QUANTITATIVE TRADING
FOR PROFESSIONALS

Upgrade Instructions

AlgoTrader 6.3

19. July 2021

INSTITUTIONAL
ALGORITHMIC TRADING
SOFTWARE FOR
TRADITIONAL SECURITIES
AND **CRYPTO ASSETS**

ALGOTrader

WWW.ALGOTRADER.COM

Upgrade: AlgoTrader 6.3

Below are the instructions on how to upgrade your AlgoTrader 6.2 installation to the 6.3 release.

Entity Changes

Strategy has been renamed to Portfolio

Change getStrategy() / setStrategy() to getPortfolio() / setPortfolio() respectively for domain entities and view objects imported from ch.algotrader.entity.*.

Additionally change `strategy` to `portfolio` in all SQL scripts.

This change also affects the KeyCloak resource: all permissions are now based on the Portfolio resource type. Old installations require manual updates to change all permissions based on the Strategy resource type to the Portfolio resource type.

When upgrading, the SERVER strategy/portfolio will be renamed to DEFAULT portfolio.

Quant strategies and portfolios

Individual strategy instances aren't bound to a Portfolio (previously also called Strategy entity) with the same name as the Strategy anymore. Now strategies can place orders and listen to events belonging to multiple portfolios that they have an access to.

To migrate seamlessly a strategy to 6.3.0 (assuming that there is a Portfolio on DB that has the same name as the strategy that is used), one need to subscribe the strategy to the events of that Portfolio:

Java:

```
@Override

public void onStart(final LifecycleEventVO event) {

  getSubscriptionService().subscribeToPortfolioUpdates(Sets.newHashSet(getStrategyName()));

}
```

Python:

```
  def on_start(self, lifecycle_event):
self.python_to_at_entry_point.subscription_service.subscribe_portfolio([self.STRATEGY_NAME])
```

API:

Strategies that have been using the REST+Websocket API does not have to change anything, as they have already been subscribing to portfolio events individually.

Library Updates

TA4J 0.12 -> 0.13

ta4j-core was upgraded to its latest stable version 0.13. If your strategies are using ta4j transitively, please take a look at the release notes and upgrade accordingly (the breaking changes are minor, only renaming)

Esper Replacement

Esper has been removed from AlgoTrader in 6.3. Any strategies that make use of Esper will have to be adjusted accordingly, or Esper needs to be purchased/licensed separately.

Engine callbacks

Esper engine callbacks have been replaced with a CallbackService which can be retrieved through getCallbackService(). The signature has slightly changed. Every callback now returns a future representing the pending action which can be cancelled. Signatures are similar to the old callback registrations.

Current time retrieval

Until now you could retrieve the current (live/simulation) time through engine.getCurrentTime(). Since Esper has been removed, current time retrieval is now possible through TimeProvider and its getCurrentTime() method. One can retrieve the TimeProvider in the strategy just by calling getTimeProvider().

Deprecations/Method Changes

Deprecated Entities

TickVO is now deprecated.
Please subscribe/rely on BidVO, AskVO, BidAskQuoteVO, etc. instead.

Configuration Properties Changes

- statement.aggregateBars was renamed to barAggregation.enabled
- Property historicalData.barSize used to have two meanings
    1. It was used in the HistoricalDataService to specify the bar size that should be downloaded and
    2. It was used in live run to specify the bar size trades should be aggregated to.

Since those two are two different contexts, we're introducing a new property barAggregation.barSize that will be solely used for bar aggregation in live running.

API Changes in Services Available to Strategies

Most methods mentioning strategy got deprecated.
There are equivalents with portfolio in the name, which should be used instead.

API Changes in Services Available to Strategies (HistoricalDataService)

It is now possible to have several historical data services active at the same time. As a consequence, historical service methods now need to specify the accountId to use in the call, e.g. for the downloadHistoricalBars and downloadHistoricalTicks methods.

API Changes in Services Available to Strategies (LookupService)

The methods related to TradingHours gor removed from the LookupService.
Please use the CalendarService methods instead.

API Changes in Services Available to Strategies (PositionService)

The resetPositions() method was removed.

API Changes in Services Available to Strategies (TransactionService)

The resetCashBalances () method was removed.

Changes in the REST interface

New endpoints added to manage portfolio hierarchies:

- GET /portfolio - get list of all available portfolios
- GET /portfolio/hierarchy - get list of all root portfolios along with their hierarchies
- GET /portfolio/upward-hierarchy/{portfolioName} - get list of portfolios upward in hierarchy, including given one
- GET /portfolio/downward-hierarchy/{portfolioName} - get list of portfolios downward in hierarchy, including given one

Modified endpoints using Portfolio instead of Strategy based naming convention

- GET /transaction/daily?strategy={strategyName}&limit={limit}
- GET /marketDataSubscriptionByStrategyAndSecurity/{strategyName}/{securityId}

- GET /marketDataSubscriptionsByStrategy/{strategyName}
- GET /positionsByStrategyName/{strategyName}
- GET /positionBySecurityAndStrategy/{securityId}/{strategyName}
- GET /openPositionsByStrategy/{strategyName}
- GET /openTradeablePositionsByStrategy/{strategyName}
- GET /openPositionsBySecurityAndStrategy/{securityId}/{strategyName}
- GET /openPositionsByStrategyAndType/{strategyName}/{fullClassName:.+}
- GET /openPositionsByStrategyTypeAndUnderlyingType/{strategyName}/{typeFullClassName:.+}/{underlyingTypeFullClassName:.+}
- GET /openPositionsByStrategyAndSecurityFamily/{strategyName}/{securityFamilyId}
- GET /openFXPositionsByStrategy/{strategyName}
- GET /cashBalancesByStrategy/{strategyName}
- PUT /subscription/marketdata/subscribe?strategyName={strategyName}&securityId={securityId}&accountId={accountId}
- PUT /subscription/marketdata/unsubscribe?strategyName={strategyName}&securityId={securityId}&accountId={accountId}
- PUT /subscription/marketdata/subscribeToCurrency?strategyName={strategyName}&currency={currency}
- PUT /subscription/marketdata/orderBook?strategyName={strategyName}&securityId={securityId}&accountId={accountId}
- PUT /subscription/marketdata/orderBook/exclusively?strategyName={strategyName}&securityId={securityId}&accountId={accountId}
- DELETE /subscription/marketdata/orderBook?strategyName={strategyName}&securityId={securityId}&accountId={accountId}
- DELETE /subscription/marketdata/orderBook/otherThan?strategyName={strategyName}&securityId={securityId}&accountId={accountId}

- PUT /subscription/marketdata/aggregatedOrderBook?strategyName={strategyName}&symbol={symbol}&securityClass={securityClass}
- DELETE /subscription/marketdata/aggregatedOrderBook?strategyName={strategyName}&symbol={symbol}&securityClass={securityClass}
- GET /subscription/marketdata/subscriptionsByStrategy
- GET /execution/order/openByStrategyAndSecurity/{strategyId}/{securityId}
- GET /execution/order/activeByStrategy/{strategyId:.+}
- POST /execution/cancelAllOrdersByStrategy
- GET /cashBalanceByStrategyName/{strategyName}
- GET /cashBalanceByDateTimeStrategyName/{strategyName}/{dateEpochMillis}
- GET /marketValueByStrategyName/{strategyName}
- GET /marketValueByDateTimeStrategyName/{strategyName}/{dateEpochMillis}
- GET /realizedPLByStrategyName/{strategyName}
- GET /unrealizedPLByStrategyName/{strategyName}
- GET /netLiqValueByStrategyName/{strategyName}
- GET /availableBalanceByStrategyName/{strategyName}
- GET /openPositionsByStrategyName/{strategyName}
- GET /openPositionsByStrategyNameAndDate/{strategyName}/{dateEpochMillis}
- GET /portfolioValueByStrategyName/{strategyName}
- GET /portfolioValueByStrategyName/{strategyName}/{dateEpochMillis}
- GET /balancesByStrategyName/{strategyName}
- GET /fxExposureByStrategyName/{strategyName}
- GET /strategyById/{strategyId}
- GET /strategyByName/{name}

Likewise, the response json schema are now returning portfolioId instead of strategyId.

Changes in the WebSocket interface
Topics with strategy in the name were changed to refer to portfolio, e.g. cash-balance.<StrategyName>.<Id> is now cash-balance.<PortfolioName>.<Id>

Likewise, the response json schema are now returning portfolioId instead of strategyId.

About AlgoTrader

AlgoTrader is the global institutional leader in algorithmic trading and execution infrastructure for both digital and traditional assets.

With its advanced software solutions, AlgoTrader gives banks, hedge funds, brokers, crypto funds and other financial institutions an end-to-end platform for best execution and event-driven alpha generation.

The company's offering includes WIRESWARM, a digital asset trading and connectivity platform for banks and brokers, AlgoTrader QUANT, an end-to-end quantitative trading solution with automated trade signal generation and order execution, and AlgoTrader OEMS, a state-of-the-art Order and Execution Management System for systematic and discretionary buy-side institutions with managed connectivity to over 400 liquidity venues. AlgoTrader's interdisciplinary team of quantitative strategy developers, senior software engineers and financial and crypto experts is shaping institutional trading.

Founded in Switzerland and based in Zurich, New York and Singapore, AlgoTrader operates globally.

For more information, please visit https://www.algotrader.com.

Contact

E-Mail info@algotrader.com

T +41 44 291 14 85

Follow AlgoTrader on Twitter and LinkedIn.

For inquiries, please contact

Support Team

T +41 44 291 14 85

support@algotrader.com