

Quantum Trading System

1. Core Engine (6 File)

File	Descrizione	Codice Esempio
app/core/engine/__init__.py	Inizializza i motori	<pre>from .trading_engine import *</pre>
app/core/engine/trading_engine.py	Motore principale	<pre>python\nclass TradingEngine:\n\tdef execute(self):\n\t\ttqc = QuantumCircuit(5)</pre>
app/core/engine/risk_engine.py	Gestione rischio quantistico	<pre>python\ndef qvar(positions):\n\treturn sum(p['exposure']**2 for p in positions)**0.5</pre>
app/core/engine/backtest_engine.py	Motore per backtesting	<pre>python\ndef run_backtest(strategy):\n\treturn strategy.calculate_pnl()</pre>
app/core/engine/execuation_engine.py	Esecuzione ordini	<pre>python\ndef send_order(exchange, order):\n\treturn exchange.execute(order)</pre>
app/core/engine/quantum_optimizer.py	Ottimizzazione portfolio	<pre>python\ndef optimize_weights(returns):\n\treturn cvxpy.Minimize(risk)</pre>

2. Analytics Module (6 File)

File	Descrizione	Formula/Codice
app/core/analytics/__init__.py	Inizializza analytics	<pre>from .volatility import *</pre>
app/core/analytics/quantum_oscillator.py	Oscillatore avanzato	$QS = \sigma(\log(p)) \times e^{i\theta} + 0.618$
app/core/analytics/pattern_detection.py	Rilevamento pattern	<pre>python\ndef find_triangles(data):\n\treturn scipy.signal.find_peaks(data)</pre>
app/core/analytics/volatility.py	Calcolo volatilità	<pre>python\ndef realized_vol(returns):\n\treturn np.sqrt(252) * np.std(returns)</pre>
app/core/analytics/correlation.py	Matrice correlazioni	<pre>python\ndef quantum_corr_matrix(assets):\n\treturn np.corrcoef(assets.T)</pre>
app/core/analytics/fourier.py	Analisi frequenze	<pre>python\ndef fft_transform(data):\n\treturn np.fft.fft(data)</pre>

3. Utils (6 File)

File	Descrizione	Codice Esempio
app/core/utils/__init__.py	Inizializza utilities	<pre>from .logger import *</pre>
app/core/utils/hybrid_cache.py	Cache multi-strato	<pre>python\ndef get(key):\n\treturn redis.get(key) or qcache[key]</pre>
app/core/utils/circuit_breaker.py	Gestione failover	<pre>python\ndef check():\n\treturn latency < 100</pre>
app/core/utils/quantum_logger.py	Logger quantistico	<pre>python\ndef log_entanglement(level):\n\tlogger.info(f"Entanglement: {level}")</pre>
app/core/utils/serializer.py	Serializzazione dati	<pre>python\ndef serialize(obj):\n\treturn pickle.dumps(obj)</pre>

File	Descrizione	Codice Esempio
app/core/utils/validator.py	Validazione input	python\ndef validate_symbol(symbol):\n\treturn symbol in ALLOWED_SYMBOLS

4. Machine Learning (6 File)

File	Descrizione	Codice Esempio
app/core/ml/__init__.py	Inizializza ML	from .predictor import *
app/core/ml/ensemble_model.py	Modello ensemble	python\nclass Ensemble:\n\tdef predict():\n\t\treturn 0.618*quantum_pred + 0.382*classic_pred
app/core/ml/training_pipeline.py	Pipeline training	python\ndef train(data):\n\tmodel.fit(data)\n\tvalidate(model)
app/core/ml/feature_engineer.py	Feature engineering	python\ndef add_quantum_features(X):\n\tX['entanglement'] = X.apply(calc_entanglement)
app/core/ml/hyperparam_tuner.py	Ottimizzazione parametri	python\ndef tune(params):\n\treturn BayesianOptimization().maximize()
app/core/ml/anomaly_detector.py	Rilevamento anomalie	python\ndef detect(X):\n\treturn IsolationForest().predict(X)

5. Servizi di Mercato (6 File)

File	Descrizione	Codice Esempio
app/services/market/__init__.py	Init servizi mercato	from .realtime import *
app/services/market/data_feed.py	Feed dati live	python\ndef stream_data(symbol):\n\treturn ccxt.binance().fetch_ohlcv(symbol)
app/services/market/historical_data.py	Dati storici	python\ndef get_history(symbol):\n\treturn pd.read_csv(f'{symbol}.csv')
app/services/market/order_book.py	Gestione order book	python\ndef get_depth(symbol):\n\treturn exchange.fetch_order_book(symbol)
app/services/market/liquidity.py	Analisi liquidità	python\ndef calc_spread(bid, ask):\n\treturn ask - bid
app/services/market/volatility.py	Volatilità in tempo reale	python\ndef update_volatility():\n\treturn np.std(last_100_prices)

6. Servizi Ausiliari (6 File)

File	Descrizione	Codice Esempio
app/services/news/__init__.py	Init servizi news	from .sentiment import *
app/services/news/sentiment.py	Analisi sentiment	python\ndef get_sentiment(text):\n\treturn TextBlob(text).sentiment.polarity
app/services/news/aggregator.py	Aggregatore notizie	python\ndef fetch_rss(feed_url):\n\treturn feedparser.parse(feed_url)
app/services/notification/__init__.py	Init notifiche	from .alerts import *
	Sistema di	

app/services/notification/alerts.py	allerta	python\ndef trigger_alert(msg):\n\tsmtp.send(msg)
app/services/notification/priority.py	Gestione priorità	python\ndef prioritize(alert):\n\treturn heapq.heappush(queue, alert)

7. API Layer (6 File)

File	Descrizione	Codice Esempio
app/api/v1/__init__.py	Init API v1	from .trading import *
app/api/v1/trading.py	Endpoint trading	python\n@ns.route('/execute')\nclass ExecuteTrade(Resource):\n\tdef post()
app/api/v1/market.py	Endpoint mercato	python\n@ns.route('/ohlcv')\nclass OHLCV(Resource):\n\tdef get()
app/api/schemas/__init__.py	Init schemi	from .trade import *
app/api/schemas/trade.py	Schema trade	python\nclass TradeSchema(Schema):\n\tquantity = fields.Float(required=True)
app/api/schemas/market.py	Schema mercato	python\nclass MarketSchema(Schema):\n\tsymbol = fields.Str(required=True)

8. Modelli e Sicurezza (6 File)

File	Descrizione	Codice Esempio
app/models/__init__.py	Init modelli	from .portfolio import *
app/models/portfolio.py	Modello portfolio	python\ndef rebalance():\n\treturn optimize_weights(returns)
app/models/asset.py	Modello asset	python\nclass Asset:\n\tdef __init__(self, symbol):\n\t\tself.symbol = symbol
app/security/__init__.py	Init sicurezza	from .auth import *
app/security/auth.py	Autenticazione	python\ndef verify_jwt(token):\n\treturn jwt.decode(token, key)
app/security/encryption.py	Crittografia	python\ndef encrypt(data):\n\treturn Fernet(key).encrypt(data)

9. Infrastruttura e Testing (7 File)

File	Descrizione	Codice Esempio
app/infrastructure/db/__init__.py	Init database	from .models import *
app/infrastructure/db/models.py	Modelli DB	python\nclass Trade(Base):\n\t__tablename__ = 'trades'
app/infrastructure/monitoring/metrics.py	Metriche	python\ntrades_counter = Counter('trades_total', 'Total trades')
tests/unit/test_engine.py	Test engine	python\ndef test_execute_order():\n\tassert result['status'] == 'filled'
tests/integration/test_api.py	Test API	python\ndef test_trade_endpoint():\n\tresponse = client.post('/trade')
scripts/deploy.py	Deployment	python\ndef deploy():\n\tkubectl.apply('deployment.yaml')
scripts/data_pipeline.py	ETL dati	python\ndef process_data():\n\tdf = pd.read_csv('data.csv')

