

### **Rapport TD3: Blockchain development**

Notre code se trouve dans le fichier Script.py contenu sur la **branche Aymeric**, qui contient notre fichier final.

#### 1. Fonction displayAllAsset()

Cette fonction appelle l'API '/api/v3/exchangeInfo' de binance et grâce à pprint() affiche le json obtenu. Nous sélectionnons ensuite l'objet 'baseAsset' contenu dans l'objet 'symbols' du résultat.

#### 2. Fonction displayBidAsk()

Cette fonction appelle l'API '/api/v3/depth?symbol=%s&limit=10' avec les paramètres symbol et limit en query parameter.

#### 3. Fonction displayOrderBook()

Cette fonction affiche l'orderBook en appelant l'API avec divers paramètres également. Même type d'affichage que dans les parties précédentes avec pprint().

#### 4. Fonction refreshDataCandle()

Cette fonction affiche une bougie avec les paramètres désirés et return avec pprint les éléments.

#### 5. Fonction CreationTable()

Cette fonction crée une table si elle n'existe pas et cette table prendra le nom des paramètres pair et duration. Elle crée également une table lastchecks si elle n'existe pas.

#### 6. Fonction updateCandle()

Cette fonction ajoute une bougie dans la base de donnée si une nouvelle bougie est disponible. Pour savoir cela, cette fonction prends en paramètre le temps de « péremption de la dernière bougie », le nom de la table de la base de donnée et le couple pair/duration pour appeler l'API. On ajoute une bougie si : «  $\text{int}(\text{round}(\text{time.time()} * 1000)) \geq \text{lastTime}$  » avec le membre de droite le temps de péremption de la bougie précédente et le membre de gauche, le timestamp actuel.

#### 7. Fonction connection\_sql()

Cette fonction créer la connexion à sqlite3 et a la base de donnée '.db'. Il y a une boucle while infinie qui actualise les bougies dès qu'une nouvelle est disponible en appelant la fonction updateCandle() toutes les 30s pour vérifier si une nouvelle bougie est disponible ou non. Il faut interrompre le terminal avec CTRL C pour sortir de la boucle while. Dès qu'une bougie est affichée, « ajout » apparait sur le terminal.

#### 8. Fonction createOrder()

Cette fonction appelle l'api binance avec une méthode POST. Dans le header de la requête, nous aurons la clé publique de l'API. Et dans l'url, nous avons une signature obtenue grâce à la fonction hmac sha256 avec en key la clé privée de l'API et en data, l'url de l'API avec les paramètres (sans la signature bien sûr)

#### 9. Fonction CancelOrder()

Cette fonction utilise le même principe que la précédente en ayant une méthode DELETE et en ayant des paramètres différents.