

SPEC FONCTIONNELLE – MarketContextAPI (Collecte des bougies Binance.Net)

1. Objectif de la fonctionnalité

Fournir un composant de surveillance temps réel du marché crypto, permettant : - La collecte continue des données de marché (bougies OHLCV) - L'analyse comportementale (prix + volume) par symbole - Le calcul d'un score de contexte (0-100) - Le déclenchement éventuel d'un signal d'achat si un pattern valide est détecté

Cette API constitue le cœur d'analyse du moteur de trading mid-frequency.

2. Description fonctionnelle

2.1 Flux de données en entrée

- Source : WebSocket Binance Spot (via Binance.Net)
- Données collectées :
 - Prix d'ouverture, clôture, plus haut, plus bas (OHLC)
 - Volume
- Timeframes gérés : M1, M5, M15
- Symboles : multi-symboles (BTC/USDT, ETH/USDT, etc.)
- Collecte uniquement des bougies clôturées (pas de stream partiel)
- Les symboles suivis sont initialement extraits via **GetExchangeInfoAsync**
- Les métadonnées complémentaires comme **le ticker (24h)** et **l'ordre book (profondeur)** sont également récupérées en amont via appel REST pour enrichir le contexte

2.2 Stockage temporaire

- Conservation des **20 dernières bougies par symbole et timeframe**
- Mémoire tampon glissante, sans persistance
- Accès en lecture interne pour les modules d'analyse

2.3 Analyse comportementale

- Objectif : identifier des situations de marché propices à un trade directionnel
- Patterns reconnus :
 - **Accumulation** : prix stables / baissiers avec volume croissant
 - **Breakout** : explosion haussière des prix + pic de volume
 - **Distribution** : hausse prolongée suivie de saturation du volume
 - **Rejet** : grande mèche avec volume anormalement élevé
- Application de règles sur les 10 à 20 dernières bougies

2.4 Calcul de score

- Score sur une échelle de 0 à 100
- Chaque symbole a un score calculé pour chaque timeframe (M1, M5, M15)
- Seuil de déclenchement configurable (ex. : 75 par défaut)
- Exemple : si score ETH/USDT M5 = 82, un signal peut être généré

2.5 Déclenchement de signal

- Si le score dépasse le seuil défini ET qu'aucun trade actif n'existe sur ce symbole
- Création d'un objet **TradeSignal** avec :
 - Symbole
 - Timeframe
 - Score
 - Pattern reconnu
 - Timestamp
- Le signal est ensuite transmis à l'engine de trading pour exécution éventuelle

2.6 Exposition API REST

- Endpoint GET `/symbols/contexts`
- Retourne la liste des scores par symbole
- Informations affichées : symbole, timeframe, score, pattern, statut du signal

2.7 Événements internes générés

- `KlineReceived` : à chaque bougie reçue
- `ContextScoreCalculated` : à chaque score recalculé
- `SignalGenerated` : à chaque signal déclenché
- Ces événements sont utilisés en interne ou envoyés vers d'autres modules (ex : TradeEngine)

3. Règles fonctionnelles

- Aucun signal ne doit être généré si un trade est actif sur le symbole concerné
- La détection de pattern ne doit pas utiliser d'indicateurs externes (RSI, MACD, etc.)
- Le système doit être réactif (score recalculé en < 1s après réception de la bougie)
- Le nombre de symboles actifs doit être limité (paramètre global configurable, ex : 10 max)
- Les informations initiales des symboles (lot size, min qty, tick size) sont extraites via **GetExchangeInfoAsync** et doivent être mises à jour régulièrement (ex : chaque heure)

4. Cas d'usage principaux

4.1 Observation multi-symboles

- Le système suit 5 paires (ex : BTC, ETH, SOL, AVAX, LINK)
- Chaque minute, une nouvelle bougie M1 est reçue et stockée
- L'analyse s'exécute automatiquement à chaque réception

4.2 Génération de signal

- Les 15 dernières bougies ETH/USDT M5 montrent une stabilisation des prix + volume montant
- Pattern détecté : accumulation
- Score calculé : 79
- TradeSignal généré car le seuil est > 75 et aucun trade ETH actif

4.3 Aucune action prise

- Le score est de 62 → pas de signal
 - Un signal aurait pu être généré, mais une position est déjà ouverte sur le symbole concerné → bloqué
-

5. Données en sortie

- Objet `ContextScore` avec symbole, score, pattern, timestamp
 - Objet `TradeSignal` avec symbole, score, raison, horodatage
 - Liste REST `GET /symbols/contexts`
-

6. Exigences non fonctionnelles liées

- 100 % en mémoire
 - Réponse API en < 300 ms
 - Pas de persistance des bougies collectées
 - Utilisation exclusive de la lib Binance.Net
-

7. Remarques

- Cette feature est le socle de l'architecture événementielle de la plateforme
 - Elle sera le point d'entrée pour le futur moteur IA (V2)
 - Les seuils, timeframes et patterns seront configurables dans une version suivante
 - L'ordre book et le ticker 24h seront disponibles via endpoints REST pour contextualiser les mouvements
-

8. Priorité et livraison

- **Priorité haute** (blocage des features de décision et d'exécution sans elle)
- Livraison prévue en **V1**