

Projet : Plateforme de Surveillance et d'Analyse des Marchés de Cryptomonnaies

Description générale

Le projet consiste à concevoir et développer une plateforme complète de suivi, d'analyse et de prévision des marchés de cryptomonnaies. Cette plateforme reposera sur deux applications complémentaires :

1. Une application console de collecte et de gestion des données. Cette application s'exécutera en continu et collectera périodiquement des données à partir d'APIs publiques (ex. CoinCap, CoinGecko, Binance ou Kraken). Elle extraira et enregistrera les informations clés des marchés (prix en temps réel, volume d'échange, capitalisation, variation horaire/journalière, indicateurs techniques, etc.) dans une base de données (ex. PostgreSQL, InfluxDB ou MongoDB).
Un système de planification (scheduler) permettra d'automatiser la fréquence de collecte et d'assurer la tolérance aux pannes (via une file de tâches comme Celery ou RabbitMQ).
2. Une application web interactive d'analyse et de visualisation. Cette application permettra aux utilisateurs de se connecter, consulter et analyser les données collectées à travers des tableaux de bord interactifs. Elle offrira :
 - Des visualisations dynamiques (graphiques en courbes, chandeliers, heatmaps, indicateurs techniques personnalisés).
 - Des filtres temporels et de cryptomonnaies (sélection de plages de temps, comparaison de plusieurs actifs).
 - Des alertes personnalisées configurables : seuils de prix, variations...
 - Des notifications automatiques envoyées par e-mail ou via une API tierce (ex. Discord).
 - Un module de prévision reposant sur des modèles simples (ex. moyennes mobiles, régression linéaire...).
 - Un mode "portefeuille virtuel" pour permettre à l'utilisateur de simuler des achats/ventes et d'évaluer ses performances hypothétiques.
 - Un système d'authentification sécurisé (JWT, OAuth2) et une gestion des rôles (administrateur, utilisateur standard).

Le développement de ce projet mettra l'accent sur les bonnes pratiques du processus de développement logiciel. Il impliquera l'application de méthodologies agiles pour la planification et la gestion des tâches, l'intégration de techniques de test avancées comme les tests unitaires, les tests de performance et de sécurité, ainsi que l'adoption de pratiques DevOps telles que l'intégration et le déploiement continus. L'utilisation de conteneurs Docker et l'orchestration avec Kubernetes sera utilisée pour déployer l'application de manière efficace et scalable.

Quels sont les attendus

- **Processus de développement logiciel**
 - Adoption d'une méthodologie Agile (Scrum ou Kanban).
 - Mise en place d'un backlog produit, de user stories détaillées et de sprints planifiés.
 - Utilisation d'outils collaboratifs tels que Jira, Trello ou GitLab Boards pour le suivi des tâches et priorités.
 - Documentation continue (README, diagrammes UML, schémas d'architecture).
- **Technique de tests**
 - Écriture de tests unitaires et tests d'intégration sur les fonctionnalités critiques (collecte, stockage, visualisation).
 - Usage de mocks pour simuler les appels aux APIs externes.
 - Mesure de la couverture de tests et analyse qualité via SonarQube.
 - Tests de performance (Locust, k6) pour évaluer la scalabilité et la latence du système.
 - Tests de sécurité automatisés avec OWASP ZAP ou Snyk.
- **DevOps et CI/CD**
 - Mise en place d'un pipeline CI/CD (GitHub Actions ou GitLab CI).
 - Dockerisation complète des services (collecte, API backend, frontend web, base de données).
 - Déploiement automatisé sur Kubernetes (via Minikube, Kind ou cluster cloud).
 - Surveillance et observabilité via Prometheus et Grafana (métriques, logs, alertes).
 - Sauvegardes automatiques de la base de données et des configurations.

Quels sont les livrables

- Code source complet et versionné sur GitHub (avec historique de commits cohérent)
- Documentation technique et utilisateur
- Tableau de bord Agile (suivi des sprints, user stories, issues).
- Pipeline CI/CD fonctionnel (tests, build, déploiement automatisé).
- Rapports de tests : unitaires, intégration, performance et sécurité
- Démonstration fonctionnelle

Exemple pour récupérer des données sur les cryptomonnaies

Exemple de commande permettant de récupérer des données sur les cryptomonnaies. Attention, il y a des limitations sur le nombre de requêtes. N'hésitez pas à consulter ce document pour en savoir plus : <https://docs.coincap.io/>

```
curl --location 'api.coincap.io/v2/assets'
```