



Duniter API HTTP



Arbre de Merkle des peers

Afin de conserver des données à jour et de rendre cette validité vérifiable, on utilise des arbres de Merkle, c'est à dire des arbres où le hash d'un noeud est le hash de la somme des hash de ses enfants, et où les feuilles sont les éléments stockés.

Les arbres de merkle sont construits sur les peers dans Dunitier.

Cela est accessible via `network/peering/peers` (GET) avec deux paramètres:

`leaves`: affiche le détail des éléments contenus dans l'arbre

`leaf`: spécifie un hash de leaf dont le détail sera fourni par la réponse

Pour les peers, les hash initiaux des feuilles sont les hash des clés publiques



Endpoints /node information des noeuds

/node/summary GET -> version logicielle

/node/sandboxes GET -> espace restant sur les transactions, identités, et membres (espace ?)



Endpoint /wot

Toile de confiance

/wot/add POST -> ajouter une identité à la toile de confiance, renvoie les infos actuelles

/wot/certify/ POST -> certifie une identité sur la toile

/wot/revoke POST -> retire une identité, retourne la réussite ou l'échec suivant si l'identité est dans la blockchain ou non

wot/lookup/[terme recherché] GET -> renvoie les clés publiques matchant, ou dont l'uuid matche le terme recherché

/wot/members GET -> renvoie la liste des membres de la toile

/wot/requirements/[pubkey] GET -> renvoie le nombre de certifications restantes à recueillir pour une identité souhaitant devenir membre

/wot/certifiers-of/[recherche] GET -> renvoie la liste des identités certifiées par une identité

/wot/certified-by/[recherche] GET -> renvoie la liste des certifieurs d'une identité donnée

wot/identity-of/[search] GET -> renvoie le nom correspondant à une identité



/blockchain/*

blockchain/parameters GET -> les paramètres blockchain du noeud

blockchain/membership POST-> permet de fournir une fiche de membre au noeud

blockchain/memberships/[search] GET -> chercher un membre en particulier

blockchain/block POST -> ajout d'un bloc à la blockchain

blockchain/block/[NUMBER] GET-> recherche du détail du bloc numéro [number]

blockchain/blocks/[COUNT]/[FROM] GET -> les count blocs suivant le bloc from

blockchain/current GET -> le dernier bloc ajouté

blockchain/hardship/[PUBKEY] GET -> la difficulté pour la clé publique donnée

blockchain/difficulties GET -> toutes les difficultés

blockchain/with/[some] GET -> trouve tous les blocs qui disposent du paramètre désignant un type de contenu.

blockchain/branches GET -> renvoie la liste des branches



network/

Propagation des informations

network/peers GET -> tous les peers connus du noeud

network/peering GET -> informations liées au peering (?)

network/peering/peers GET -> arbre de merkle sur les peers

network/peering/peers POST -> ajouter une entrée dans l'arbre



transactions et dividende

/tx/* /ud/*

tx/process POST -> ajouter une transaction

tx/sources/[pubkey] GET -> liste des sources (?) du payeur

tx/history/[pubkey] GET -> liste de l'historique des transactions d'une clé publique donnée

tx/history/[PUBKEY]/blocks/[from]/[to] GET

et tx/history/[pubkey]/times/[from]/[to] GET -> liste des transactions = historique

ud/history/[pubkey] GET -> historique du dividende



Création de connexions Websockets

`ws/block` -> crée une connexion websocket pour les blocks

`ws/peer` -> crée une connexion websocket pour les peers