Detailed Technical Report: Long Straddle Bot on Saxo API

Giacomo Checchi

April 10, 2025

Contents

1 Understanding the Straddle Strategy in Depth

1.1 The Mechanics and Rationale Behind a Long Straddle

A long straddle is a non-directional options strategy implemented when a trader believes that the underlying asset will experience significant volatility in the near future but is uncertain about the direction of the move. This setup involves purchasing both a call and a put option with the same strike price and expiration date.

The rationale behind the strategy lies in its ability to profit from large price swings, which are common around key events like earnings announcements, regulatory approvals, or economic data releases. The net cost of the strategy is the combined premium paid for the call and the put. This upfront investment is the maximum risk the trader faces.

If the price of the underlying rises significantly above the strike, the call option gains intrinsic value and can be sold for a profit, potentially far exceeding the premium paid. Conversely, if the price drops well below the strike, the put option becomes valuable. The position does not require market direction forecasting, only that the move be large enough to exceed the breakeven thresholds.

This makes the long straddle an attractive candidate for algorithmic deployment around predefined volatile events — especially when combined with reliable calendar data like earnings dates.

From a payoff perspective, the profit-loss profile is V-shaped, with maximum loss at the strike price and potentially unlimited gains as the price diverges in either direction. The breakeven points are:

$$\text{Upper BE} = \text{Strike} + (\text{Call Premium} + \text{Put Premium})$$
$$\text{Lower BE} = \text{Strike} - (\text{Call Premium} + \text{Put Premium})$$

In practice, traders consider implied volatility and option greeks to assess the viability of a straddle. A high implied volatility might make the strategy more expensive and harder to profit from unless an extraordinary move occurs. Low IV, on the other hand, could indicate underpricing of future moves, especially before earnings.

1.2 Scenarios, Risks, and Position Management in a Straddle

In any real market scenario, managing a long straddle requires keen monitoring and clear exit criteria. The strategy performs well in three primary scenarios:

1. Large upward moves – the call grows in value, offsetting the cost of both options.
2. Large downward moves – the put appreciates significantly.
3. IV spike post-entry – even if the underlying does not move much, a volatility expansion can make both options more valuable.

However, a trader must also account for the risks:

- **Theta decay:** Both options lose value with time, especially if the underlying stays near the strike.
- **Volatility crush:** After events like earnings, IV may collapse, devaluing the options.
- **Slippage and illiquidity:** Especially near expiry, options may exhibit wide bid-ask spreads.

To manage these risks, traders often define:

- Stop-loss thresholds (e.g., -30%) to prevent full premium loss.
- Profit targets (e.g., +250%) to exit on favorable moves.
- Time-based exits, e.g., exiting by Friday to avoid weekend time decay.

The strategy is most suitable when:

- The trader has access to timely and reliable event data.
- The options are liquid with reasonable spreads.
- The time to expiry is less than one week (for short-term straddles).

In summary, the long straddle is a tool not only for speculation but also for volatility trading and market-neutral strategies. With precise timing and proper execution, it can deliver significant returns, especially when automated with risk control layers.

2 Architecture and Execution Logic of the Straddle Bot

2.1 Overview of API Integration and Data Flow

The straddle bot integrates three core services:

- **TradingEconomics API:** to fetch earnings calendars.
- **Saxo Bank OpenAPI:** to look up instruments, get prices, and place trades.
- **Streamlit:** to build an interactive UI.

The bot initiates its process by querying TradingEconomics for upcoming earnings events in the next 7 days. From the JSON response, it extracts U.S.-listed ticker symbols expected to report. These tickers are then fed into Saxo's reference lookup API to retrieve the UICs, which are required for tradeable instrument interaction.

For each ticker, the bot:

- Gets the current spot price.
- Retrieves available call and put options with expiry set to the current week's Friday.
- Filters for ATM options based on proximity to the spot price.
- Extracts strike price, premiums, UICs of the call/put.
- Computes total cost, breakeven levels, and decision thresholds.

All this data is compiled into a Streamlit dataframe for user inspection. Trades are not automatically triggered but instead require a user to select a ticker and click a confirmation button. This hybrid design ensures control and monitoring while still automating the bulk of the research and decision support.

2.2 Decision Logic: Breakouts, Profit/Loss Management, and Automation

The heart of the bot is its decision-making engine, which evaluates whether a breakout move has occurred or whether exit conditions are met. For each ticker:

- **Breakout Check:** If the price has moved more than 3% outside either breakeven level, it assumes a directional move has occurred. One leg is closed (e.g., if price >upper BE, close put).
- **Stop Loss:** If the combined value of call + put drops by more than 30% from the initial cost, both legs are closed.
- **Take Profit:** If the combined value increases by more than 250%, both legs are closed to lock in gains.
- **Forced Closure:** At certain hours on weekdays (20:00 Friday, 21:00 otherwise), it reminds the user to exit positions.

All orders are placed using Saxo's trading endpoint. The bot constructs market orders with fixed size (1 contract), account key, and UIC. Orders are submitted in real time, and the result is logged in Streamlit's UI.

The bot does not store state or persist data. Each session re-evaluates based on the real-time API responses. This ensures fresh decision-making and minimizes data inconsistencies.

While the bot is designed for clarity and simplicity, it is structured in a modular way so that more features can be added later, such as:

- Dynamic position sizing
- Filtering by implied volatility
- Integration with Telegram or Discord alerts
- Risk-adjusted performance dashboards

In its current form, the bot offers a powerful tool for semi-automated volatility trading with a focus on earnings volatility, encapsulating research, screening, and execution within a clean interface.