

Sebastien LEFEVRE
Maxence BRUNET

Enseignant : Guillaume HIVERT

UE DAAR RENDU PROJET 2

Décembre 2020



SCIENCE ET TECHNOLOGIE DU LOGICIEL
SORBONNE UNIVERSITÉ
ANNÉE 2020/2021

Introduction

L'objectif du projet est d'implémenter un Wikipedia complet de manière décentralisée sur Ethereum.

1 Mode d'emploi

Avant toute chose, vérifier que :

-**Ganache** est bien lancé.

-**Metamask** est lancé et que vous avez sélectionné le réseau Ganache.

Le site web est composé de 4 pages :

Page d'accueil : Permet de naviguer vers les pages qui effectuent des actions de transactions entre le client et notre blockchain d'Ethereum.

Add article : Permet d'ajouter un article en échange d'éther. On entre le titre de l'article, son contenu et on appuie sur le bouton "publier". Une demande de transaction d'éther s'ouvre (Avec metamask). On accepte la transaction pour publier l'article.

All Articles : Permet de visualiser tous les articles publiés. On entre l'id de l'article, on appuie sur le bouton "Rechercher". On accède à une page avec le titre et le contenu de l'article: On peut mettre à jour l'article à partir d'ici.

Mettre à jour un article : Après avoir recherché un article, on accède à la page de ce dernier. On double click sur le contenu de l'article pour le modifier. On appuie sur le bouton "O" pour accepter les changements. Une demande de transaction d'éther s'ouvre (Avec metamask). On accepte la transaction pour mettre à jour l'article.

2 Implémentation

Pour permettre une modélisation plus propre de nos articles, nous avons légèrement modifier la structure en lui ajoutant un titre. La structure est de la forme :

```
struct Article {  
    string title;  
    string content;  
}
```

En ce qui concerne la liste des articles, nous avons une map qui a pour clé un id unique et un article. Nous allons expliquer en détails chaque fonctions que nous avons mis en place pour la réalisation du site web décentralisé.

2.1 Le contrat wikipedia

En plus des méthodes déjà présente dans *wikipedia.sol*, on en rajoute 4 autres:

- *getID* : Qui permet de généré automatiquement un id unique à la creation d'un article.
- *articleTitle* : Renvoie le titre d'un article.
- *addArticle* : Ajoute un article.
- *updateArticle* : Met à jour un article.

2.2 App.js

Nous avons repris le concept (déjà en place) d'utiliser des composants fonctionnelle pour représenter nos différentes pages.

On modifie le composant App:

On souhaite dédier une page entière lorsque l'on consulte un article. Pour cela, nous devons récupérer l'id de l'article sélectionné dans le composant *AllArticles*. On passe l'id de l'article en parametre de notre composant d'affichage *ArticleRechercher*. Enfin, on affiche ses informations sur notre nouvelle page.

On modifie le composant AllArticles:

On récupère à partir du contrat Wikipedia, les ids de tous les articles. Pour chaque articles on récupère le titre et le contenu. Dans le cas de base, on affiche l'ensemble des articles (id, titre) sur la page ainsi qu'une barre de recherche. Si une recherche est effectuée, on récupère l'id et on effectue une redirection vers le composant *ArticleRechercher*.

Le composant NewArticle:

On crée un formulaire qui récupère le titre et le contenu de l'article. Puis on appelle notre méthode du contrat *addArticle* pour effectuer une transaction (Ajouter un article coûte de l'éther mais consulter un article est "gratuit").

Le composant ArticleRechercher:

On récupère l'id passer en argument par le composant App. Et on affiche l'article. Lorsque l'on double click sur le contenu, on peut le modifier. Pour effectuer la modification, on appelle notre méthode *updateArticle* de notre contrat où on effectue une transaction similaire à celle effectuée lors de l'ajout d'un article.

3 Ouverture

Nous avons malheureusement pas eu le temps de réaliser un frontend de meilleure qualité. En ce qui concerne la sécurité du site web, certaines routes ne sont pas protégées (On peut accéder à une route en tapant son url) mais pourrait l'être avec un protected router. Cependant, lorsqu'un accès via l'url est effectué, nous avons mis un message pour avertir l'utilisateur qu'il ne peut pas effectuer d'action. Plusieurs articles peuvent avoir le même titre. Mais cela ne pose aucun problème puisque qu'ils possèdent des id unique différents.

4 Conclusion

Le site web décentralisé permet de publier des articles avec un id unique. Un utilisateur peut publier un article avec un titre et un contenu (l'id est généré automatiquement via la méthode *getID*) ainsi que modifier un article déjà existant.