

# REFACTORISATION D'UN SIMULATEUR DE TRADING DE CRYPTOMONNAIES



Thèse de Bachelor présentée par

**Benjamin ANTHONIOZ**

pour l'obtention du titre Bachelor of Science HES-SO en

**Informatique et systèmes de communication avec orientation en  
Logiciel**

**Mars 2025**

Professeur-e HES responsable

**Niklaus EGGENBERG**

Légende et source de l'illustration de couverture :

*Illustration 1 : Illustration trading de cryptomonnaies*

*Source : ref. URL01*

## TABLE DES MATIÈRES

<b>Remerciements.....</b>	<b>iv</b>
<b>Liste des acronymes .....</b>	<b>vi</b>
<b>Liste des illustrations.....</b>	<b>vii</b>
<b>Introduction .....</b>	<b>1</b>
<b>1. Chapitre 1 : Présentation du projet .....</b>	<b>3</b>
1.1. Cahier des charges.....	3
1.2. Le trading de cryptomonnaies.....	3
a) Définition .....	3
b) Termes techniques du trading.....	4
c) Les différentes stratégies .....	6
1.3. Description du projet.....	7
<b>2. Chapitre 2 : Présentation des technologies.....</b>	<b>9</b>
2.1. Outils utilisés .....	9
2.2. Choix technologiques .....	9
a) Frontend.....	9
b) Backend.....	9
c) Schéma de technologies .....	11
<b>3. Chapitre 3 : Etapes du projet.....</b>	<b>13</b>
3.1. Récupération des données.....	13
a) Listes non exhaustives des APIs .....	13
b) HitBTC.....	13
c) Kraken .....	14
d) CryptoCompare .....	14
e) Binance.....	14
f) Poloniex.....	14
g) Crypto.com.....	14
3.2. Interpolation du prix .....	15
3.3. Indicateurs techniques.....	18
3.4. Ordres d'achats/ventes.....	18
<b>4. Chapitre 4 : Implémentation du projet .....</b>	<b>19</b>
4.1. Routes de l'api.....	19
4.2. Classes du backend.....	19
a) Package config.....	20
b) Package controller.....	20
c) Package model .....	20
d) Package service .....	20
4.3. Fonctionnement du frontend .....	20
<b>Conclusion .....</b>	<b>24</b>
<b>Références documentaires .....</b>	<b>25</b>

## **REMERCIEMENTS**

Je remercie Mr. Eggenberg pour m’avoir aidé durant ce travail. Je remercie également mes amis pour leurs conseils dans la réalisation de ce travail.

## Résumé

Dans ce mémoire, nous aborderons la refactorisation d'un simulateur de trading de cryptomonnaies dans un langage orienté objet et avec un typage fort. Nous aborderons également les différentes stratégies de trading de cryptomonnaies ainsi que des explications sur différents indicateurs techniques nécessaires à la réalisation de ce projet. Nous verrons tout au long de ce mémoire comment ce projet a été réalisé. Nous passerons à travers ce projet, en expliquant les termes relatifs au trading de cryptomonnaies comme les indicateurs techniques, la volatilité du marché des cryptomonnaies ainsi que ce qu'est une bougie. Par la suite nous verrons une comparaison de différents langages de programmation qui nous permettent de choisir le langage pour le backend et les choix faits pour le frontend. Ensuite nous verrons une description des étapes du projet. Ces étapes expliquent comment nous avons choisi la source de nos données, provenant maintenant d'une API ainsi que des explications plus techniques concernant le traitement fait à ces données et comment nous déterminerons à quel moment il faut acheter ou vendre. Enfin, nous verrons également comment ce projet a été implémenté aussi bien du côté backend que du côté frontend et de leur communication en passant par une API. Ce projet reprend les travaux de Bachelor ayant été réalisés par le passé par messieurs Pighini, Toniutti et Souza Luz en 2022, 2023 et 2024 respectivement. Ce projet de semestre permet d'avoir un premier prototype et permettra de continuer la refactorisation du simulateur de trading de cryptomonnaies dans un langage orienté objet lors du travail de Bachelor.



Candidat-e :

**Anthonioz Benjamin**

Filière d'études : ISC

Professeur-e(s) responsable(s) :

**Eggenberg Niklaus**

Travail de semestre soumis à une convention de stage en entreprise : non

Travail soumis à un contrat de confidentialité : non

## LISTE DES ACRONYMES

- OHLC : Open High Low Close
- ETFs : Exchange Traded Funds
- RSI : Relative strength index
- MACD : Moving Average Convergence/Divergence
- CURL : client URL request library
- MFI : Money Flow Index
- DCA : Dollar Cost Average
- IDE : Integrated Development Environment
- HTML : Hypertext Markup Language
- CSS : Cascading Style Sheets
- JS : JavaScript
- API : Application Programming Interface
- NPM : Node Package Manager
- HTTP : Hypertext Transfer Protocol
- JVM : Java Virtual Machine
- JSON : JavaScript Object Notation

## LISTE DES ILLUSTRATIONS

Illustration 1 : Illustration trading de cryptomonnaies .....	ii
Illustration 2 : Schéma des niveaux de soutien et résistance.....	4
Illustration 3 : Schéma de bougies .....	5
Illustration 4 : Schéma des technologies .....	12
Illustration 5 : OHLC .....	15
Illustration 6 : O=HLC .....	15
Illustration 7 : OHLC .....	16
Illustration 8 : OHL=C .....	17
Illustration 9 : O=HL=C .....	17
Illustration 10 : OLHC .....	17
Illustration 11 : O=LHC .....	17
Illustration 12 : OLH=C .....	18
Illustration 13 : O=LH=C .....	18
Illustration 14 : Page d'accueil.....	21
Illustration 15 : Bougies récupérées depuis l'application Binance .....	22
Illustration 16 : Courbe du prix interpolé grâce aux bougies .....	22
Illustration 17 : Bougies récupérées depuis l'application Binance .....	22
Illustration 18 : Courbe du prix interpolé grâce aux bougies .....	23

## Références des URL

URL01	<a href="https://coveringcompanies.journalism.cuny.edu/category/stocks-2/">https://coveringcompanies.journalism.cuny.edu/category/stocks-2/</a>
URL02	<a href="https://www.avatrade.fr/cryptocurrencies/how-to-trade-cryptocurrencies#Strat%C3%A9gies%20de%20Trading%20de%20Cryptomonnaies">https://www.avatrade.fr/cryptocurrencies/how-to-trade-cryptocurrencies#Strat%C3%A9gies%20de%20Trading%20de%20Cryptomonnaies</a>
URL03	<a href="https://www.ig.com/fr-ch/strategies-de-trading/comprendre-les-figures-en-chandeliers-190419">https://www.ig.com/fr-ch/strategies-de-trading/comprendre-les-figures-en-chandeliers-190419</a>





## INTRODUCTION

Le trading de cryptomonnaies se démocratise de plus en plus notamment grâce à la multiplication des plateformes de trading comme Binance, Coinbase ou encore Crypto.com. Ces plateformes permettent d'acheter et de vendre des cryptomonnaies assez facilement notamment grâce à leurs applications sur smartphone. Cette accessibilité et cette simplification de l'achat et revente de cryptomonnaies permet à un plus grand public, même débutant, d'utiliser et de posséder des cryptomonnaies.

Avec une augmentation de 33% sur un an, le nombre de personnes détenant des cryptomonnaies dans le monde atteint 562 millions en 2024 contre 420 millions en 2023<sup>1</sup>. Cette augmentation de l'utilisation des cryptomonnaies montre que l'intérêt pour les cryptomonnaies augmentent que ce soit pour un simple investissement en vue de faire un profit ou même, pour certains, d'avoir une alternative aux monnaies fiduciaires.

Beaucoup de personnes remarquent la grande volatilité des prix des cryptomonnaies et font du profit grâce à ces variations de prix. Cependant, bien que certains fassent effectivement des bénéfices, d'autres font de grosses pertes. Et c'est pourquoi les simulateurs de trading de cryptomonnaies sont utiles. Ces simulateurs permettent de s'entraîner avec les stratégies existantes et de mieux comprendre le marché des cryptomonnaies sans pour autant avoir à réellement investir dans une cryptomonnaie et risquer de perdre leur capital investi. Ces simulateurs reproduisent les fluctuations des différentes cryptomonnaies soit en temps réel soit sur les données passées. S'entraîner sur une période passée peut paraître à première vue pas vraiment utile, mais en réalité cela peut permettre de s'entraîner sur une situation passée qui pourrait se reproduire à l'avenir. Les simulateurs sont donc aussi utiles à des débutants, pour leur faire découvrir le trading de cryptomonnaies sans leur faire perdre d'argent, ils peuvent y découvrir ce qu'est un ordre d'achat ou de vente et comment en placer, qu'à des traders plus expérimentés en leur permettant de revenir sur des périodes passées, ils peuvent y tester de nouvelles stratégies en utilisant les différents indicateurs à disposition. En réalisant ces tests sur un simulateur, il pourra par la suite voir les résultats obtenus et les appliquer dans le vrai trading de cryptomonnaies. Les simulateurs de trading de cryptomonnaies sont donc très utiles pour découvrir le trading de cryptomonnaies, pour tester et améliorer les différents outils et stratégies de trading de cryptomonnaies.

---

<sup>1</sup> <https://journalducoin.com/actualites/bilan-adoption-cryptos-monde-2024-rapport-triple-a/>

Ce projet est la suite de projets déjà réalisés par Mr. Pighini, Mr. Toniutt et Mr. SouzaLuz. Il est réalisé dans le cadre des projets de semestre et de Bachelor à HEPIA et a pour objectif d'améliorer le simulateur de trading de cryptomonnaies existant, notamment grâce à des notions de développement logiciel comme la programmation orientée objet, l'utilisation d'API et la réalisation d'une interface web. Ces méthodes de développement permettront de modulariser le code pour plusieurs raisons comme une meilleure maintenabilité, un code plus intuitif et de meilleures performances. En plus de la réimplémentation du projet dans un langage orienté objet, les données seront récupérées d'une API et plus d'une base de données. Ce changement de source de données nous permet d'avoir les données les plus récentes et de ne plus être restreint à des données fixes sur une certaine durée comme avoir les données de l'année 2022 seulement.

Dans ce rapport, nous commencerons par faire une présentation du projet en expliquant les termes techniques relatifs au trading et en définissant le cahier des charges et le projet. Ensuite, nous verrons les choix technologiques fait pour ce projet. Puis nous détaillerons les étapes du projet et enfin comment nous l'avons implémenté.

# 1. CHAPITRE 1 : PRÉSENTATION DU PROJET

## 1.1. CAHIER DES CHARGES

L'objectif de ce projet est de refactoriser un simulateur de trading de cryptomonnaies dans un langage orienté objet. Le choix du langage de programmation sera expliqué plus tard dans la sous-section “ Backend ” de la partie 2.2 du chapitre 2. Ce projet sera réalisé en suivant plusieurs points clés de ce cahier des charges.

- Choisir une API permettant de récupérer les données de cryptomonnaies depuis au moins 5ans.
- Afficher un graphique des bougies d'une cryptomonnaie sur une certaine période
- Créer le simulateur permettant de simuler l'évolution du prix d'une cryptomonnaie toutes les x secondes
- Faire un système d'achats et de ventes d'ordre
- Implémenter les différents indicateurs
- Créer et gérer un système de cache pour les données (moins important pour le moment)

## 1.2. LE TRADING DE CRYPTOMONNAIES

### a) DÉFINITION

Pour comprendre le trading de cryptomonnaies, il faut comprendre le terme de trading et ce qu'est une cryptomonnaie.

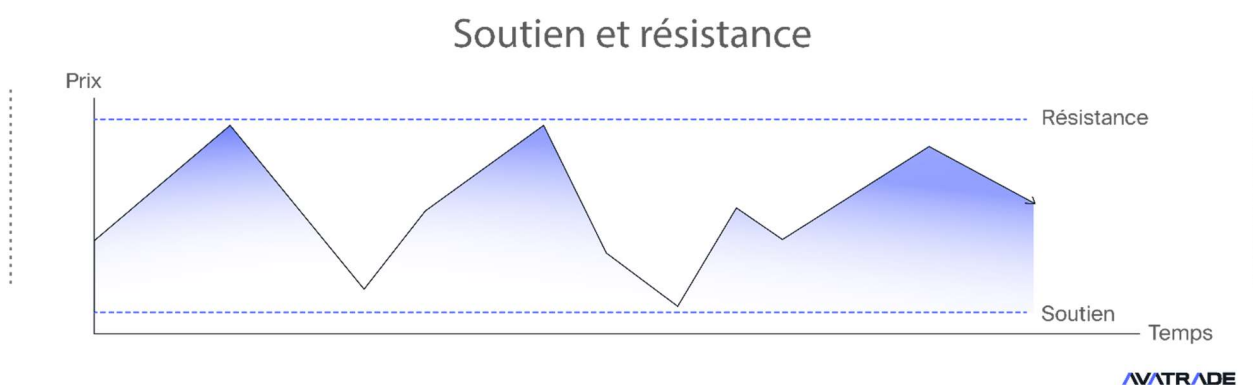
Le trading, c'est le fait d'acheter et de vendre des actifs comme des actions dans des entreprises (Google, NVIDIA, Apple...), des ETFs (groupes d'actions, pour être moins impacté si le cours d'une action chute), des matières premières (comme le pétrole, l'or, le blé...), afin de faire des bénéfices.

Une cryptomonnaie est un actif numérique qui est sécurisé et géré par une blockchain. Cette blockchain s'occupe d'enregistrer toutes les transactions effectuées depuis son lancement. Ce sont l'offre et la demande qui déterminent sa valeur donc en étudiant les achats et les ventes effectués on peut déterminer si la valeur va augmenter ou diminuer notamment grâce aux volumes achetés ou vendus.

## b) TERMES TECHNIQUES DU TRADING

Le trading a plusieurs termes techniques que nous allons définir ici. Un actif est un bien que l'on peut posséder pour pouvoir le revendre plus tard pour avoir des bénéfices si sa valeur a augmenté entre le moment de l'achat et le moment de la vente.

Les niveaux de résistance et de soutien entre lesquels le prix évolue sur une certaine durée. Le niveau de soutien est déterminé là où il y a assez de demandes pour que le prix ne descende pas plus bas et le niveau de résistance est déterminé là où il y a assez d'offres pour que le prix n'augmente pas plus.



*Illustration 2 : Schéma des niveaux de soutien et résistance*

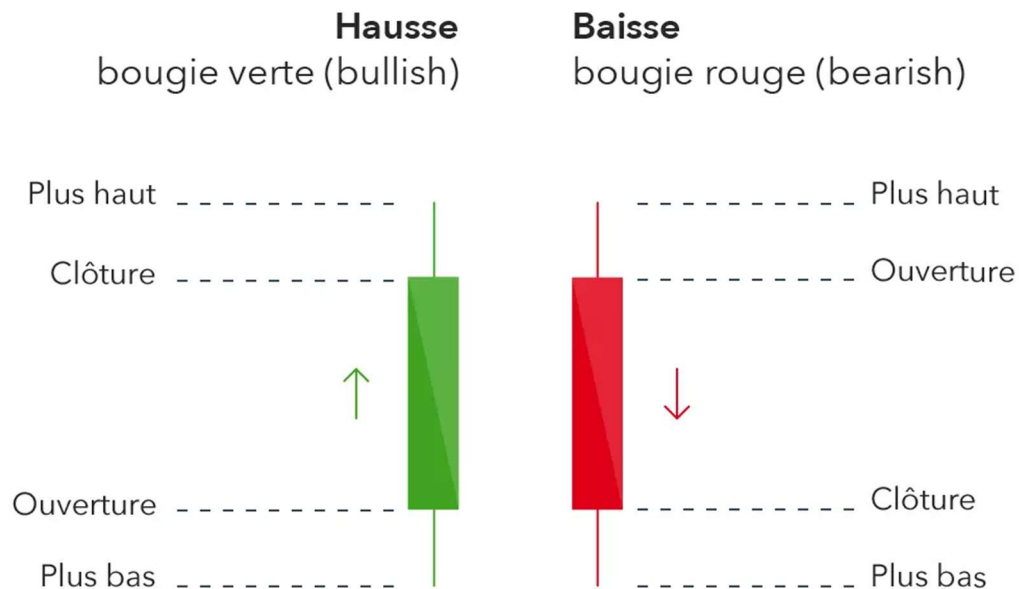
*Source : tiré de [avatrader.fr](https://www.avatrader.fr), ref. URL02 / réalisé par AvaTrade*

Le bear market désigne une période pendant laquelle le marché est en baisse. Ce bear market peut durer dans le temps en fonction des événements dans le monde comme par exemple une crise économique, une guerre. Dans ces cas-là, les détenteurs d'actifs se mettent à vendre leurs actifs pour se réfugier vers d'autres actifs plus stables en temps de crise. L'inverse du bear market est le bull market qui désigne un marché en hausse.

La volatilité du prix d'une cryptomonnaie représente le risque que le cours de la cryptomonnaie augmente ou diminue fortement sans avoir été prévu à un certain moment.

Les bougies représentent l'évolution du prix sur une certaine durée. Il y a deux catégories de bougies, les bougies vertes qui montrent une augmentation du prix à la fin de la durée de la bougie par rapport au début de la bougie et les bougies rouges qui montrent une baisse du prix. Les bougies atteignent quatre niveaux de prix différents lors de leur période. Les quatre niveaux de prix sont le prix d'ouverture, le prix de fermeture, le prix le plus haut et le prix le plus bas. Ces quatre niveaux peuvent être abrégés par OHLC pour Open/High/Low/Close en anglais. Les

bougies sont représentées par un corps et des mèches. Le corps d'une bougie est la zone située entre le prix d'ouverture et le prix de fermeture. Et les mèches sont les extrémités en dehors du corps d'une bougie qui représentent le prix le plus haut et le plus bas.



*Illustration 3 : Schéma de bougies*

*Source : tiré de tradingeducation.fr; ref. URL03 / réalisé par Chris Beauchamp*

Les indicateurs techniques permettent aux traders de savoir quand acheter ou quand vendre. Ces indicateurs techniques sont basés sur différents calculs mathématiques.

- RSI (Relative Strength Index)

Le RSI se représente par une valeur entre 0 et 100. Lorsque le RSI passe au-dessus de 70, cela veut dire que la cryptomonnaie connaît une phase de surachat. Cette phase de surachat pourrait impliquer une correction du prix. Grâce à cette indication les traders peuvent savoir qu'il faut vendre avant que le prix ne descende. La situation inverse se produit lorsque le RSI passe en dessous de 30.

- Moyennes mobiles

- Moyenne mobile simple

La moyenne mobile simple permet de mettre en évidence les tendances du marché sans être influencé par les variations à court terme. La moyenne mobile peut être calculé grâce à une période plus ou moins longue. Cependant, plus la période

de données est longue, moins les variations à court terme influencent la moyenne et donc met mieux en évidence les tendances de marché.

- Moyenne mobile exponentielle

La moyenne mobile exponentielle fonctionne comme la moyenne mobile simple à la simple mais donne plus d'importance aux données les plus récentes.

- Moyenne mobile convergence/divergence (MACD)

La MACD est la différence entre une moyenne mobile exponentielle courte et une moyenne mobile exponentielle longue. On calcul également la moyenne mobile exponentielle de la MACD. Lorsque la courbe de la MACD passe au-dessus de la ligne de signal, il faut acheter et lorsqu'elle passe en-dessous, il faut vendre.

- Indice du mouvement directionnel moyen (ADX)

L'ADX permet de déterminer la force d'une tendance de marché mais ne permet pas de déterminer sa direction. L'ADX s'exprime par une valeur entre 0 et 100. Cet indicateur permet donc de déterminer si une tendance va continuer ou s'arrêter.

### c) LES DIFFÉRENTES STRATÉGIES

Il existe plusieurs stratégies de trading de cryptomonnaie chacune avec ses avantages et ses inconvénients.

Le HODling est une stratégie de trading de cryptomonnaie qui consiste à investir dans une ou plusieurs cryptomonnaie sur du long terme. Les traders appliquant cette stratégie investissent dans des cryptomonnaies dont ils sont convaincus que le cours aura augmenté à long terme (souvent plusieurs années). Ils ne se préoccupent pas de des variations qu'il peut y avoir sur quelques jours ou mois. Pour savoir si le cours va augmenter ou non, ces traders peuvent s'appuyer sur les performances de la blockchain associée à la cryptomonnaie qui peuvent influencer sur le cours de la cryptomonnaie. Par exemple la blockchain Solana permet d'effectuer plusieurs dizaines de milliers de transactions par seconde contre 7 pour Bitcoin et 15 pour Ethereum<sup>2</sup>. Ce genre de différence de performances pourrait faire augmenter le cours de Solana.

Le DCA (Dollar Cost Average) consiste à investir une certaine somme d'argent à une certaine fréquence (une/deux fois par semaine/mois...), selon ce qu'on est capable d'investir et

---

<sup>2</sup> <https://kriptomat.io/fr/blockchain/reseaux-blockchain-les-plus-populaires/>

quel que soit le cours de la cryptomonnaie dans laquelle on souhaite. Cette stratégie permet de lisser le prix moyen d'achat s'il y a des moments quand on achète où le prix est plutôt haut ce prix haut sera compensé avec un autre moment où le prix sera plus bas.

La stratégie de suivre la tendance consiste à utiliser des indicateurs, que nous avons vu plus tôt, pour essayer de savoir si le cours a une tendance à la hausse ou à la baisse. Cette stratégie se base plutôt sur les tendances du marché sur le long terme. Elle ressemble à la stratégie du HODling mais est différente dans le fait que l'on ne s'appuie pas sur les performances de la blockchain mais plutôt sur des indicateurs techniques comme par exemple le MFI.

Le Swing trading est une stratégie qui se base sur l'utilisation d'indicateurs techniques pour trouver une hausse ou une baisse sur du court à moyen terme. Pour réaliser cette stratégie, nous pouvons utiliser les indicateurs techniques comme le MACD, les moyennes mobiles et le MACD.

La stratégie de Day trading est une stratégie de trading à court terme permettant de profiter de la volatilité du cours des cryptomonnaies en faisant des transactions qui durent quelques heures voire minutes seulement. Cette stratégie de trading nécessite d'être très attentifs aux changements de prix qui peuvent être très volatiles.

Le Scalping est une stratégie encore plus instantanée par rapport au Day trading car cette fois-ci, nous devons faire des transactions en quelques secondes ou minutes. Cette stratégie exploite les faibles mouvements de prix pour réaliser de très petits bénéfices et demande de faire un gros volume de transactions pour avoir beaucoup de petits bénéfices.

La stratégie de Breakout consiste à acheter si le niveau de résistance est dépassé et à vendre si le niveau de support est dépassé. Donc cette stratégie, nous incite à penser que lorsqu'un niveau est franchi, le cours va continuer dans la même direction un certain temps.

La stratégie de range est une stratégie que l'on peut utiliser lorsque le cours d'une cryptomonnaie varie entre le niveau de support et le niveau de résistance. Si le cours est proche du niveau de résistance, il faut vendre et si le cours est proche du niveau de support, il faut acheter. Avec cette stratégie, on considère que le cours ne devrait pas ou peu dépasser les niveaux et donc on peut deviner si le cours va monter ou descendre.

### **1.3. DESCRIPTION DU PROJET**

Ce projet consiste à réimplémenter un simulateur de trading de cryptomonnaies dans un langage orienté objet afin de pouvoir faire une interface commune aux différents indicateurs. Ces

indicateurs seront représentés par des classes qui devront implémenter cette interface commune. Contrairement au projet déjà existant en python, ce projet sera fait dans un langage fortement typé, c'est-à-dire que les objets auront un type assigné explicitement dès leur création et le garderont jusqu'à la fin de leur cycle de vie.

Le simulateur devra récupérer des données de type bougies depuis une API au lieu de les récupérer depuis une base de données. Puis il devra interpoler le prix, pour pouvoir par la suite réaliser des ordres d'achats et de ventes. Les ordres d'achat et de ventes seront automatisé grâce aux informations des différents indicateurs. L'affichage final affichera la courbe du prix ainsi que les ordres d'achats et de ventes pour savoir quand il aurait fallu acheter ou vendre. Les différents indicateurs seront également affichés.



## **2. CHAPITRE 2 : PRÉSENTATION DES TECHNOLOGIES**

### **2.1. OUTILS UTILISÉS**

Visual Studio Code a été utilisé pour implémenter le code de la partie frontend, l'IDE IntelliJ IDEA a également été utilisé pour la partie backend, en Java. Git a été utilisé comme gestionnaire de version.

### **2.2. CHOIX TECHNOLOGIQUES**

#### **a) FRONTEND**

J'ai commencé par faire un frontend simple avec HTML-CSS-JS pour pouvoir me concentrer plus sur les fonctionnalités du projet. C'est pourquoi dans le backend certaines routes retournent les pages HTML. Plus tard, je ferai un frontend plus avancé et réactif en utilisant un framework tel qu'Angular, React ou Vue.js.

Dans ce frontend, j'utilise également la librairie open-source lightweight-charts de TradingView qui me permet de créer des graphiques avec les données reçues depuis le backend. Cette librairie est utilisée notamment pour faire l'affichage graphique de l'évolution du prix de la cryptomonnaie via des bougies.

#### **b) BACKEND**

Pour le backend, le choix du langage de programmation est libre tant que le langage est orienté objet. Nous allons voir les avantages et désavantages de plusieurs langages afin de choisir un langage qui correspondra à notre utilisation.

- C#

Avantages :

- Accès au framework .NET
- Orienté objet permet de faire des classes et des interfaces
- Langage open-source et appartient à Microsoft

Désavantages :

- C# est plutôt utilisé sur Windows et n'est donc pas très adapté pour une application multi-plateformes

- Python

Avantages :

- Utilisation du langage facile avec de nombreuses fonctions et librairies déjà implémentées qui ne le sont pas forcément dans d'autres langages
- Implémentation d'une API assez simple grâce à FastAPI ou Flask<sup>3</sup>

Désavantages :

- Langage interprété ce qui implique de moins bonnes performances
- Pas réellement un langage orienté objet

- Node.js/Typescript

Avantages :

- Gestionnaire de package NPM
- Permet de faire de faire des classes et interfaces
- Mise en place d'une API simplifiée grâce au package Express
- On peut utiliser un seul langage pour le frontend et le backend

Désavantages :

- Peu efficace pour les calculs lourds
- Le type d'une variable n'est pas à écrire obligatoirement

- Java/Spring

Avantages :

- Configuration du projet simplifiée grâce à Spring Boot
- Spring utilise Maven ou Gradle pour la gestion des dépendances et de la compatibilité entre elles
- Spring gère les objets et de leur cycle de vie

---

<sup>3</sup> <https://www.browserstack.com/guide/top-python-rest-api-frameworks>

- Plusieurs fonctionnalités simplifiées comme la connexion à une base de données, l'implémentation d'une API ou la création de requêtes HTTP
- Exécution multi-plateforme grâce à la JVM

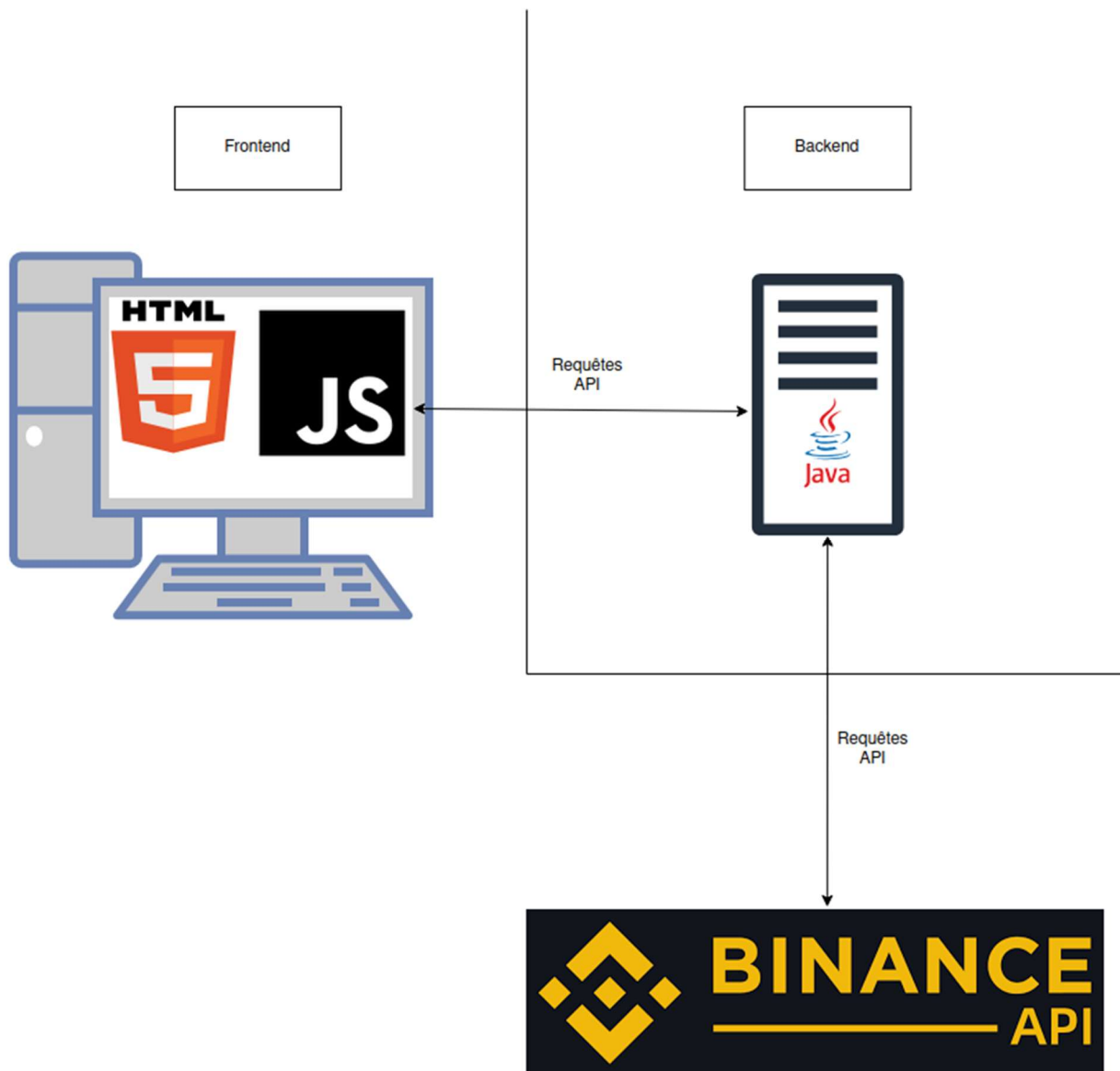
Désavantages :

- La prise en main peut être compliquée notamment si l'on ne connaît pas les différentes annotations et à quoi elles servent

Parmi tous ces langages, il n'y a que le C# et le Java qui nous permettent d'avoir un typage fort et strict ainsi que de l'orienté objet permettant de faire des interfaces et de l'héritage. Nous choisirons le Java car il est multiplateforme contrairement au C#.

### **c) SCHÉMA DE TECHNOLOGIES**

L'utilisation des technologies dans ce projet ressemble à l'illustration suivante.



*Illustration 4 : Schéma des technologies*

*Source : Schéma réalisés avec diagrams.net / réalisé par Anthonioz Benjamin*

### **3. CHAPITRE 3 : ÉTAPES DU PROJET**

#### **3.1. RÉCUPÉRATION DES DONNÉES**

Pour ce projet, les données vont provenir d'une API. L'utilisation d'une API pour récupérer les données au lieu de les récupérer depuis une base de données, permettra non seulement de ne pas devoir gérer une énorme base de données mais en plus, cela nous permettra d'avoir les données sur la période que l'on veut, même les données les plus récentes sans devoir, si on utilisait une base de données, ajouter les données dans la base de données.

L'API va nous permettre de récupérer les bougies de la cryptomonnaie que l'on veut pour qu'ensuite on puisse estimer le prix au temps que l'on veut. Cette estimation est expliquée plus tard dans la section 3.2.

J'ai pu essayer les différentes API en faisant des requêtes grâce à curl. Cela m'a permis de vérifier que les APIs fonctionnent comme il faut et aussi de vérifier si les documentations des APIs étaient correctes sur le nombre de données qu'elles peuvent retourner.

##### **a) LISTES NON EXHAUSTIVES DES APIs**

- HitBTC
- Kraken
- CryptoCompare (Coindesk)
- Binance
- Poloniex
- Crypto.com

##### **b) HITBTC**

HitBTC permet de récupérer au maximum 1000 bougies par requête qui ont un intervalle d'une minute, c'est à dire qu'entre deux bougies il y aura un temps d'une minute.

**c) KRAKEN**

Kraken permet de récupérer au maximum 720 bougies les plus récentes avec un intervalle d'une minute dans les bougies. Cette API couvre donc seulement 720 minutes soit 12 heures si l'intervalle est en minutes.

Cette API ne convient donc pas pour notre cas car il faut pouvoir avoir les bougies depuis plusieurs années.

**d) CRYPTOCOMPARE**

CryptoCompare permet de récupérer au maximum 2000 bougies par requête avec un intervalle de 1 minute.

**e) BINANCE**

Binance retourne au maximum 1000 bougies par requête avec un intervalle de 1 seconde.

**f) POLONIEX**

Poloniex retourne 500 bougies par requête avec un intervalle d'une minute.

**g) CRYPTO.COM**

Crypto.com nous retourne au maximum 300 bougies par requête avec un intervalle de 1 minute au minimum.

Parmi toutes ces APIs, on est sûr de ne pas choisir l'API Kraken car il n'y a pas assez de données retournées. Parmi les APIs restantes, elles correspondent à nos besoins pour ce projet. J'ai finalement choisi l'API de Binance car c'est une des plateformes de trading de cryptomonnaies la plus connue et avec le plus gros volume de trading sur 24h<sup>4</sup>.

Les données sont récupérées par intervalle de 15 minutes. C'est-à-dire que lorsque l'on récupère les données, on récupère la bougie du temps T de départ puis la bougie du temps T+15min, puis T+30min et ainsi de suite jusqu'à avoir toutes les bougies de la période souhaitée.

---

<sup>4</sup> <https://www.forbes.fr/finance/classement-les-plateformes-de-cryptomonnaies-les-plus-fiabiles-au-monde/>,  
<https://coinmarketcap.com/fr/rankings/exchanges/>

Une fois les bougies récupérées, nous devons faire une interpolation pour connaître le prix à un intervalle de temps plus précis comme par exemple à chaque seconde.

### 3.2. INTERPOLATION DU PRIX

Comme on récupère les bougies à un intervalle de 15 minutes, on doit faire une interpolation du prix à partir du prix d'entrée, de sortie, le plus haut et le plus bas sur un espace de 15 minutes. Cette interpolation permettra de connaître le prix à chaque instant sur l'intervalle de 15 minutes. Toutefois, pour la simulation, nous utiliserons les prix interpolés à chaque seconde.

Pour connaître le prix à chaque seconde, nous utiliserons l'interpolation de Lagrange. Pour faire cette interpolation, nous connaissons déjà quatre prix ce qui nous donne les coordonnées  $y$  des points et nous connaissons aussi le temps du prix d'ouverture et de fermeture ce qui nous donne seulement les coordonnées  $x$  de deux points. Pour le temps du prix le plus haut et le plus bas, nous prendrons deux points aléatoires entre le temps du prix d'ouverture et de celui de fermeture. Grâce à ces quatre points, nous pouvons utiliser l'interpolation de Lagrange qui nous permet d'obtenir une courbe passant par ces quatre points. Ainsi, grâce à cette courbe nous connaissons le prix à tout moment.

Selon les données récupérées de l'API de Binance, il y a plusieurs cas de figure qui se présente pour l'interpolation.

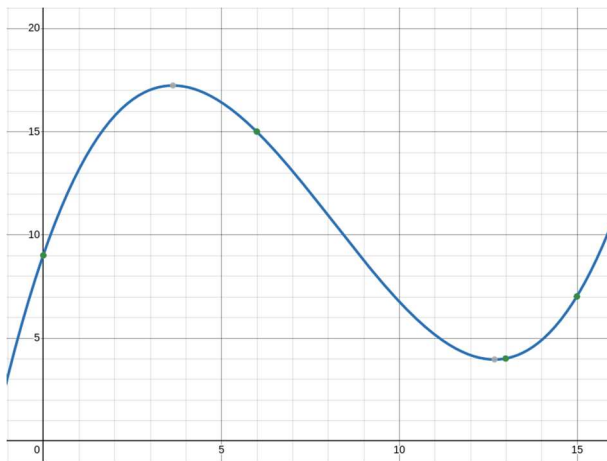


Illustration 5 : OHLC

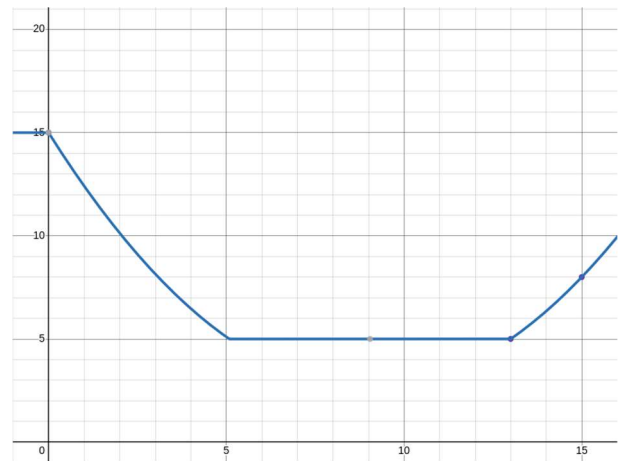
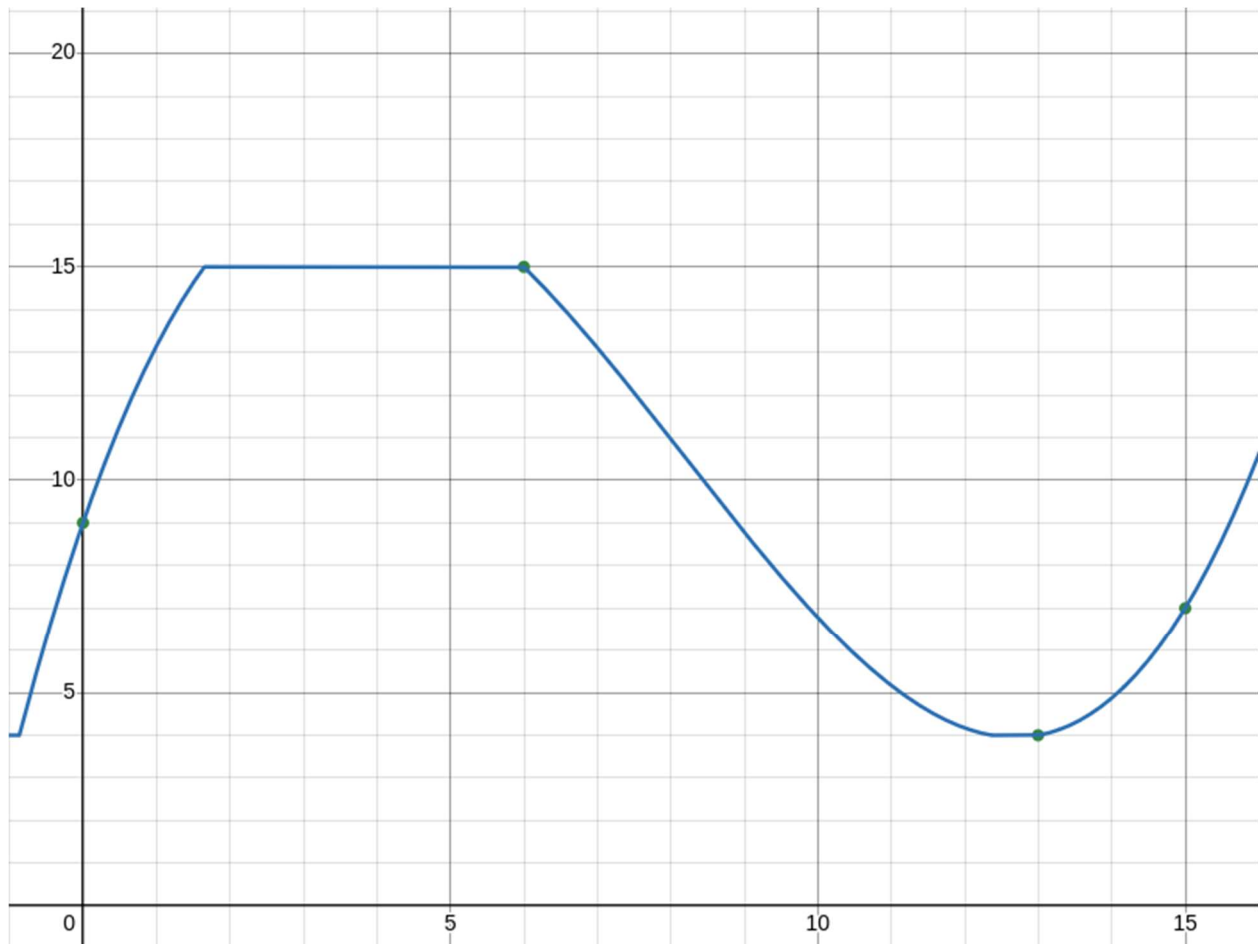


Illustration 6 : O=HLC

Source : Schémas réalisés avec Desmos / réalisés par Anthonioz Benjamin.

L'illustration 5 montre l'interpolation du prix lorsque les données sont dans l'ordre : Open/High/Low/Close. Pour cette illustration, nous avons choisi les points suivant dans l'ordre OHLC : (0, 9) pour l'ouverture, (6, 15) pour le prix le plus haut, (13, 4) pour le prix le plus bas et

(15, 7) pour la fermeture. Cependant, il y a un problème nous pouvons voir que la courbe dépasse le prix le plus haut pour aller jusqu'à 17 et dépasse aussi pour le prix le plus bas. C'est pourquoi dans notre simulateur le prix sera réajusté au prix le plus haut lorsqu'il le dépasse et au prix le plus bas lorsqu'il le dépasse aussi. Ainsi notre courbe de l'illustration n°5 deviendra la courbe de l'illustration n°7. Tous les autres schémas représenteront directement les courbes réajustées.

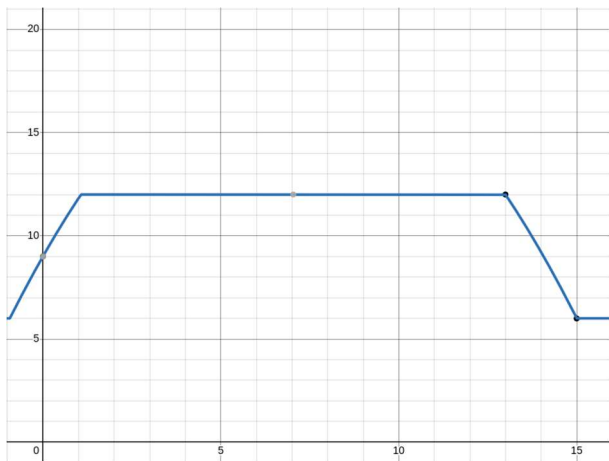


*Illustration 7 : OHLC*

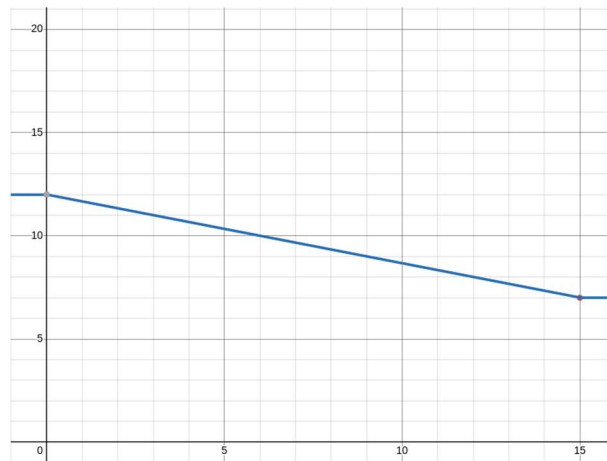
*Source : Schémas réalisés avec Desmos / réalisé par Anthonioz Benjamin.*

L'illustration 6 montre l'interpolation : Open=High/Low/Close. C'est à dire que le prix d'ouverture est aussi le prix le plus haut puis ensuite la courbe descend pour atteindre le prix le plus bas, puis remonte et atteint le prix de fermeture.





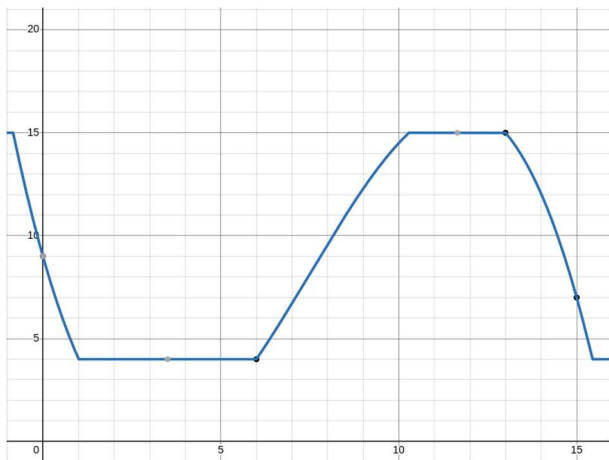
*Illustration 8 :  $OHL=C$*



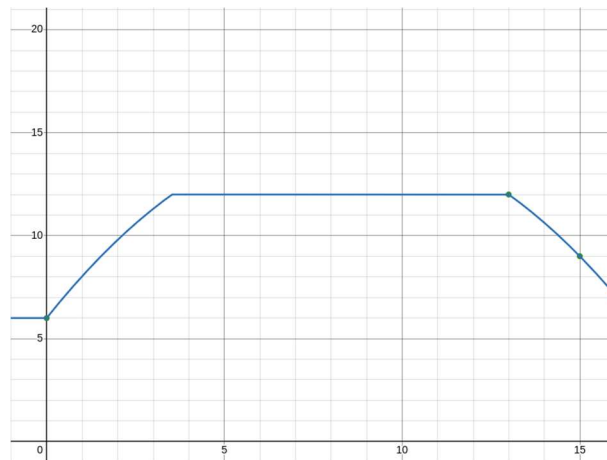
*Illustration 9 :  $O=HL=C$*

*Source : Schémas réalisés avec Desmos / réalisés par Anthonioz Benjamin.*

L'illustration 8 montre le cas où les données sont dans l'ordre Open/High/Low=Close. Dans ce cas le prix le plus bas est aussi le prix de fermeture. L'interpolation de l'illustration 9 représente le cas où le prix d'ouverture est aussi le prix le plus haut et le prix le plus bas est aussi le prix de fermeture.



*Illustration 10 :  $OLHC$*



*Illustration 11 :  $O=LHC$*

*Source : Schémas réalisés avec Desmos / réalisés par Anthonioz Benjamin.*

L'illustration 10 est le cas où l'ordre est Open/Low/High/Close. L'illustration 11 représente le cas où le prix d'ouverture est aussi le prix le plus bas, puis le prix monte pour atteindre le prix le plus haut et redescend pour atteindre le prix de fermeture.

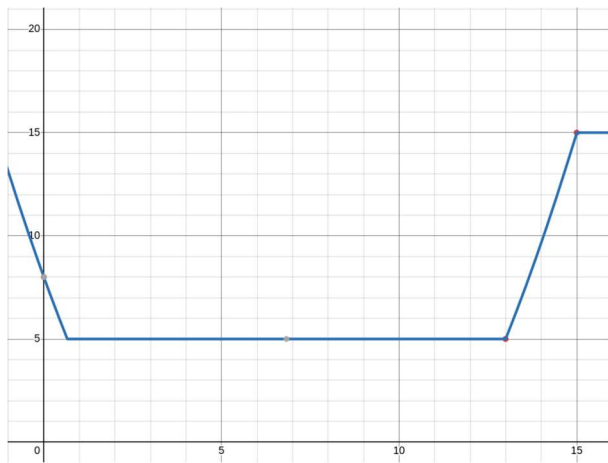


Illustration 12 :  $OLH=C$

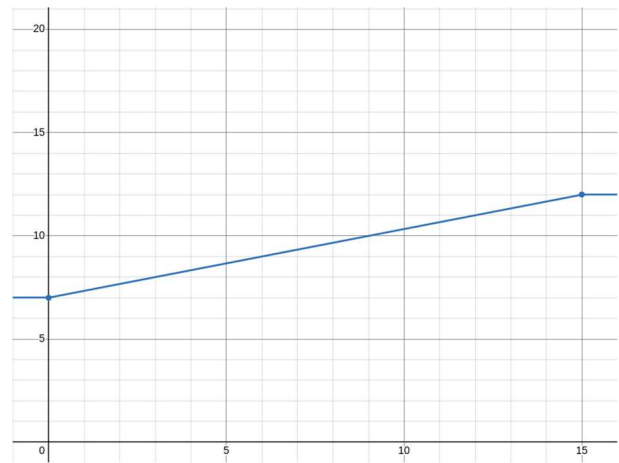


Illustration 13 :  $O=LH=C$

Source : Schémas réalisés avec Desmos / réalisés par Anthonioz Benjamin.

L'illustration 12 montre le cas où l'interpolation du prix est dans l'ordre Open/Low/High=Close. Dans ce cas, le prix le plus haut est aussi le prix de fermeture. L'illustration 13 représente le cas où le prix d'ouverture est aussi le prix le plus bas et le prix le plus haut est aussi le prix de fermeture.

Les cas des illustrations 9 et 13 sont des cas plus rares que les autres car la probabilité pour qu'il y ait 2 fois 2 prix qui soient égaux est plus faible. Ensuite les cas un peu plus fréquents sont les cas des illustrations 6, 8, 11 et 12. Enfin les cas les plus fréquents que l'on va retrouver le plus souvent sont les cas des illustrations 7 et 10.

### 3.3. INDICATEURS TECHNIQUES

Pour représenter un indicateur technique, nous utiliserons une interface qui contiendra ce que les différents indicateurs techniques ont en commun. Les différents indicateurs techniques seront représentés à travers des classes qui implémenteront chacune l'interface commune que l'on aura créée. Puis chaque classe aura les attributs et méthodes spécifiques à chaque indicateur.

### 3.4. ORDRES D'ACHATS/VENTES

L'utilisateur aura la possibilité de sélectionner les indicateurs techniques qui influenceront le fait de créer des ordres d'achat ou de vente. Les ordres d'achats et de vente permettront de garder en mémoire le moment de l'achat ou de vente afin qu'à la fin de la simulation nous puissions retrouver chaque valeur d'achat et de vente. Puis nous ferons la somme des bénéfices et pertes de chaque ordre d'achat et de vente pour connaître le bénéfice total.

## 4. CHAPITRE 4 : IMPLÉMENTATION DU PROJET

### 4.1. ROUTES DE L'API

L'API est pour l'instant séparé en 2 parties. Une partie retourne les pages web et sera remplacée par un réel frontend comme expliqué dans la sous-section “ Frontend ” de la partie 2.2 du chapitre 2. L'autre partie de l'API, s'occupe du traitement et de l'envoi des données.

La première partie de cette API se trouve à la base de localhost.

- /, GET
- /candlesGraph, GET

La seconde partie de l'API commence par /api

- /btcusdt, GET
- /sendDates, POST

La route “/” permet de nous renvoyer la page de base de l'application. Cette page contient un formulaire permettant de sélectionner la plage de date sur laquelle l'utilisateur veut exécuter la simulation de trading.

La route “/candlesGraph” nous renvoie la page où les graphiques de la simulation s'affichent.

La route “/api/btcusdt” permet de récupérer les données des bougies en format JSON.

La route “/api/sendDates” permet d'envoyer la date de début et la date de fin de la simulation au backend.

### 4.2. CLASSES DU BACKEND

Le backend est séparé en 4 packages différents. Il y a le package config qui contient la configuration du backend, le package controller qui contient les classes qui gèrent les requêtes HTTP faites à notre backend, le package model qui contient les classes qui représentent les données et le package service qui contient la logique de l'application, permet de faire des manipulations sur les données, aussi appelée logique métier.

**a) PACKAGE CONFIG**

La classe AppConfig est le point d'entrée de l'application et qui crée un objet RestTemplate qui permet de simplifier la création de requêtes HTTP qui est utile lorsque l'on fait des requêtes à l'API Binance.

**b) PACKAGE CONTROLLER**

Ce package contient deux classes controller qui nous permettent de réaliser notre API. La classe controller permet de créer les routes de la partie /api et la classe PageApiController permet de créer les routes qui s'occupent de retourner les pages web.

**c) PACKAGE MODEL**

La classe Candle permet de représenter une bougie grâce à ses attributs time et closeTime qui détermine son timestamp d'ouverture et de fermeture et ses prix d'ouverture, de fermeture, le prix le plus haut et le prix le plus bas. En plus de ces attributs, il y a la méthode checkPrice qui permet de réajuster le prix s'il dépasse le prix le plus haut ou le plus bas comme on l'a vu dans la partie 3.2. La classe Price permet de représenter un prix à un certain temps. La classe Interval permet de représenter le temps qui est passé entre le début d'une bougie et sa fin. La classe DateRange permet d'enregistrer les dates de début et de fin de la simulation envoyées par l'utilisateur. Elle permet également de convertir ces dates en timestamp en format secondes (nécessaire pour utiliser la librairie lightweight-chart) ou millisecondes (nécessaire pour les requêtes avec l'API Binance).

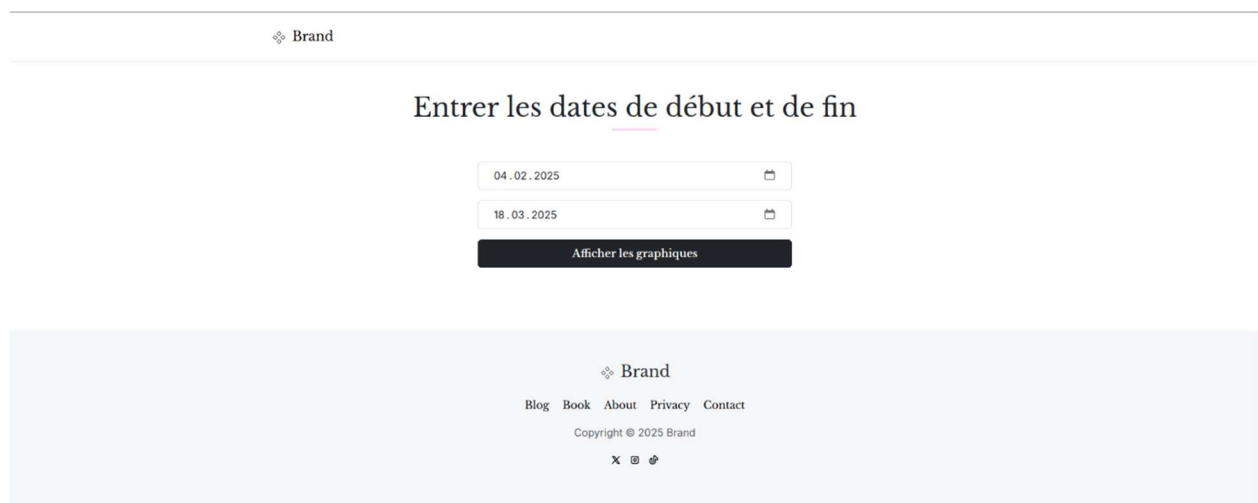
**d) PACKAGE SERVICE**

Ce package contient les deux classes contenant la logique applicative. La classe ApiService permet de faire la logique de l'API en récupérant les données depuis l'API Binance et la classe SimulateurService qui s'occupe de la logique du simulateur.

### **4.3. FONCTIONNEMENT DU FRONTEND**

Le frontend est pour l'instant de simples pages HTML accompagnées de script Javascript qui sont envoyées par le backend. Le frontend est constitué de 2 pages web accessibles par la route "/". La première page du site (voir l'illustration n°10) est une page qui permet de choisir la période que l'on veut grâce à un formulaire qui demande la date de début et la date de fin. Lorsque l'on

clique sur le bouton d'envoi, les données sont envoyées au backend et on est redirigé automatiquement sur la deuxième page du site (voir les illustrations n°11 et 12). Cette deuxième page affiche deux graphiques différents grâce à la librairie Lightweight Chart de TradingView. Le premier graphique affiche les bougies de la période sélectionnée sur la première page. Ces bougies sont récupérées directement depuis l'API de Binance avec un intervalle de 15 minutes. Le deuxième graphique affiche la courbe du prix interpolé grâce aux bougies. Le prix est calculé pour chaque seconde de la période sélectionnée. Lorsque l'on regarde les 2 graphiques en même temps, on peut voir que le tracé de la courbe du prix est à peu près le même que l'évolution des bougies. Les images suivantes montrent les résultats du projet.



*Illustration 14 : Page d'accueil*

*Source : Capture d'écran du prototype réalisé lors du projet de semestre / réalisé par Anthonioz Benjamin.*

## Bougies récupérées depuis Binance



### *Illustration 15 : Bougies récupérées depuis l'application Binance*

Source : Capture d'écran du prototype réalisé lors du projet de semestre / réalisé par Anthonioz Benjamin.

## Courbe de prix interpolé



### *Illustration 16 : Courbe du prix interpolé grâce aux bougies*

Source : Capture d'écran du prototype réalisé lors du projet de semestre / réalisé par Anthonioz Benjamin.

Les deux illustrations suivantes montrent les courbes plus zoomées sur la même période.

## Bougies récupérées depuis Binance



### *Illustration 17 : Bougies récupérées depuis l'application Binance*

Source : Capture d'écran du prototype réalisé lors du projet de semestre / réalisé par Anthonioz Benjamin.

## Courbe de prix interpolé



*Illustration 18 : Courbe du prix interpolé grâce aux bougies*

*Source : Capture d'écran du prototype réalisé lors du projet de semestre / réalisé par Anthonioz Benjamin.*

## CONCLUSION

Le but de ce projet était de réimplémenter un simulateur de trading existant en python dans un langage orienté objet avec un fort typage. En plus de cela, nous avons changé la source de données, qui auparavant provenait d'une base de données, par l'appel à une API. Ce changement de source de données, nous a permis de ne plus être obligé d'effectuer la simulation sur une certaine période et nous permet à présent d'effectuer des simulations en remontant sur plusieurs années.

Pour commencer ce projet, il a fallu déterminer quel langage de programmation utiliser en listant les langages orienté objet ainsi que les avantages et désavantages. Il a également fallu trouver une librairie nous permettant d'afficher plusieurs types de graphique. Une fois, que nous avons déterminé les technologies que nous allions utiliser, Nous avons pu faire l'implémentation du simulateur. Ce simulateur permet à l'utilisateur de sélectionner la période de simulation souhaitée. Une fois la période sélectionner, les graphiques des bougies et du prix sont calculés et affichés.

La réalisation de ce projet m'a permis d'acquérir de nouvelles compétences aussi bien sur le plan personnel que professionnel en découvrant plus en profondeur le monde des cryptomonnaies et du trading ainsi que de découvrir le framework Spring, qui pourra m'être utile pour le futur.

Ce simulateur devra être continuer dans le futur pour qu'il puisse proposer à l'utilisateur de sélectionner les indicateurs techniques qui seront utilisé dans la simulation. À l'aide de ces indicateurs, le simulateur devra ouvrir et fermer des ordres d'achats et de ventes afin de générer le plus de bénéfices possibles. Enfin le simulateur affichera le bénéfice total de la simulation pour que l'utilisateur puisse comparer avec le début de la simulation. Cette amélioration du simulateur sera effectuée lors du travail de Bachelor.



## RÉFÉRENCES DOCUMENTAIRES

- [1] W. Hoogenraad, « 6 langages de programmation modernes et leurs inconvénients - Bienvenue professionnel de l'informatique », Welcome IT professional. Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://fr.itpedia.nl/2023/07/13/6-moderne-programmeertalen-en-hun-downsides/>
- [2] « API Documentation ». Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://api.hitbtc.com/#candles>
- [3] M. Baillif, « Approximation polynomiale 2024 ». 7 juin 2024.
- [4] « Avantages et inconvénients de Node.js ». Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://fr.linkedin.com/pulse/avantages-et-inconv%C3%A9nients-de-nodejs-ali-abbas>
- [5] R. Chandran, « Avantages et inconvénients de TypeScript | Software Developer India ». Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://www.software-developer-india.com/fr/avantages-et-inconv%C3%A9nients-de-typescript/>
- [6] « Bear Market – Définition ». Consulté le: 20 mars 2025. [En ligne]. Disponible sur: <https://www.cashbee.fr/lexique/bear-market>
- [7] K. C, « Bilan de l'adoption des cryptos dans le monde en 2024 - Rapport Triple-A », Journal du Coin. Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://journalducoin.com/actualites/bilan-adoption-cryptos-monde-2024-rapport-triple-a/>
- [8] « Bull Market – Définition ». Consulté le: 20 mars 2025. [En ligne]. Disponible sur: <https://www.cashbee.fr/lexique/bull-market>
- [9] « Classement : les 25 plateformes de cryptomonnaies les plus fiables au monde - Forbes France ». Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://www.forbes.fr/finance/classement-les-plateformes-de-cryptomonnaies-les-plus-fiables-au-monde/>
- [10] « Classement des plateformes d'échange principales par volume », CoinMarketCap. Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://coinmarketcap.com/fr/rankings/exchanges/>

[11] « CoinDesk | API Documentation: Integrating Live & Historical Crypto Data », CoinDesk.io. Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://developers.coindesk.io/documentation>

[12] « Comment trader les cryptomonnaies ? | Guide AvaTrade », AvaTrade. Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://www.avatrade.fr/cryptocurrencies/how-to-trade-cryptocurrencies>

[13] « Convergence et divergence des moyennes mobiles - définition », IG. Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://www.ig.com/fr-ch/glossaire-trading/convergence-et-divergence-des-moyennes-mobiles>

[14] A. Srikanth, « Creating a REST API with Node.js and Express », Postman Blog. Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://blog.postman.com/how-to-create-a-rest-api-with-node-js-and-express/>

[15] freyastreamlit, *freyastreamlit/streamlit-lightweight-charts*. (13 mars 2025). Python. Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://github.com/freyastreamlit/streamlit-lightweight-charts>

[16] « Get OHLC Data | Kraken API Center ». Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://kraken.com/api/docs/rest-api/get-ohlc-data/>

[17] « Getting started | Lightweight Charts ». Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://tradingview.github.io/lightweight-charts/docs>

[18] « Glossaire de trading », IG. Consulté le: 21 mars 2025. [En ligne]. Disponible sur: <https://www.ig.com/fr-ch/glossaire-trading>

[19] « Home », Home. Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://spring.io/>

[20] « How to Call or Consume External API in Spring Boot? », GeeksforGeeks. Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://www.geeksforgeeks.org/how-to-call-or-consume-external-api-in-spring-boot/>

[21] « Interpolation lagrangienne », *Wikipédia*. 8 mars 2025. Consulté le: 27 mars 2025. [En ligne]. Disponible sur: [https://fr.wikipedia.org/w/index.php?title=Interpolation\\_lagrangienne&oldid=223685638](https://fr.wikipedia.org/w/index.php?title=Interpolation_lagrangienne&oldid=223685638)

[22] « Langage Python : c'est quoi ? Que peut-on faire avec en 2024 ? » Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://www.data-bird.co/blog/langage-python>, <https://www.data-bird.co/blog/langage-python>

[23] « Les 5 meilleures stratégies de trading crypto que tout trader devrait connaître », IG. Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://www.ig.com/fr-ch/strategies-de-trading/les-5-meilleures-strategies-de-trading-crypto-que-tout-trader-de-220926>

[24] « Les meilleures stratégies de trading sur les crypto-monnaies que tout investisseur devrait connaître - Kriptomat ». Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://kriptomat.io/fr/finances-et-investissements/les-meilleures-strategies-de-trading-sur-les-crypto-monnaies-que-tout-investisseur-devrait-connaître/>

[25] « lightweight-charts examples - CodeSandbox ». Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://codesandbox.io/examples/package/lightweight-charts>

[26] « Market Data endpoints | Binance Open Platform ». Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://developers.binance.com/docs/binance-spot-api-docs/rest-api/market-data-endpoints#klinecandlestick-data>

[27] « Market Data endpoints | Binance Open Platform ». Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://developers.binance.com/docs/binance-spot-api-docs/rest-api/market-data-endpoints>

[28] « MDN Web Docs ». Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://developer.mozilla.org/fr/>

[29] « Meilleure plateforme crypto de 2025 : le top des échanges », Journal du Geek. Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://www.journaldugeek.com/crypto/>

[30] « Overview (Java Platform SE 8 ) ». Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://docs.oracle.com/javase/8/docs/api/>

[31] « public/get-candlestick – Exchange API v1 ». Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://exchange-docs.crypto.com/exchange/v1/rest-ws/index.html#public-get-candlestick>

[32] « Python (langage) », *Wikipédia*. 14 mars 2025. Consulté le: 27 mars 2025. [En ligne]. Disponible sur: [https://fr.wikipedia.org/w/index.php?title=Python\\_\(langage\)&oldid=223878623](https://fr.wikipedia.org/w/index.php?title=Python_(langage)&oldid=223878623)

[33] « Qu'est-ce qu'une « cryptomonnaie » ? », AMF. Consulté le: 20 mars 2025. [En ligne]. Disponible sur: <https://www.amf-france.org/fr/quest-ce-quune-cryptomonnaie>

[34] A. Grisey, « Qu'est-ce que C# et pourquoi l'apprendre ? », FreelanceTalks. Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://talks.freelancerepublik.com/pourquoi-comment-apprendre-c-sharp/>

[35] « Quel avenir pour le monde des crypto-actifs ? », Banque de France. Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://www.banque-france.fr/fr/interventions-gouverneur/quel-avenir-pour-le-monde-des-crypto-actifs>

[36] « Spring @ResponseStatus Annotation », GeeksforGeeks. Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://www.geeksforgeeks.org/spring-responsestatus-annotation/>

[37] Concept && Coding - by Shrayansh, *Spring boot - Exception Handling | @ControllerAdvice | @ResponseStatus | @ExceptionHandler*, (26 octobre 2024). Consulté le: 27 mars 2025. [En ligne Vidéo]. Disponible sur: <https://www.youtube.com/watch?v=xfUspC01iDM>

[38] Java Techie, *Spring Boot | REST API Request Validation & Exception Handling Realtime Example | JavaTechie*, (3 avril 2022). Consulté le: 27 mars 2025. [En ligne Vidéo]. Disponible sur: [https://www.youtube.com/watch?v=gPnd-hzM\\_6A](https://www.youtube.com/watch?v=gPnd-hzM_6A)

[39] chillotech, *spring boot 101 - Créer votre première application avec Spring Boot*, (14 mai 2023). Consulté le: 18 mars 2025. [En ligne Vidéo]. Disponible sur: <https://www.youtube.com/watch?v=k6Nmt-l1Bzc>

[40] Marco, « Sur quels sites s'entraîner à trader gratuitement ? », Cryptoast. Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://cryptoast.fr/sur-quels-sites-sentraîner-a-trader-gratuitement/>

[41] « Top 10 Python REST API Frameworks in 2024 », BrowserStack. Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://browserstack.wpengine.com/guide/top-python-rest-api-frameworks/>

[42] « Tout ce qu'il faut savoir sur les ETF | VZ VermögensZentrum ». Consulté le: 20 mars 2025. [En ligne]. Disponible sur: <https://www.vermoegenszentrum.ch/fr/competences/tout-ce-qu-il-faut-savoir-sur-les-etf#a>

[43] crpto\_me, « Trading Crypto : Notre Guide du débutant pour 2025 », Journal du Coin. Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://journalducoin.com/trading/trading-crypto-notre-guide-debutant/>

[44] wadepickett, « Tutoriel : Créer une API web basée sur un contrôleur avec ASP.NET Core ». Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://learn.microsoft.com/fr-fr/aspnet/core/tutorials/first-web-api?view=aspnetcore-9.0>

[45] « TypeScript ou JavaScript ? Quel est le bon langage pour mon application? », Coursera. Consulté le: 27 mars 2025. [En ligne]. Disponible sur: [https://www.coursera.org/fr-FR/articles/typescript-vs-javascript?utm\\_source=chatgpt.com](https://www.coursera.org/fr-FR/articles/typescript-vs-javascript?utm_source=chatgpt.com)

[46] « Typescript, Est ce une si bonne idée? », Loopbin. Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://loopbin.dev>

[47] « TypeScript, Quand et Pourquoi l'Utiliser? », laConsole. Consulté le: 27 mars 2025. [En ligne]. Disponible sur: <https://laconsole.dev/blog/typescript-quand-pourquoi-utiliser>

[48] P. Devraye, « Understanding RestTemplate in Java Spring: When and How to Use It », Medium. Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://medium.com/@psdevraye/understanding-resttemplate-in-java-spring-when-and-how-to-use-it-79500987a53c>

[49] « What Is Java Spring Boot? | IBM ». Consulté le: 18 mars 2025. [En ligne]. Disponible sur: <https://www.ibm.com/think/topics/java-spring-boot>