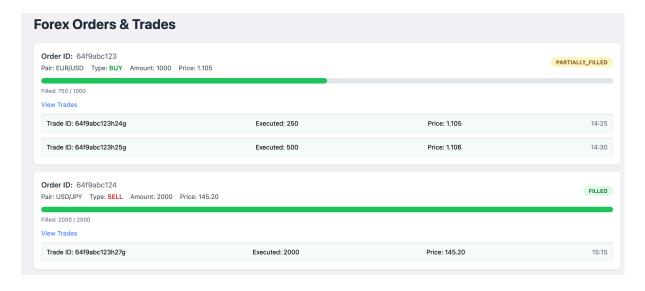
Server Interview - Orders & Trades

This module handles orders with nested trades and real-time updates for a Forex trading system.

The **screenshot** illustrates how orders and trades should appear use it as the basis for your schema and structure design.



Requirements

Orders

- Contain details such as pair, type, amount, price, status.
- · Have a list of trades beneath them.
- Support status progression: PENDING → PARTIALLY_FILLED → FILLED

Trades

- Represent individual executions (amount, price, timestamp).
- Adding trades updates the order's **filled progress** and **status** automatically.

Business Rules

- The sum of all executed trades must not exceed the order's amount.
- If total filled < amount → PARTIALLY_FILLED.
- If total filled = amount → FILLED.
- If adding a trade would overshoot → reject (400).

API Endpoints (JWT required)

Method	Path	Purpose
POST	/orders	Create a new order
GET	/orders	List the current user's orders
GET	/orders/:id	Get a single order + its trades

Method	Path	Purpose
PATCH	/orders/:id/status	Update status manually (e.g., set CANCELLED)
POST	/orders/:id/trades	Add a trade (updates order status automatically)

WebSocket (Real-time Updates)

- Gateway: /ws/orders
- · Broadcast events whenever:
 - o An order's status changes, or
 - A new trade is added.

Example payload:

```
{
"orderId": "64f9abc123",
"event": "TRADE_ADDED",
"trade": {
   "tradeId": "64f9abc123h25g",
   "executedAmount": 500,
   "executedPrice": 1.106
},
   "newStatus": "PARTIALLY_FILLED"
}
```

Simulation (Swagger/Postman)

- 1. Create an order (e.g., BUY EUR/USD 1000).
- 2. Add trades (e.g., 250 + 500 executed).
- 3. Observe status updates in API + real-time events in WebSocket.

Deliverables

Phase 1: High-Level Design (HLD)

Before implementation, provide a short HLD document covering:

- Data modeling approach (how you'll design Orders and nested Trades based on the UI).
- API design flow (how endpoints map to backend operations).
- WebSocket update mechanism (when and what to broadcast).

Phase 2: Implementation

- Implement Orders + Trades module.
- Add API endpoints + validation.
- Add WebSocket gateway + events.
- · Test simulation with Swagger/Postman.