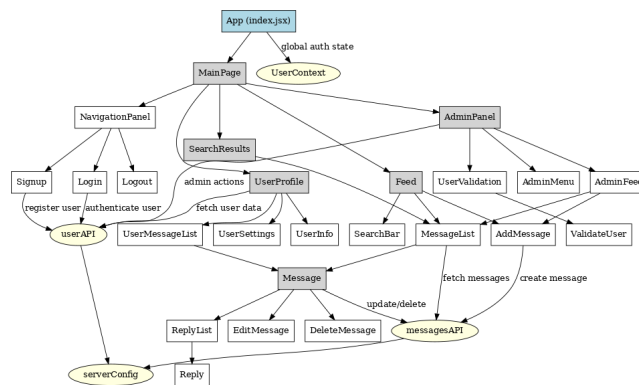


Figure 1: Diagramme UML des composants

Liste des composants et leur description

App

Fonction : Composant principal gérant la navigation et l'état global.

Props : Aucune directe.

State : Utilisateur connecté, statut administrateur.

Composants inclus : MainPage.

MainPage

Fonction : Page principale affichant la navigation et le contenu central.

Props : Aucune directe.

State : Aucun.

Composants inclus : NavigationPanel, Feed, UserProfile, SearchResults, AdminPanel.

NavigationPanel

Fonction : Gère l'authentification des utilisateurs.

Props : Aucune.

State : Aucun.

Composants inclus : Login, Logout, Signup.

Login

Fonction : Permet à un utilisateur de se connecter.

Props : onLoginSuccess(user) – Fonction appelée après une connexion réussie.

State : email (string), password (string), loading (bool).

Composants inclus : Aucun.

Logout

Fonction : Permet à un utilisateur de se déconnecter.

Props : onLogout() – Fonction déclenchée à la déconnexion.

State : Aucun.

Composants inclus : Aucun.

Signup

Fonction : Permet à un utilisateur de créer un compte.

Props : onSignupSuccess(user) – Fonction appelée après l'inscription.

State : email (string), password (string), confirmPassword (string), loading (bool).

Composants inclus : Aucun.

Feed

Fonction : Affiche les messages du forum.

Props : Aucun.

State : Liste des messages.

Composants inclus : SearchBar, MessageList, AddMessage.

SearchBar

Fonction : Permet d'effectuer une recherche dans les messages.

Props : onSearch(query) – Fonction appelée lors d'une recherche.

State : query (string) – Terme recherché.

Composants inclus : Aucun.

MessageList

Fonction : Liste des messages affichés sur le forum.

Props : Liste des messages.

State : Aucun.

Composants inclus : Message.

Message

Fonction : Affichage d'un message avec ses réponses.

Props : Contenu, auteur, date.

State : Aucun.

Composants inclus : ReplyList, EditMessage, DeleteMessage.

ReplyList

Fonction : Liste des réponses à un message.

Props : Liste des réponses.

State : Aucun.

Composants inclus : Reply.

Reply

Fonction : Affichage d'une réponse.

Props : Contenu, auteur, date.

State : Aucun.

Composants inclus : Aucun.

EditMessage

Fonction : Modification d'un message existant dans la discussion.

Props : messageId, messageContent.

State : newMessageContent.

Composants inclus : MessageEditor.

DeleteMessage

Fonction : Suppression d'un message spécifique de la discussion.

Props : messageId.

State : Aucun.

Composants inclus : Aucun.

AddMessage

Fonction : Ajout d'un nouveau message dans la discussion.

Props : Aucun.

State : newMessageContent.
Composants inclus : MessageEditor.

UserProfile

Fonction : Affichage du profil utilisateur.
Props : Aucune.
State : Aucun.
Composants inclus : UserInfo, UserSettings, UserMessageList.

UserInfo

Fonction : Affiche les informations d'un utilisateur.
Props : user (objet) – Données de l'utilisateur.
State : Aucun.
Composants inclus : Aucun.

UserSettings

Fonction : Permet à un utilisateur de modifier ses paramètres (email, mot de passe, préférences).
Props : user (objet) – Informations de l'utilisateur actuel.
State : email (string), password (string), notifications (bool).
Composants inclus : Aucun.

UserMessageList

Fonction : Affiche les messages publiés par un utilisateur.
Props : userId (string) – Identifiant de l'utilisateur.
State : messages (array) – Liste des messages.
Composants inclus : Message.

SearchResults

Fonction : Affiche les résultats d'une recherche de messages.
Props : results (array) – Liste des messages trouvés.
State : Aucun.
Composants inclus : MessageList.

AdminPanel

Fonction : Gère les actions administratives.
Props : Aucune.
State : Aucun.
Composants inclus : AdminMenu, UserValidation, AdminFeed.

AdminMenu

Fonction : Menu permettant d'accéder aux outils d'administration.

Props : Aucun.

State : selectedSection (string) – Section en cours d'affichage.

Composants inclus : Aucun.

UserValidation

Fonction : Gestion des inscriptions et validation des utilisateurs.

Props : Liste des utilisateurs en attente.

State : Aucun.

Composants inclus : ValidateUser.

ValidateUser

Fonction : Permet d'approuver ou rejeter les utilisateurs en attente.

Props : user (objet) – Informations de l'utilisateur à valider.

State : Aucun.

Composants inclus : Aucun.

AdminFeed

Fonction : Affichage des discussions du forum privé.

Props : Aucun.

State : Aucun.

Composants inclus : MessageList, AddMessage.

api/messagesAPI

Fonction : Gestion des requêtes liées aux messages (création, modification, suppression), utilisé dans Feed, MessageList, Message, AddMessage, EditMessage, DeleteMessage, et AdminFeed.

Props : Aucun.

State : Aucun.

Composants inclus : Aucun.

api/userAPI

Fonction : Gestion des données utilisateur (connexion, déconnexion, récupération des informations utilisateur), utilisé dans Login, Logout, Signup, UserProfile, UserSettings, et UserValidation.

Props : Aucun.

State : Aucun.

Composants inclus : Aucun.

api/serverConfig

Fonction : Configuration du serveur (paramètres généraux, connexion à la base de données), utilisé par messagesAPI.jsx et userAPI.jsx pour éviter de répéter l'URL de l'API dans chaque requête.

Props : Aucun.

State : Aucun.

Composants inclus : Aucun.

context/UserContext

Fonction : Fourniture de l'état de l'utilisateur (authentification, informations utilisateur) à travers l'application via le contexte, utilisé dans App.jsx pour gérer l'état global de l'authentification.

Props : Aucun.

State : user, isAuthenticated.

Composants inclus : UserProvider, useUserContext.