Réseau social avec Graphql

Ce projet permet de mettre en œuvre un réseau social et ses fonctionnalités de base.

Technologies:

- TypeScript
- Graphql
- React / tsx
- Apollo Server/Clien
- Graphql codegen
- Prisma

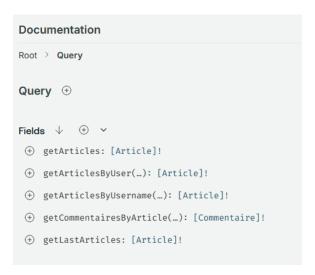
Sqlite

Le projet est séparé en deux parties: front et back.

Le back est une api Graphql et le front est un client en React. Les deux utilisent Apollo, TypeScript et Graphql Codegen.

Les éléments avec lesquels on eut intéragir sont: les utilisateur, les articles, les commentaires et les likes.

On peut retrouver la documentation plus précise de ce qui est possible sur l'interface Apollo:



Documentation Root > Mutation Mutation ⊕ Fields ↓ ⊕ ∨ ⊕ createUser(...): CreateUserResponse! ⊕ deleteArticle(...): DeleteArticleResponse! ⊕ deleteCommentaire(...): DeleteCommentaireResponse! ⊕ deleteLikeArticle(...): DeleteLikeResponse! ⊕ patchArticle(...): PostArticleResponse! ⊕ patchCommentaire(...): PostCommentaireResponse! ⊕ postArticle(...): PostArticleResponse! ⊕ postCommentaire(...): PostCommentaireResponse! ⊕ postLikeArticle(...): PostLikeResponse! ⊕ postLikeArticle(...): PostLikeResponse! ⊕ signIn(...): SignInResponse!

Documentation

Root > Query > getLast Articles

getLastArticles: [Article]! ①

Retrieves last 20 articles

Fields ↓ ⊕ ∨

- commentaires: [Commentaire]!
- ⊕ id: ID!
- ⊕ likes: [Like]!
- postDate: DateTime
- text: String!
- + user: User!

Documentation

Root > Query > getLast Articles > text

The article's content

Metadata for String Type

The String scalar type represents textual data, represented as UTF-8 character sequences. The String type is most often used by GraphQL to represent free-form human-readable text.

Sur le client on peut se créer un compte, se connecter, voir ses posts, rechercher des posts, poster des articles et des commentaires et les liker:

screeshots front

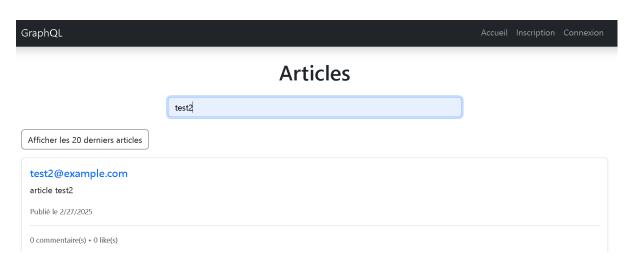
Notre interface se présente comme ci-dessous. Initialement, nous serons sur la page des articles, où l'on peut voir le contenu de l'article, l'auteur, ainsi que le nombre de likes et de commentaires.



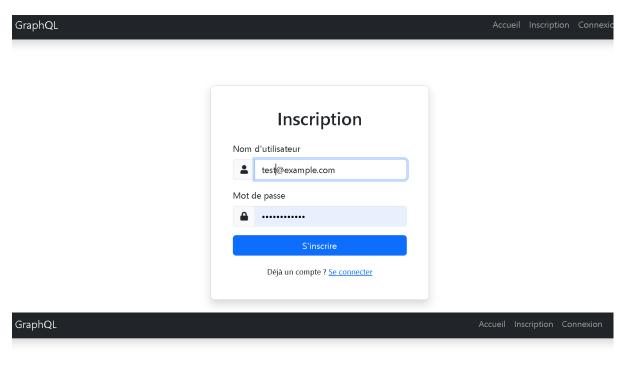
Sur la page initiale, on pourra afficher uniquement les 20 derniers articles.



On peut également rechercher les articles par nom d'auteur.

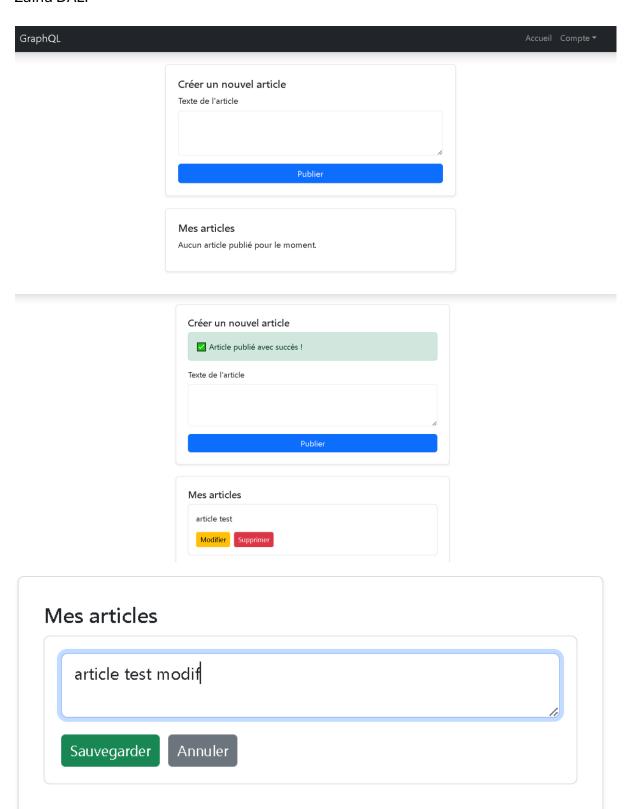


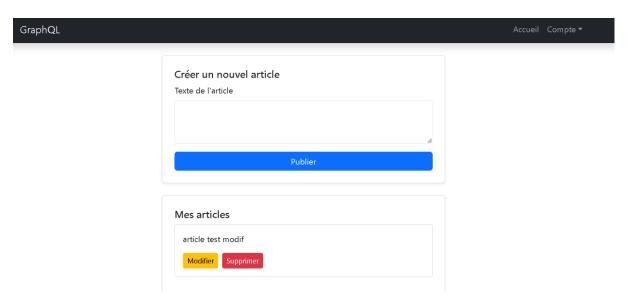
Un auteur pourra créer un compte et se connecter.



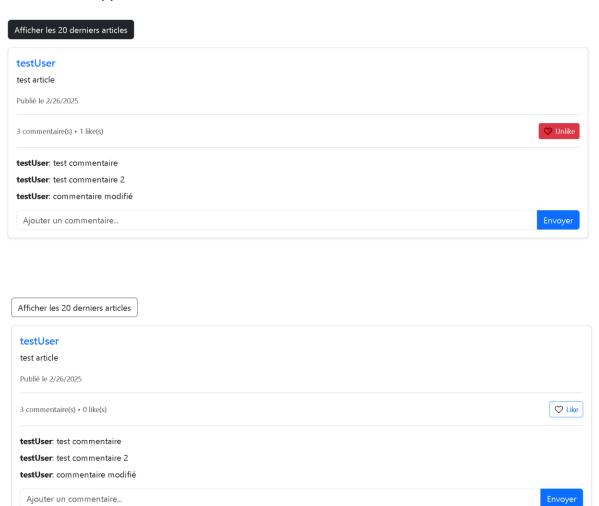


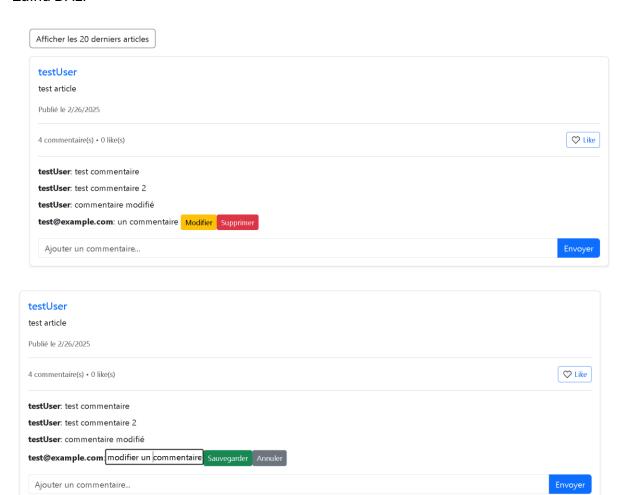
Une fois inscrit, l'auteur pourra publier des articles, les modifier ou les supprimer.

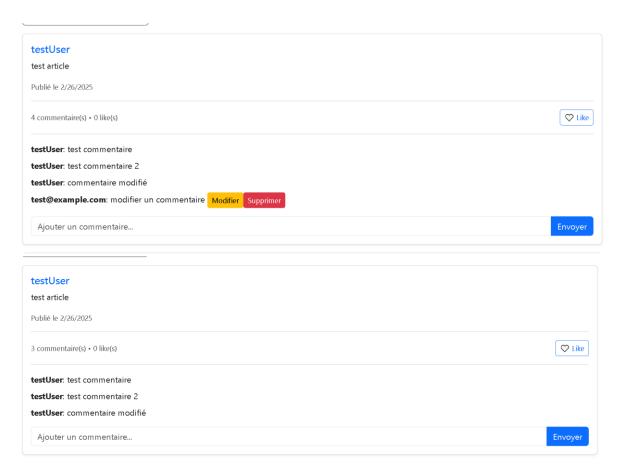




Un utilisateur connecté pourra également liker et commenter un article, ainsi que modifier ou supprimer son commentaire.







Un utilisateur connecté pourra se déconnecter

