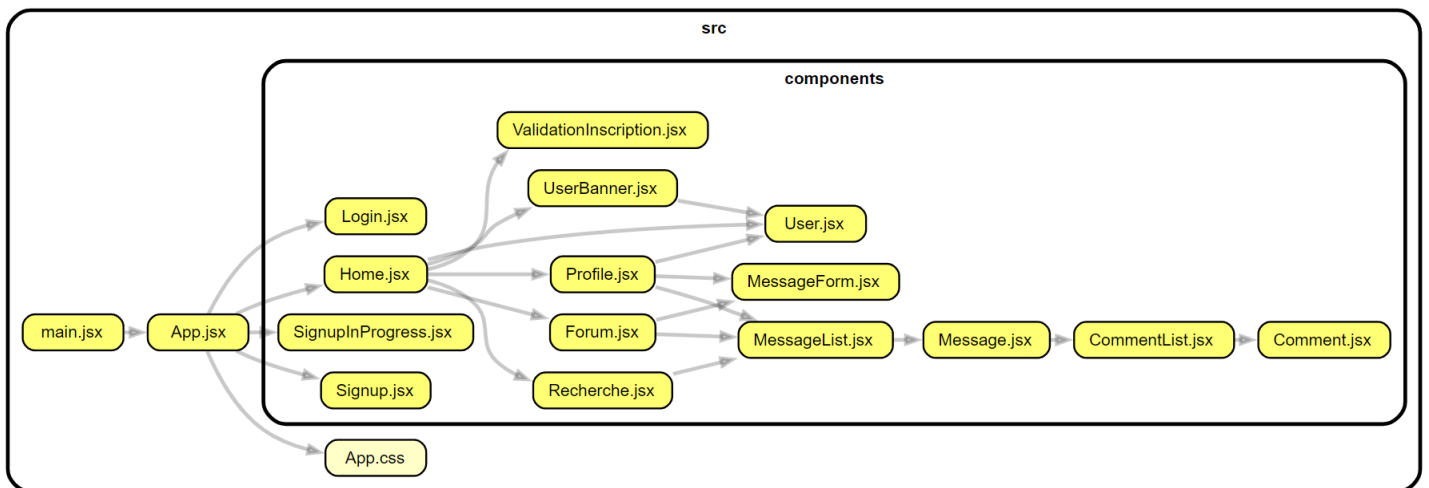


Document de mi-parcours - Groupe 2

PINHO FERNANDES Enzo - 21107465
PIRON-PALLISER Maximilien - 21107603
LU3IN017 (2023-2024)

Graphe de dépendances entre composants



Explication de chaque composant

App :

- Ce composant est le point d'entrée du site, permettant d'alternier entre plusieurs pages : la page de connexion, d'inscription, d'attente de validation et enfin la page principale (Home).
- Elle ne prend aucun props.
- Pour le moment, il contient deux états. L'un est un booléen permettant de voir si l'utilisateur est connecté (par défaut en false), et l'autre un string représentant chacun une page (par défaut "login").
- Il contient les composants : Home, Login, Signup, SignupInProgress.

Login :

- Ce composant permet à l'utilisateur de se connecter au site.
 - Il prend deux props : "login" qui est une fonction pour changer l'état de connexion de App, "signup_page" qui est une fonction permettant d'aller à la page d'inscription. Ces deux props seront toujours renseignés.
 - Pour le moment, il contient deux états. Un string pour représenter l'email que l'utilisateur inscrit sur le formulaire, et l'autre le mot de passe.
 - Il ne contient aucun composant.
-

Signup :

- Ce composant permet à l'utilisateur de s'inscrire au site.
 - Il prend deux props : "signupInProgress_page" qui est une fonction permettant d'aller à la page d'attente de validation, et "login_page" pour aller à la page de connexion. Ces deux props seront toujours renseignés.
 - Pour le moment, il contient cinq états, plus particulièrement cinq string pour stocker ce que l'utilisateur inscrit sur le formulaire : le prénom, nom, email, mot de passe et confirmation du mot de passe.
 - Il ne contient aucun composant.
-

SignupInProgress :

- Ce composant rappelle à l'utilisateur que son compte est en attente de validation.
 - Il prend deux props : "signup_page" qui est une fonction permettant d'aller à la page d'inscription, et "login_page" pour aller à la page de connexion. Ces deux props seront toujours renseignés.
 - Il ne contient aucun état.
 - Il ne contient aucun composant.
-

Home :

- Ce composant représente tout le reste du site, que ce soit le forum, les profils... Tout se passe là dedans.
 - Il prend un seul props : "logout" qui est une fonction pour se déconnecter et donc revenir à la page de connexion.
 - Pour le moment, il contient deux états : "isAdmin" (booléen) pour tester facilement le comportement d'un compte utilisateur ou administrateur, qui sera facilement remplacé par un composant User quand il sera développé, et "homeCBpage" (string) qui permet de changer la bannière centrale du site. En effet, cette bannière à elle seule aura plusieurs fonctions qu'on choisira : forum, profil, recherche...
 - Il contient les composants : Forum, Profile, Recherche, UserBanner, ValidationInscription, User.
-

ValidationInscription:

- Ce composant permet aux administrateurs de voir les inscriptions sur le site, et de les valider ou non.
 - Nous n'avons absolument pas travaillé sur ce côté, il reste que très théorique. Nous supposons qu'il y aura un composant "InscriptionList" et un composant "InscriptionWaiting".
-

Profile :

- Ce composant représente le profil d'un utilisateur.
 - Nous supposant qu'il aura au moins un props pour représenter un utilisateur et un pour représenter l'utilisateur courant. Naturellement, pour le premier, on a un ID de compte en tête. Si le prop est vide, alors on affiche le profil de l'utilisateur courant. Pour l'autre également.
 - Il ne contient aucun état.
 - Il contient les composants : User, MessageList, MessageForm.
-

Recherche :

- Ce composant permet de rechercher précisément un utilisateur ou des messages. Il y aura des filtres, ou des conditions (intervalle de temps, auteur...)
 - Il prend deux props : "signup_page" qui est une fonction permettant d'aller à la page d'inscription, et "login_page" pour aller à la page de connexion. Ces deux props seront toujours renseignés.
 - Pour chaque inscription de l'utilisateur sur le formulaire de recherche, il y a un état : ce qu'il recherche, intervalle de temps, auteur etc.
 - Il contient les composants : MessageList.
-

UserBanner:

- Ce composant représente la bannière gauche du site, ayant les boutons de déconnexion, ou pour changer de mode (forum, profil, validation, recherche).
 - Il prend trois props pour le moment : "isAdmin" pour tester si le bouton validation n'apparaît que si l'utilisateur est administrateur, "logout" pour se déconnecter, "setHomeCBpage" qui est une fonction pour changer de mode.
 - Il ne possède aucun état.
 - Il contient le composant : User.
-

Forum:

- Ce composant représente le forum public et privé, avec de quoi envoyer et lire les messages.
 - Il ne prend qu'un props pour le moment, permettant de savoir si l'utilisateur est administrateur ou non.
 - Il ne possède qu'un état : "privateForum" pour pouvoir changer entre le forum public et privé, si l'utilisateur est administrateur.
 - Il contient les composants : MessageList, MessageForm.
-

MessageForm:

- Ce composant permet de remplir un formulaire pour créer un message.
 - Il ne prend qu'un props pour le moment, permettant de savoir si l'utilisateur est administrateur ou non.
 - Il possède un état par input du formulaire : Titre, mode publique / privée pour les administrateurs, contenu.
 - Il ne contient aucun composant.
-

MessageList:

- Ce composant représente juste une liste de messages que l'on peut dérouler.
 - Il ne prend aucun props.
 - Il ne possède aucun état.
 - Il contient le composant Message.
-

Message:

- Ce composant représente un message du forum.
 - Pour le moment, il ne prend aucun props, mais il y en aura sûrement un pour représenter l'ID du message, afin d'y retrouver les informations essentielles (User → nom, prénom, admin, contenu du message ; chaque commentaire etc.).
 - Pour le moment, il ne possède aucun état.
 - Il contient le composant : CommentList
-

CommentList:

- Ce composant représente une liste de commentaires sous un message.
 - Il ne prend aucun props.
 - Il ne possède aucun état.
 - Il contient le composant Comment.
-

Comment:

- Ce composant représente un message du forum.
- Pour le moment, il ne prend aucun props, mais il y en aura sûrement un pour représenter l'ID du commentaire, afin d'y retrouver les informations essentielles (User → nom, prénom, admin, contenu du message). Ici, on suppose qu'on ne peut pas envoyer un commentaire SOUS un commentaire. Tout sera traité de façon linéaire sous le message original.
- Pour le moment, il ne possède aucun état.
- Il ne contient aucun composant.

User:

- Ce composant permet de changer des informations à propos de l'utilisateur.
- Il ne prend aucun props.
- Il prend un état par information : nom, prénom, email, photo de profil etc.
- Il ne contient aucun composant.

=====

Certains de ces composants ne sont qu'au stade très théorique, avec aucune pratique dessus (User, les commentaires, Recherche etc.) en attente des cours sur les bases de données. Par conséquent, certains props, états et même l'architecture pourraient changer.