



Que peut-on faire avec Substrate côté client web ?

Généralités sur les noeuds

Un noeud à deux “couches” : une interne appelé runtime qui compte toute la logique d'interaction avec la blockchain, et une couche externe qui s'occupe de toute la partie réseau (découverte de pairs, gestion des demandes de transactions, consensus avec les autres noeuds, RPC calls).

Il existe différents types de noeuds avec chacun des caractéristiques différents, en fonction du rôle que l'on veut pour le noeud :

- Full nodes, les plus répandus, peuvent stocker de la data dans la blockchain, participer aux différentes opérations tel que l'autorisation et validation des blocs, réception et vérification des transactions, répondre aux requêtes des autres utilisateurs. Ils gardent en mémoire les 256 derniers blocs (valeur par défaut modifiable).
- Archive nodes pareil que précédemment, sauf qu'ils contiennent tout l'historique des blocs. Utiles pour des applications qui nécessitent de pouvoir accéder à toutes les infos.
- Light client node : peuvent simplement lire les données de la blockchain et ne font quasi rien d'autre.

Les noeuds “Light Client”, au coeur de l'interface web

Les noeuds dits Light Client sont ceux qui servent à faire des applications/sites web d'exploration de la blockchain.

Les noeuds light client sont implémentés en un webAssembly client, appelé Smoldot. Pour faciliter l'intégration dans une application js, substrate met à disposition un package javascript appelé “Substrate Connect”. Ainsi notre web app peut facilement communiquer avec un noeud Light Client, et avoir accès aux données de la blockchain. On a la possibilité d'utiliser un noeud tournant en local, pour éviter de passer par un noeud inconnu ou par d'autres serveurs.




Voici quelques points intéressants qui sont mentionnés dans la documentation :

- On peut se connecter à n'importe quelle chaîne basée sur Substrate, comme Polkadot par exemple
- Compatible avec des apps utilisant Polkadot-JS API



- On peut facilement en suivant la doc se connecter à une blockchain connue, à une para-chain, ou à une chaîne locale.
- Dispose d'une extension navigateur, permettant d'être connecté à la blockchain tant que le navigateur est ouvert.
- Lien d'un template hébergé en ligne, connecté a la blockchain polkadot, pour voir les fonctionnalités disponible :
<https://substrate-developer-hub.github.io/substrate-front-end-template/?rpc=wss://rpc.polkadot.io>
- lien du code source du template :
<https://github.com/substrate-developer-hub/substrate-front-end-template>

Exemple d'interface réalisable avec le framework

 Create an account with Polkadot-JS Extension (Chrome, Firefox)  Select an account 

Parity Polkadot
Polkadot
wss://rpc.polkadot.io

v0.9.36-dc25abc712e

Metadata
v14

Show Metadata

13,856,233
CURRENT BLOCK

4

13,856,230
FINALIZED BLOCK

6


Balances

No accounts to be shown

Transfer

1 Unit = 1000000000000

Transfer more than the existential amount for account with 0 balance

Select from available addresses 

To address

Amount

Submit

Upgrade Runtime

Wasm File

Choisir un fichier

Aucun fichier choisi

Upgrade

Pallet Interactor

Interaction Type ☒ Extrinsic ☐ Query ☐ RPC ☐ Constant

Pallets / RPC

Callables

Unsigned or Signed or SUDO

Events

 system:ExtrinsicSuccess
[{"weight":
{"refTime":"487,964,346,000","proofSize":"0"},"class":"Mandatory","payFee":"Ye

 paraInclusion:CandidateBacked
[{"descriptor":
{"paraId":"2,040","relayParent":"0x796c227dc777b307b370edc28e71720374d

 paraInclusion:CandidateBacked
[{"descriptor":
{"paraId":"2,035","relayParent":"0x796c227dc777b307b370edc28e71720374d