Bybit V5 Connector Documentation

Overview

The Bybit V5 connector provides a comprehensive interface to interact with the Bybit V5 API. It allows the DeepAgent Kraken trading bot to authenticate with API keys, fetch account information, get market data, place orders, and manage positions.

Features

- Authentication with API keys
- Fetching account information (balance, positions)
- Getting market data (OHLCV, orderbook)
- Placing orders (market, limit)
- Managing positions (modifying, closing)
- Error handling and rate limiting

Installation

The connector uses the `pybit` library for direct API connections. Make sure it's installed:

```
pip install pybit>=5.0.0
```

Usage

Initialization

```
from app.connectors.bybit.v5_connector import BybitV5Connector

# Initialize with API keys
connector = BybitV5Connector(
    api_key="YOUR_API_KEY",
    api_secret="YOUR_API_SECRET",
    testnet=False  # Set to True for testnet
)

# Initialize the client
await connector.initialize()
```

Fetching Account Information

```
# Get account balances
balances = connector.get_balances()

# Get positions
positions = connector.get_positions(symbol="BTCUSDT")
```

```python
# Get account balance summary
account_info = connector.get_account_balance()
```

Getting Market Data

```python
# Get OHLCV data
ohlcv = connector.get_ohlcv(
    symbol="BTCUSDT",
    interval="1",  # 1 minute
    limit=100
)

# Get orderbook
orderbook = connector.get_orderbook(
    symbol="BTCUSDT",
    limit=50
)

# Get market data summary
market_data = connector.get_market_data(symbol="BTCUSDT")
```

Placing Orders

```python
# Place a market order
market_order = connector.place_order(
    symbol="BTCUSDT",
    side="Buy",
    order_type="Market",
    qty=0.01
)

# Place a limit order
limit_order = connector.place_order(
    symbol="BTCUSDT",
    side="Buy",
    order_type="Limit",
    qty=0.01,
    price=30000
)

# Place an order with take profit and stop loss
order_with_tp_sl = connector.place_order(
    symbol="BTCUSDT",
    side="Buy",
    order_type="Market",
    qty=0.01,
    take_profit=32000,
    stop_loss=28000
)
```

Managing Orders and Positions

```
# Modify an order
modified_order = connector.modify_order(
    symbol="BTCUSDT",
    order_id="ORDER_ID",
    price=31000,
    qty=0.02
)

# Cancel an order
cancelled_order = connector.cancel_order(
    symbol="BTCUSDT",
    order_id="ORDER_ID"
)

# Cancel all orders
cancelled_all = connector.cancel_all_orders(symbol="BTCUSDT")

# Close a position
closed_position = connector.close_position(symbol="BTCUSDT")
```

Executing Trades from Signals

```
# Execute a trade based on a signal
signal = {
    "symbol": "BTCUSDT",
    "side": "Buy",
    "order_type": "Market",
    "qty": 0.01,
    "price": None,
    "take_profit": 32000,
    "stop_loss": 28000
}

trade_result = connector.execute_trade(signal)
```

Error Handling and Rate Limiting

The connector includes built-in error handling and rate limiting features:

1. **Rate Limit Handling**: The connector automatically handles rate limit errors by implementing exponential backoff when rate limits are hit.
2. **Retry Mechanism**: Critical operations are decorated with retry logic to handle temporary connection issues.
3. **Error Logging**: Comprehensive error logging provides visibility into API interactions.

Integration with DeepAgent Kraken

The Bybit V5 connector is integrated into the DeepAgent Kraken trading bot through the connector factory:

```python
# In app/connectors/connector_factory.py
def get_exchange_connector(exchange_name, priority=ExchangePriority.PRIMARY):
    if exchange_name.lower() == "bybit_v5":
        from app.connectors.bybit.v5_connector import BybitV5Connector
        return BybitV5Connector(priority=priority)
    # ...
```

To use the Bybit V5 connector in your trading strategies, simply request it from the factory:

```python
from app.connectors.connector_factory import get_exchange_connector
from app.core.config import ExchangePriority

# Get the Bybit V5 connector
connector = get_exchange_connector("bybit_v5", priority=ExchangePriority.PRIMARY)
await connector.initialize()

# Use the connector in your strategy
market_data = connector.get_market_data("BTCUSDT")
```

Configuration

The connector uses the following configuration settings from the environment:

- `BYBIT_API_KEY`: Your Bybit API key
- `BYBIT_API_SECRET`: Your Bybit API secret
- `BYBIT_TESTNET`: Whether to use the testnet API (True/False)

These settings can be overridden when initializing the connector.