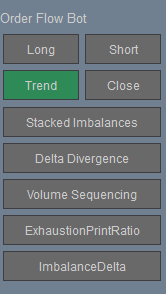
Bonjour,

J’espère que vous allez bien,

Ce code présente un tool que j’utilise sur Ninja Trader dont j’ai activement contribué.

Pour faire fonctionner le bot, il suffit de télécharger Ninja Trader, activé les volumétriques Data et l’importer en tant que Adds-on.



les Imbalances sur les marchés peuvent présenter une opportunité pour les traders de tirer parti des déséquilibres entre les ordres d'achat et de vente. En identifiant et en analysant ces déséquilibres, le bot peut détecter des zones où une pression d'achat ou de vente significative pourrait entraîner des mouvements de prix importants.

En pratique, le bot utilise diverses stratégies basées sur l'analyse des données de flux d'ordres (Order Flow) pour repérer ces imbalances. Il peut alors exécuter des ordres automatiquement en réponse à ces signaux, optimisant ainsi les chances de profiter des opportunités de trading tout en minimisant les risques.

**Fonctionnement général du bot**

1. **Analyse des données de marché :** Le bot collecte et analyse en temps réel les données de volume, de prix et d'ordre pour identifier les imbalances.
2. **Détection des opportunités :** En utilisant des algorithmes spécifiques, le bot détecte les situations où les imbalances sont significatives et susceptibles de précéder des mouvements de prix.
3. **Exécution des stratégies :** Selon les imbalances détectées, le bot applique des stratégies prédéfinies, telles que le Delta Divergence, l'Exhaustion Print Ratio, ou le Volume Sequencing.
4. **Gestion des risques :** Le bot inclut des mécanismes de gestion des risques pour protéger le capital, comme des stops et des limites de position.
5. **Adaptabilité :** Le bot peut être configuré et ajusté pour s'adapter aux différentes conditions de marché et aux préférences du trader.

Cette approche peut également être appliquée au marché des cryptomonnaies et, avec l'intégration du machine learning, elle pourrait offrir de réelles opportunités de profit.

Il est divisé en 5 parties :

* **Common (Propriétés et Enums)**

Cette partie contient des définitions partagées et des propriétés utilisées dans toute l'application.

OrderFlowBot/Common/OrderFlowBotEnums.cs

OrderFlowBot/Common/OrderFlowBotProperties.cs

OrderFlowBot/Common/OrderFlowBotState.cs

* **Databar**

Cette partie gère les "DataBars", qui sont des unités de données de trading. Elle inclut également des dépendances pour des calculs spécifiques.

OrderFlowBot/DataBar/OrderFlowBotDataBar.cs

OrderFlowBot/DataBar/OrderFlowBotDataBars.cs

Dépendances :

OrderFlowBot/DataBar/Dependencies/Deltas.cs

OrderFlowBot/DataBar/Dependencies/Imbalances.cs

OrderFlowBot/DataBar/Dependencies/Prices.cs

OrderFlowBot/DataBar/Dependencies/Ratios.cs

OrderFlowBot/DataBar/Dependencies/Volumes.cs

* **Stratégies et Indicateurs**

Cette partie contient les stratégies de trading et les indicateurs utilisés par l'application. Elle inclut des contrôleurs, des configurations, des bases de stratégies, des interfaces, ainsi que des implémentations spécifiques pour les indicateurs et les stratégies.

StrategiesIndicators :

OrderFlowBot/StrategiesIndicators/StrategiesController.cs

OrderFlowBot/StrategiesIndicators/StrategiesIndicatorsConfig.cs

OrderFlowBot/StrategiesIndicators/StrategyBase.cs

OrderFlowBot/StrategiesIndicators/StrategyInterface.cs

Indicators :

OrderFlowBot/StrategiesIndicators/Indicators/Ratios.cs

OrderFlowBot/StrategiesIndicators/Indicators/RatiosLastExhaustionAbsorptionPrice.cs

OrderFlowBot/StrategiesIndicators/Indicators/SinglePrint.cs

Strategies :

OrderFlowBot/StrategiesIndicators/Strategies/DeltaDivergence.cs

OrderFlowBot/StrategiesIndicators/Strategies/ExhaustionPrintRatio.cs

OrderFlowBot/StrategiesIndicators/Strategies/ImbalanceDelta.cs

OrderFlowBot/StrategiesIndicators/Strategies/StackedImbalances.cs

OrderFlowBot/StrategiesIndicators/Strategies/VolumeSequencing.cs