

Documentation sur le développement

YANG Chen

04/12/2023

Table des matières

Table des matières	1
1 Introduction	2
2 Conception détaillée	2
2.1 Service	2
2.1.1 Registration Service (Service d'Enregistrement)	2
2.1.2 Authentication Service (Service d'Authentification)	2
2.1.3 Security Service (Service de Sécurité)	2
2.1.4 Portfolio Service (Service de Portefeuille)	2
2.1.5 Transaction Service (Service de Transaction)	2
2.1.6 Market Data Service (Service des Données du Marché)	2
2.2 Repository	3
2.2.1 User Repository (Dépôt Utilisateur)	3
2.2.2 Data Repository (Dépôt de Données)	3
2.2.3 Transaction Repository (Dépôt de Transaction)	3
2.2.4 Portfolio Repository (Dépôt de Portefeuille)	3
2.3 Model	3
2.3.1 Analytics Tool (Outil d'Analyse)	3
2.3.2 Performance Monitor (Surveillance de la Performance)	3
2.3.3 User (Utilisateur)	4
2.3.4 Data (Donnée)	4
2.3.5 Authentication Result (Résultat d'Authentification)	4
2.3.6 Registration Result (Résultat d'Inscription)	4
2.3.7 Portfolio (Portefeuille)	4
2.3.8 Asset (Actif)	4
2.3.9 Transaction (Transaction)	4

1 Introduction

Ce document décrit la conception détaillée du système de gestion de portefeuille, mais ne reflète pas la conception des vues et des contrôleurs dans le document, car ceux-ci n'ont pas été conçus jusqu'à présent dans notre processus de développement. Certains outils et constantes encapsulés ne sont pas non plus reflétés dans ce document. De même, la gestion des exceptions n'est que partiellement couverte.

2 Conception détaillée

2.1 Service

2.1.1 Registration Service (Service d'Enregistrement)

Responsabilités : Traite la logique d'enregistrement des nouveaux utilisateurs.

Méthodes :

- **register**(userName : String, password : String, passwordEnsurance : String) : RegistrationResult

Exceptions :

- **IOException** : Si le fichier de stockage des utilisateurs n'est pas accessible.

2.1.2 Authentication Service (Service d'Authentification)

Responsabilités : Traite la logique d'authentification des utilisateurs.

Méthodes :

- **authenticate**(userName : String, password : String) : AuthenticationResult
- **checkPassword**(passwordHash : String, password : String) : Boolean

2.1.3 Security Service (Service de Sécurité)

Responsabilités : Gère les opérations de sécurité et d'autorisation.

Méthodes :

- **encryptData**(plainData : String) : String
- **decryptData**(encryptedData : String) : String

2.1.4 Portfolio Service (Service de Portefeuille)

Responsabilités : Gère les services de portefeuille d'investissement.

Méthodes :

- **getPortfolio**(portfolioId : int) : Portfolio
- **updatePortfolio**(portfolio : Portfolio) : Boolean
- **addPortfolio**(portfolio : Portfolio) : Boolean

2.1.5 Transaction Service (Service de Transaction)

Responsabilités : Gère la logique des transactions.

Méthodes :

- **recordTransaction**(transaction : Transaction) : void
- **getTransactionHistory**(portfolioId : int) : List<Transaction>

2.1.6 Market Data Service (Service des Données du Marché)

Responsabilités : Fournit des services de données de marché.

Méthodes :

- **fetchData**(assetId : int) : List<Asset>
- **fetchHistoricalData**(startTime : Time, endTime : Time) : List<Asset>

Exceptions :

- **IOException** : Si l'API externe est inaccessible.

2.2 Repository

2.2.1 User Repository (Dépôt Utilisateur)

Responsabilités : Gère le stockage et la récupération des données utilisateur.

Méthodes :

- **findAll**() : List<User>
- **save**(user : User) : Boolean
- **findByUserName**(userName : String) : User
- **writeUsersToFile**(users : List<User>) : void

2.2.2 Data Repository (Dépôt de Données)

Responsabilités : Gère le stockage et la récupération des données d'application.

Méthodes :

- **findAll**() : List<Asset>
- **save**(Data : data) : Boolean
- **findByDataTimeStamp**(dataTimeStamp : Time) : Data
- **findByDataTimeStamp**(startTime : Time, endTime : Time) : List<Data>
- **writeDatasToFile**(datas : List<Data>) : void

2.2.3 Transaction Repository (Dépôt de Transaction)

Responsabilités : Gère le stockage et la récupération des données de transaction.

Méthodes :

- **findAll**() : List<Transaction>
- **save**(transaction : Transaction) : Boolean
- **findByPortfolioId**(portfolioId : int) : List<Transaction>
- **writeTransactionsToFile**(transactions : List<Transaction>) : void

2.2.4 Portfolio Repository (Dépôt de Portefeuille)

Responsabilités : Gère le stockage et la récupération des données de portefeuille d'investissement.

Méthodes :

- **findAll**() : List<Portfolio>
- **save**(portfolio : Portfolio) : Boolean
- **findByPortfolioId**(portfolioId : int) : Portfolio
- **writePortfoliosToFile**(portfolios : List<Portfolio>) : void

2.3 Model

2.3.1 Analytics Tool (Outil d'Analyse)

Responsabilités : Fournit des outils d'analyse de données.

2.3.2 Performance Monitor (Surveillance de la Performance)

Responsabilités : Surveille la performance du Service des Données du Marché pour déterminer si l'API externe est trop lente.

2.3.3 User (Utilisateur)

Responsabilités : Représente un utilisateur.

Propriétés :

- **userName** : String
- **userId** : int
- **passwordHash** : String
- **portfolios** : List<Portfolio>

2.3.4 Data (Donnée)

Responsabilités : Représente une donnée collecté.

Propriétés :

- **price** : BigDecimal
- **timeStamp** : Time

2.3.5 Authentication Result (Résultat d'Authentification)

Responsabilités : Représente le résultat d'une tentative d'authentification.

Propriétés :

- **sucess** : Boolean
- **failureType** : AUTHENTICATION_FAILIURE_TYPE
- **user** : User

2.3.6 Registration Result (Résultat d'Inscription)

Responsabilités : Représente le résultat d'une tentative d'inscription.

Propriétés :

- **sucess** : Boolean
- **failureType** : REGISTRATION_FAILIURE_TYPE
- **user** : User

2.3.7 Portfolio (Portefeuille)

Responsabilités : Représente un portefeuille d'investissement.

Propriétés :

- **portfolioId** : int
- **ownerId** : int
- **assets** : List<Asset>

2.3.8 Asset (Actif)

Responsabilités : Représente un actif.

Propriétés :

- **assetId** : int
- **assetName** : String
- **assetType** : ASSET_TYPE
- **quantity** : int
- **price** : BigDecimal
- **value** : BigDecimal

2.3.9 Transaction (Transaction)

Responsabilités : Représente une transaction.

Propriétés :

- **transactionId** : int
- **portfolioId** : int
- **assetId** : int
- **quantity** : int
- **transactionType** : TRANSACTION_TYPE
- **timeStamp** : Time