Blockchain Programming TD3

Nicolas BAIN et Vincent MOUTEL

ShowCryptoCurrencies : To use the request function, we need to put an URL in the parameters. By using the function r_json.load, we stock in the variable r_json the different currencies in a string format. We use a list all_currencies to separate the different currencies stocked in r_json. Enfin on trie la liste par ordre alphabétique grâce à la fonction sort.

getDepth: This function is usefull to have the display for the ask or bid price of an asset. We choose BTC-USD and BTC-EUR as an exemple to test our program. We only need to put the pair in the URL.

LaunchgetDepth: We use this function to specify the direction and the name of the asset pair to launch the getDepth function with this two parameters.

GetOrderBook: With this function we have the order book at the third level (more information). We have the price, the size and the order ID of the product.

LaunchOrderBook: Same thing as the Launch function above but we only specify the pair of the asset.

refreshDataCandle: It is function used to read aggregated trading data (candles) where the granularity correspond to the timeslice of the candle in seconds. Those parameters (pair and granularity) are specified in the LaunchCandles function.

CreateSqlTable : We first connect python with sqlite3 server on ‘test.db’. Then we use the function execute to apply an SQL request, here it is the request ‘CREATE TABLE’. Everytime we open a connection, we need to close it with the function close. Now the table is created.

CreateCandlesDB: This function is used to specify the parameters of the SQL table we need to create to store the candles data in an sql database. This Table will correspond to a candle so we need to ask for the exchange plateform, the pair of the asset and the duration of the candle. We couldn’t use the setTableName into an SQL request because of an error we couldn’t fix after a lot of researches so we enter manually the name of the Table.

refreshData : This function is used to extract all available trades data. We specify the pair asset we want to extract in the LaunchrefreshData function.

FullDataSet : We create this function to set a new data table to store the trades data into it.

createOrder : We used the class **CoinBaseExchangeAuth** given in the API of coinbase to connect with the api_key, the secret_key and the passphrase. Then we created an order with {} to fit in the json format. Then we used the post request to create an order with the given parameters.

cancelOrder: To cancel an order, we also need to connect with the api_key, the secret_key and the passphrase and we need the uuid to have the id of the order we want to delete.

To protect the account we remove the passphrase , api_key and the secret_key on the git hub .