

Binance Futures Testnet Trading Bot - Final Report (with Mock Screenshots)

Purpose & Scope

Analytical report describing the bot's design, responsibilities, and limitations.

System Architecture

Modular CLI scripts for each order type (market, limit, advanced). Shared utilities handle .env loading, authentication, validation, and logging.

Market Orders

Market orders execute immediately at market price. Below is a mock terminal run showing quantity adjustment (+confirmation) and a filled market order.

```
$ python src/market_orders.py --symbol BTCUSDT --side BUY --quantity 0.001
[2025-08-09 10:36:40 UTC] INFO   Loaded API keys from .env
[2025-08-09 10:36:40 UTC] WARNING Quantity 0.001 below Binance minimum notional (required 0.00171 BTC).
[2025-08-09 10:36:40 UTC] PROMPT Adjust to 0.00171? (y/n): y
[2025-08-09 10:36:40 UTC] INFO   Quantity adjusted to 0.00171
[2025-08-09 10:36:40 UTC] INFO   Placing MARKET order: symbol=BTCUSDT side=BUY quantity=0.00171
[2025-08-09 10:36:40 UTC] RESPONSE {'symbol': 'BTCUSDT', 'orderId': 12345678, 'status': 'FILLED', 'side': 'BUY', 'orderQty': '0.00171', 'executedQty': '0.00171', 'avgPrice': '58643.31'}
```

Limit Orders

Limit orders execute at a specified price. The example demonstrates the minimum notional adjustment prompt and a NEW limit order response.

```
$ python src/limit_orders.py --symbol BTCUSDT --side SELL --quantity 0.001 --price 59000
[2025-08-09 10:36:40 UTC] INFO Loaded API keys from .env
[2025-08-09 10:36:40 UTC] WARNING Quantity 0.001 below Binance minimum notional (required 0.00169 BTC).
[2025-08-09 10:36:40 UTC] PROMPT Adjust to 0.00169? (y/n): y
[2025-08-09 10:36:40 UTC] INFO Quantity adjusted to 0.00169
[2025-08-09 10:36:40 UTC] INFO Placing LIMIT order: symbol=BTCUSDT side=SELL price=59000 quantity=0.00169
[2025-08-09 10:36:40 UTC] RESPONSE
{'symbol': 'BTCUSDT', 'orderId': 12345679, 'status': 'NEW', 'type': 'LIMIT', 'price': '59000', 'origQty': '0.00169'}
```

\$

OCO Orders

OCO places a take-profit limit and a stop-market stop-loss; when one triggers the other is cancelled. The mock shows a stop triggering and cancelling the TP.

```
$ python src/advanced/oco.py --symbol BTCUSDT --side SELL --quantity 0.002 --take-price 59000 --stop-price 57000
[2025-08-09 10:36:40 UTC] INFO Loaded API keys from .env
[2025-08-09 10:36:40 UTC] INFO Placing TAKE-PROFIT (LIMIT) order: price=59000 qty=0.002
[2025-08-09 10:36:40 UTC] INFO Placing STOP-MARKET order: stopPrice=57000 qty=0.002
[2025-08-09 10:36:40 UTC] RESPONSE TAKE-PROFIT orderId=22334455 STOP orderId=22334456
[2025-08-09 10:36:40 UTC] EVENT STOP order filled at 56980.00, cancelling TAKE-PROFIT 22334455
[2025-08-09 10:36:40 UTC] INFO TAKE-PROFIT 22334455 canceled successfully
```

\$

TWAP Orders

TWAP breaks a large order into slices; the mock shows 5 slices executed sequentially with individual fill reports.

```
$ python src/advanced/twap.py --symbol BTCUSDT --side BUY --total-qty 0.01 --slices 5 --duration 60
[2025-08-09 10:36:40 UTC] INFO Loaded API keys from .env
[2025-08-09 10:36:40 UTC] INFO TWAP starting: total=0.01 BTC slices=5 interval=12s qty_per_slice=0.002
[2025-08-09 10:36:40 UTC] INFO Slice 1 placed: orderId=33110011 qty=0.002 status=FILLED avgPrice=58640.12
[2025-08-09 10:36:40 UTC] INFO Slice 2 placed: orderId=33110012 qty=0.002 status=FILLED avgPrice=58642.50
[2025-08-09 10:36:40 UTC] INFO Slice 3 placed: orderId=33110013 qty=0.002 status=FILLED avgPrice=58644.00
[2025-08-09 10:36:40 UTC] INFO Slice 4 placed: orderId=33110014 qty=0.002 status=FILLED avgPrice=58645.75
[2025-08-09 10:36:40 UTC] INFO Slice 5 placed: orderId=33110015 qty=0.002 status=FILLED avgPrice=58647.10
[2025-08-09 10:36:40 UTC] INFO TWAP completed: total_executed=0.01 BTC
```

\$

Validation & Error Handling

The bot validates symbol, quantity (min notional), and logical relationships (e.g., OCO stop vs take), and logs API errors.

Logging

All requests, responses, adjustments, and errors are logged to bot.log for auditability.

Limitations & Recommendations

Testnet-only by default; recommend websocket integration, exchange filter fetching, and more advanced strategy tooling for production.

Conclusion

A robust foundation for futures automation in a test environment, suitable for extension and production hardening.