RISKCORE Week 2 Agenda

Data Ingestion Layer

Week of January 12, 2026 - January 16, 2026

Week Goal

Build the complete data ingestion layer: REST API endpoints for positions and trades, file upload handling (CSV/Excel), and FIX protocol message parsing.

Monday - Jan 12: FastAPI Foundation

Tasks:

- [ ] Set up FastAPI application structure
- [ ] Create backend/main.py entry point
- [ ] Configure CORS, middleware, error handling
- [ ] Set up database connection pool
- [ ] Create Pydantic models for Position and Trade
- [ ] Write requirements.txt with all dependencies

Deliverable: Running FastAPI app at localhost:8000 with /docs

Tuesday - Jan 13: Position API

Tasks:

- [ ] Create POST /api/v1/positions endpoint
- [ ] Create GET /api/v1/positions endpoint (with filters)
- [ ] Integrate validation pipeline for input validation
- [ ] Integrate security master for identifier resolution
- [ ] Add position to database with proper tenant isolation
- [ ] Write unit tests for position endpoints

Deliverable: Position CRUD working with validation

Wednesday - Jan 14: Trade API + P&L

Tasks:

- [ ] Create POST /api/v1/trades endpoint
- [ ] Create GET /api/v1/trades endpoint (with filters)
- [ ] Implement P&L calculation service
- [ ] Calculate: (quantity x current_price) - (quantity x avg_cost)
- [ ] Update position from trade (quantity, avg_cost)

- [ ] Write unit tests for trade endpoints

Deliverable: Trade ingestion with automatic P&L calculation

Thursday - Jan 15: File Upload

Tasks:

- [ ] Create POST /api/v1/upload endpoint
- [ ] Implement CSV parser with column auto-detection
- [ ] Implement Excel parser (.xlsx, .xls support)
- [ ] Create upload preview response (first 10 rows)
- [ ] Create POST /api/v1/upload/confirm for batch import
- [ ] Write unit tests for file parsing

Deliverable: CSV/Excel upload with preview and batch import

Friday - Jan 16: FIX Protocol + Testing

Tasks:

- [ ] Implement FIX parser using simplefix library
- [ ] Parse ExecutionReport (tag 35=8) messages
- [ ] Parse PositionReport (tag 35=AP) messages
- [ ] Create POST /api/v1/fix endpoint
- [ ] Integration tests for all endpoints
- [ ] Documentation and code cleanup

Deliverable: Complete data ingestion layer ready for Week 3

Technical Specifications

API Endpoints to Build:

Method	Endpoint	Description
POST	/api/v1/positions	Create new position
GET	/api/v1/positions	List positions (filterable)
GET	/api/v1/positions/{id}	Get single position
POST	/api/v1/trades	Create new trade
GET	/api/v1/trades	List trades (filterable)
POST	/api/v1/upload	Upload file, get preview
POST	/api/v1/upload/confirm	Confirm batch import
POST	/api/v1/fix	Parse FIX message

Files to Create:

File	Purpose
backend/main.py	FastAPI application entry point
backend/api/__init__.py	API router initialization
backend/api/positions.py	Position endpoints
backend/api/trades.py	Trade endpoints
backend/api/upload.py	File upload endpoint
backend/api/fix.py	FIX message endpoint
backend/models/position.py	Position Pydantic models
backend/models/trade.py	Trade Pydantic models
backend/services/file_parser.py	CSV/Excel parser
backend/services/fix_parser.py	FIX message parser
backend/services/pnl.py	P&L calculation
requirements.txt	Python dependencies

Dependencies from Week 1:

- validation.py - For validating incoming position/trade data
- security_master.py - For resolving security identifiers
- openfigi.py - For looking up unknown securities
- Database schema - positions, trades, securities tables

Week 2 Acceptance Criteria

[ ] POST /api/v1/positions accepts valid data, returns validation errors for bad data

[ ] POST /api/v1/trades creates trade and updates position quantities

[ ] P&L calculates correctly: (qty x price) - (qty x avg_cost)

[ ] CSV upload auto-detects columns (ticker, quantity, price)

[ ] Excel upload handles .xlsx and .xls formats

[ ] FIX ExecutionReport (35=8) parses correctly

[ ] FIX PositionReport (35=AP) parses correctly

[ ] All endpoints return proper HTTP status codes

[ ] OpenAPI docs available at /docs

[ ] >80% test coverage on new code