

VSX: Vinci Stock eXchange

Projet Spring 2023

Contexte

Vous désirez créer une nouvelle place boursière (stock exchange) qui soit complètement virtuelle. Son fonctionnement sera le suivant.

Compte

Chaque investisseur sur votre plateforme doit posséder un compte pour effectuer ses opérations. Sur un compte, il peut y avoir du cash et des titres. Le cash est une quantité de monnaie dont le prix ne varie pas (c'est une hypothèse simplificatrice par rapport à la réalité, mais c'est mieux pour comprendre). Les titres, quant à eux représentent des quantités d'actions achetées par le propriétaire du compte. Le prix de chacun de ces titres fluctue continuellement en fonction des échanges qui sont réalisés au sein de votre plate-forme.

Par exemple, lorsque madame Lisa MONA ouvre un compte sur votre plateforme, sa banque peut y déposer 10.000 euros (cash), 10 actions AAPL et 5 actions MSFT (titres).

Il est important de noter que votre plateforme permet la "vente a découvert" et les "achats sur marge". Ce qui veut dire qu'il est possible de vendre des titres qu'on ne possède pas et d'acheter des titres avec de l'argent qu'on n'a pas. Il est donc possible d'avoir un compte qui ait un solde de cash négatif, et de posséder un nombre négatif de certains titres.

Ordres

Il y a deux types d'ordre qui peuvent être passés sur votre plateforme, à savoir : des ordres 'MARKET' pour lesquels l'achat ou la vente d'un titre se fait au prix actuel du marché ; et des ordres 'LIMIT' pour lesquels on va spécifier un prix minimum de vente (prix maximum d'achat). Pour chacun de ces deux types d'ordres, on peut se placer de deux 'cotés' : soit vendeur, soit acheteur.

Précision importante :

Un ordre ne porte jamais que sur un titre à la fois. Il n'est pas possible de passer un ordre qui dise je veux vendre 3 AAPL et 5 MSFT. Pour faire ça, il est nécessaire de passer deux ordres distincts.

Par exemple :

Si Jean-Michel veut vendre 5 actions LNRD et qu'il veut vendre chacune d'elles à 100 euro minimum, il passera un ordre 'LIMIT' du coté 'vendeur' et imposera un prix limite de 100 euros par unité.

De la même façon, si Marie-Francine veut acheter 3 LNRD sans imposer de prix limite, elle passera un ordre de type 'MARKET' du coté 'acheteur'.

Fixation des prix et Exécution

Comme dans toute bourse qui se respecte, c'est l'offre et la demande qui fixeront les prix sur votre plateforme. Concrètement cela signifie que votre plateforme va traiter les ordres qu'elle reçoit, et elle va essayer de les satisfaire aussi vite que possible déterminant ainsi les prix qui sont d'application.

De façon cohérente avec ce qui est écrit ci-dessus, il est important de se rendre compte que seule l'exécution d'un ordre 'LIMIT' aura un impact sur les prix. En effet, l'exécution d'un ordre 'MARKET' accepte toujours le prix fixé par les autres acteurs du marché. Par ailleurs, il est possible qu'un ordre soit 'partiellement satisfait' (ce qui veut dire qu'il peut être nécessaire d'avoir plusieurs acheteurs pour satisfaire un gros ordre de vente).

Voici quelques exemples qui devraient vous permettre d'y voir un peu plus clair.

Exemple 1

1. Alice passe un ordre de vente 'LIMIT' pour 5 actions LNRD avec un prix limite de 10 euros par action
2. Bob passe un ordre d'achat 'LIMIT' pour 5 actions LNRD avec un prix maximum d'achat de 13 euros par action.

Le système exécute les ordres de Alice et Bob. Les 5 actions sont vendues au prix de 11,5 euros car c'est le prix qui se trouve à mi-chemin entre les deux limites.

Exemple 2

1. Alice passe un ordre de vente 'LIMIT' pour 5 actions LNRD avec un prix limite de 50 euros par action
2. Bob passe un ordre d'achat 'LIMIT' pour 5 actions LNRD avec un prix maximum d'achat de 10 euros par action.

Le système ne fait rien car il est impossible de satisfaire Alice et Bob. Leurs deux ordres restent donc dans le système jusqu'à ce que d'autres ordres introduits plus tard puissent permettre leur bonne exécution.

Exemple 3

1. Alice passe un ordre de vente 'LIMIT' pour 5 actions LNRD avec un prix limite de 50 euros par action
2. Bob passe un ordre d'achat 'MARKET' pour 5 actions LNRD.

Le système exécute les ordres de Alice et Bob. Les 5 actions sont vendues au prix de 50 euros par action.

Exemple 4

1. Alice passe un ordre de vente 'MARKET' pour 5 actions LNRD.
2. Bob passe un ordre d'achat 'LIMIT' pour 5 actions LNRD. Il fixe son prix d'achat maximum à 50 euros.

Le système exécute les ordres de Alice et Bob. Les 5 actions sont vendues au prix de 50 euros par action.

Exemple 5

1. Alice passe un ordre de vente 'MARKET' pour 5 actions LNRD.
2. Bob passe un ordre d'achat 'MARKET' pour 5 actions LNRD.

Le système exécute les ordres de Alice et Bob. Le prix des actions dépendra du dernier échange de ce type d'action ayant eu lieu. Par exemple, si Cédric avait vendu 3 actions LNRD à Fred au prix de 7 euros chacune, alors le prix utilisé par Alice et Bob serait aussi de 7 euros par action.

Dans le cas (théorique) où les ordres de Alice et Bob seraient les tout premiers ordres relatifs à des actions LNRD sur votre plateforme, alors on décidera *par convention* que le prix est de 1 euro par action (ça ne se passe pas comme ça dans la réalité, mais c'est la logique que vous devez implémenter).

Exemple 6

1. Alice passe un ordre 'LIMIT' pour vendre 200 actions LNRD au prix de 10 euros par actions minimum.
2. Bob passe un ordre 'MARKET' pour acheter 50 actions LNRD.
3. Cédric passe un ordre 'MARKET' pour acheter 50 actions LNRD.
4. Fred passe un ordre 'LIMIT' pour acheter 100 actions LNRD à 11 euros max par unité.

Le système commence par satisfaire l'ordre de Bob en lui vendant 50 actions au prix de 10 euros. L'ordre de Bob est considéré comme *exécuté* et celui de Alice comme *partiellement exécuté*. Le système exécute ensuite l'ordre de Cédric qui achète lui aussi 50 actions à 10 euros. L'ordre de Cédric est *exécuté* et celui de Alice n'est toujours que *partiellement exécuté*. Enfin, le système exécute l'ordre de Fred qui achète 100 actions LNRD à Alice au prix de 10,5 euros chacune. Les ordres de Fred et Alice sont dès lors considérés comme *exécutés* tous les deux.

Exemple 7

1. Alice passe un ordre 'LIMIT' pour acheter 200 actions LNRD au prix de 10 euros par actions maximum
2. Bob passe un ordre 'LIMIT' pour vendre 200 actions LNRD au prix de 100000 euros par actions
3. Cédric passe un ordre 'LIMIT' pour vendre 200 actions LNRD au prix de 9,5 euros par actions minimum.

Le système satisfait l'ordre de Alice en lui vendant les actions de Cédric. L'ordre de Bob est toujours pendant et ne sera satisfait que si quelqu'un passe un ordre d'achat 'MARKET'. Ou si un ordre d'achat 'LIMIT' est passé avec une limite supérieure ou égale à 100000 euros par actions.

Hypothèses Simplificatrices

Par souci de simplicité, on fera les hypothèses suivantes :

- Il n'y a qu'une seule devise (currency) en circulation sur votre plate-forme. Le cash présent sur tous les compte est libellé dans cette devise. De la même façon, le prix de tous les titres échangés sur votre plat-forme est lui aussi libellé dans cette devise.
- Vous pouvez considérer que votre plate-forme boursière est la seule qui existe. Pour fixer les prix des échanges, vous n'êtes pas tenus de considérer d'informations extérieures à votre système.
- On considère qu'il n'y a pas de frais associés à la vente à découvert, ni aux achats sur marge. Lors d'une vente à découvert, il n'est pas nécessaire non plus que votre plateforme trouve un autre investisseur à qui emprunter des titres.

Ce que vous devez faire

Dans le cadre de ce projet, il vous est demandé de développer plusieurs fonctionnalités essentielles pour la plateforme Vinci Stock eXchange. Tout d'abord, vous devrez permettre la création, la suppression et la modification des comptes d'investisseurs. Ces modifications incluent l'ajout de cash ou de titres. Il est également essentiel que les utilisateurs puissent consulter à tout moment l'état de leur compte ainsi que sa valeur totale.

Ensuite, la gestion des ordres est au cœur de votre plateforme. Vous devrez donc permettre la création de nouveaux ordres, la modification de leur statut, qu'ils soient nouveaux, exécutés ou partiellement exécutés. Il est crucial de pouvoir lister tous les ordres et de connaître leur statut actuel. Lorsqu'un ordre est complètement exécuté, une mise à jour de l'état du compte associé doit être effectuée, que ce soit pour ajouter du cash ou pour ajouter ou retirer des titres.

Un autre aspect fondamental est la fourniture du prix courant des différents titres échangés sur la plateforme. Les utilisateurs doivent pouvoir consulter ces prix en temps réel. De plus, l'exécution correcte des ordres est primordiale. Vous devrez vous assurer que les ordres introduits sont bien exécutés. Suite à l'exécution d'ordres d'achat et de vente, des mises à jour de prix doivent être envoyées. Il est également nécessaire d'informer du statut d'exécution d'un ordre, qu'il soit totalement ou partiellement exécuté.

Vous utiliserez le framework Spring pour développer le backend de votre application en utilisant les concepts et techniques vus au cours. A cet effet, vous utiliserez une architecture micro-services et documenterez l'API de tous les services et toutes les routes créées avec OpenAPI 3.0. Il n'est pas nécessaire de développer un front-end pour votre application.

Tests

Chaque service devra être accompagné d'un fichier de test http comprenant suffisamment de requêtes pour tester efficacement toutes les routes du service. Vous devrez aussi écrire un fichier de test http pour le gateway.

Utilisation de Github

Nous vous demandons de collaborer grâce à Github. Vous devez créer un projet sur Github pour le groupe, sur lequel se trouvera votre projet IntelliJ contenant vos modules (un pour chaque service). Invitez-nous comme maintainers dans ce projet en choisissant comme expiration date le 1 février. Nous utiliserons l'historique des commits en cas de doute sur la participation d'un membre du groupe ; il est donc dans votre intérêt d'utiliser cet outil correctement et régulièrement.

Vous devrez remplir un formulaire mettant en lien votre identifiant Git avec votre nom réel.

Organisation

Pour le mardi 17/10 à 20h :

Remise de la composition des groupes

Pour le vendredi 10/11 à 20h :

Remise de la spécification des services. Fournissez les spécifications des services REST, sous forme de fichiers yaml. Cette soumission sera évaluée par groupe. Dans votre rapport, vous décrirez de façon succincte la responsabilité de chacun de vos services et inclurez un schéma décrivant les interactions entre vos différents services (sur une seule face A4).

Pour le vendredi 01/12 à 20h :

Remise du projet implémenté. Veillez à push la dernière version de votre projet sur votre repo GitLab/Github. Nous ne regarderons pas les commits ayant eu lieu après la deadline. Dans ce document, vous reprendrez la découpe en services de votre livrable précédent et indiquerez la responsabilité de chaque étudiant lors de la phase de réalisation.

Évaluation

La note sera formée d'une partie commune au groupe et d'une partie individuelle, chaque partie comptant pour 50% de la note. Les critères sont les suivants :

Partie commune :

Documentation

La documentation respecte les besoins exprimés par le présent document et la spécification Open API. Le Gateway ainsi que tous les autres services sont spécifiés correctement en OpenAPI.

Architecture

L'architecture respecte les principes de conception en micro-services vue au cours et notamment le principe de responsabilité unique.

Partie individuelle :

- L'implémentation des services dont l'étudiant est responsable respecte leur spécification.
- Le code est de qualité :
 - Documenté
 - Utilise Spring correctement
 - Respecte les conventions Java
 - Les tests sont présents, pertinents et corrects pour chaque service
 - L'étudiant était présent aux séances de projet

Un malus de 20 % sera appliqué par séance durant laquelle l'étudiant n'était pas ou insuffisamment présent.