

TP - Récupération de données cryptographiques avec Binance et Apache Hop

Clément AGRET

1 Introduction

Dans ce TP, nous allons apprendre à récupérer des données cryptographiques à l'aide de l'API Python de Binance et à les intégrer avec Apache Hop. Nous utiliserons ces données pour simuler des scénarios d'achat et de vente de cryptomonnaies.

2 Prérequis

- Python 3.x installé
- Bibliothèque Python `requests` installée (`pip install requests`)
- Bibliothèque Python `binance` installée (`pip install python-binance`)
- Apache Hop installé

3 Récupération des données avec l'API Binance

3.1 Création d'un compte Binance

Rendez-vous sur <https://www.binance.com/> et créez un compte. Une fois connecté, allez dans l'onglet **API Management** et créez une nouvelle clé API. Notez votre **API Key** et votre **Secret Key**, nous en aurons besoin plus tard.

3.2 Utilisation de l'API Python Binance

Voici un exemple de code Python pour récupérer les derniers prix des cryptomonnaies Bitcoin (BTC) et Ethereum (ETH) en utilisant l'API Binance :

```
from binance.client import Client

api_key = 'VOTRE_API_KEY'
api_secret = 'VOTRE_SECRET_KEY'

client = Client(api_key, api_secret)
```

```

btc_price = client.get_symbol_ticker(symbol="BTCUSDT")
eth_price = client.get_symbol_ticker(symbol="ETHUSDT")

print(f"Prix du BTC: {btc_price['price']}")
print(f"Prix de l'ETH: {eth_price['price']}")

```

Remplacez `VOTRE_API_KEY` et `VOTRE_SECRET_KEY` par les clés que vous avez obtenues précédemment.

3.3 Écriture des données dans un fichier CSV

Modifions le code précédent pour écrire les prix dans un fichier CSV :

```

import csv

# ...

with open('prices.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(['Symbole', 'Prix'])
    writer.writerow(['BTC', btc_price['price']])
    writer.writerow(['ETH', eth_price['price']])

```

Cela créera un fichier `prices.csv` avec les prix actuels du BTC et de l'ETH.

4 Intégration avec Apache Hop

4.1 Importation du fichier CSV dans Hop

Ouvrez Apache Hop et créez un nouveau workflow. Ajoutez une transformation "CSV file input" et configurez-la pour lire le fichier `prices.csv` créé précédemment.

4.2 Transformation des données

Ajoutez une transformation "Select values" pour ne garder que les colonnes `Symbole` et `Prix`. Vous pouvez ensuite ajouter d'autres transformations selon vos besoins.

4.3 Écriture des données transformées dans une base de données

Ajoutez une transformation "Table output" pour écrire les données transformées dans une base de données de votre choix (par exemple, PostgreSQL, MySQL, etc.). Configurez la connexion à votre base de données et spécifiez la table cible.

4.4 Exécution du workflow

Sauvegardez votre workflow et exécutez-le. Vérifiez que les données sont bien écrites dans votre base de données.

5 Simulation d'achat et de vente

5.1 Création d'un nouveau workflow

Créez un nouveau workflow dans Hop pour simuler des scénarios d'achat et de vente de cryptomonnaies.

5.2 Lecture des données depuis la base de données

Ajoutez une transformation "Table input" pour lire les données de prix depuis votre base de données.

5.3 Définition des règles d'achat et de vente

Ajoutez une transformation "Filter rows" pour définir les règles d'achat et de vente. Par exemple :

- Si le prix du BTC est inférieur à 50000\$, achetez.
- Si le prix de l'ETH est supérieur à 3500\$, vendez.

5.4 Génération des ordres d'achat et de vente

Utilisez une transformation "Calculator" pour générer des ordres d'achat et de vente fictifs basés sur les règles définies précédemment. Ajoutez des champs pour la quantité, le prix total, etc.

5.5 Écriture des ordres dans un fichier CSV

Utilisez une transformation "CSV file output" pour écrire les ordres générés dans un fichier CSV.