

# Spécification Fonctionnelle - MarketDataCollector (Mise à jour)

## 1. Objectif du service

Le service MarketDataCollector a pour rôle de : - Écouter en temps réel les bougies clôturées (candlesticks) sur Binance Spot via WebSocket. - Surveiller des symboles actifs pour plusieurs timeframes (M1, M5, M15). - Comparer chaque nouvelle bougie à l'état précédent connu pour le couple (symbol, timeframe). - Détecter un ou plusieurs changements significatifs selon des critères définis. - Émettre un événement `MarketDataChangeEvent` uniquement si un changement est détecté.

## 2. Périmètre fonctionnel

Le service n'a aucune responsabilité sur : - Le recalcul de statistiques complexes. - Le stockage long terme ou persistance de données de marché. - La génération de journaux d'opérations ou rapports client.

Le service partage le **même Hub SignalR** que le module SymbolDownloader pour publier les événements vers les clients abonnés par symbole.

## 3. Déclenchement de la rotation

- La rotation WebSocket est effectuée toutes les 12 heures.
- Elle comprend la récupération des symboles actifs via l'API interne SymbolDownloader, la fermeture de l'ancienne session WS, et la création d'une nouvelle session WS multiplexée.

## 4. Critères de détection

- VolumeSpike : volume actuel comparé à la moyenne glissante des X dernières bougies.
- PriceJump : variation du Close vs Close précédent.
- VolatilitySpike : (High - Low) comparé à sa moyenne glissante.
- Autres critères à venir : Breakout, TrendReversal (non activés dans cette version).

## 5. Structure de l'événement `MarketDataChangeEvent`

L'événement contient : - Symbol : identifiant du symbole (ex: BTCUSDT) - Timeframe : M1, M5, M15 - Timestamp : horodatage UTC - ChangeTypes : liste de types de détection activés - Candle : données OHLCV et Open/Close time - Context : moyennes et ratios glissants - Source : "MarketDataCollector"

## 6. Fréquence et tolérance

- Un seul événement est émis par (symbol, timeframe) et par bougie.
- Aucune garantie de persistance ou de relecture n'est prévue.
- **SignalR est utilisé en mode fire-and-forget** : les pertes d'événements sont acceptables dans le cadre de ce module.

## 7. Données internes conservées

- Dernière bougie clôturée connue.
- Buffers circulaires FIFO (Volume, Close, High-Low) par (symbol, timeframe).

## 8. Configuration

- Tous les seuils, tailles de fenêtre et intervalles sont configurables via `appsettings.json`.
- Les événements SignalR sont envoyés à un **groupe par symbole**.

## 9. Communication

- Le service émet uniquement vers SignalR Hub via groupes identifiés par le symbole.
- L'appel à l'API SymbolDownloader est interne et direct (pas d'appel REST dans cette intégration).

## 10. Exclusions explicites

- Pas d'analyse de tendance globale.
- Pas de persistance disque.
- Pas de retry SignalR.
- Pas de sécurisation du hub pour le POC (auth à gérer plus tard).

---

Souhaites-tu que je passe à la version technique maintenant ?