Thought for 5 seconds

**Bulk‑Order Workflow — Complete Endpoint Guide**

Below is the exact call‑sequence your UI should follow, with every endpoint’s purpose and required body parameters (just the names, inside { }) so your front‑end team can implement without any back‑and‑forth.

**1 · Buyer creates a bulk order**

**POST /bulk\_orders/**  {product, quantity, budgeted\_price, [optional aggregator]}  
Creates the initial order.

* If *aggregator* is omitted, the order is broadcast to all available aggregators.
* Response returns the new order in PENDING\_AGGREGATOR state.

**2 · Buyer (or UI) lists their active orders**

**GET /bulk\_orders/**  
Shows the buyer every order that is not *CANCELLED*.

**3 · Buyer views one order’s details**

**GET /bulk\_orders/{orderId}/**  
Full snapshot, including product details, buyer profile, any assigned aggregator, and current status.

**4 · Buyer cancels an order (optional)**

**PATCH /bulk\_orders/{orderId}/cancel/**  
Sets the order to *CANCELLED* and automatically VOIDs any pools that might exist.

**5 · Aggregator discovers available orders**

**GET /bulk\_order/aggregator/orders/**  
Returns:

* Orders already assigned to this aggregator, **plus**
* Broadcast orders that match the aggregator’s category specialties (only if the aggregator is flagged *is\_available=true*).  
  Query‑string filter ?status= lets them narrow by order status if desired.

**6 · Aggregator claims an order**

**PATCH /bulk\_order/aggregator/orders/{orderId}/accept/**  
Locks the order to that aggregator and flips status to AGGREGATOR\_ACCEPTED.  
Only callable if the order is NEW or PENDING\_AGGREGATOR.

**7 · Aggregator creates a pooling round**

**POST /bulk-order-pools/**  {bulk\_order, [optional aggregator\_note]}  
Opens a new pool tied to the accepted order.  
Automatically stamped with OPEN status and the current aggregator profile.

**8 · Aggregator lists all their pools**

**GET /bulk-order-pools/**  
Useful dashboard endpoint; shows every pool the aggregator owns.

**9 · Aggregator (or UI) fetches one pool’s details**

**GET /bulk-order-pools/{poolId}/**  
Includes current participants, total price (zero until finalized), status, and notes.

**10 · Aggregator finalizes the pool**

**PATCH /bulk-order-pools/{poolId}/finalize/**  
Checks that the sum of all *supplier‑accepted* quantities ≤ order quantity, computes total\_price, and sets status to FINALIZED.  
After this, the buyer can decide.

**11 · Aggregator lists participants in a pool**

**GET /bulk\_order/pools/{poolId}/participants/**  
Returns every supplier entry with quantity, unit\_price, and whether the supplier has accepted.

**12 · Aggregator adds a supplier participant**

**POST /bulk\_order/pools/{poolId}/participants/**  {supplier, quantity, unit\_price}  
Creates a new participant line.  
supplier\_accepted is initially **false** (pending supplier action).

**13 · Aggregator updates or removes a participant**

* **PATCH** /bulk\_order/pools/{poolId}/participants/{participantId}/  {quantity, unit\_price}  
  Updates quantity and/or price and resets supplier\_accepted=false.
* **DELETE** /bulk\_order/pools/{poolId}/participants/{participantId}/  
  Removes the line completely.  
  *Only allowed while the pool status is OPEN.*

**14 · Supplier responds to an invitation**

**PATCH /bulk\_order/supplier/participants/{participantId}/respond/**  {action, [proposed\_quantity], [proposed\_price]}  
action must be **accept**, **decline**, or **propose**.

* *accept* → sets supplier\_accepted=true.
* *decline* → deletes the participant record.
* *propose* → updates quantity/price and leaves supplier\_accepted=false for aggregator review.

**15 · Buyer accepts one finalized pool**

**PATCH /bulk\_order\_pools/{poolId}/accept-by-buyer/**  
Marks that pool ACCEPTED\_BY\_BUYER, voids all sibling pools, and flips the order status to BUYER\_ACCEPTED.

**16 · Buyer pays and completes the order**

**PATCH /bulk\_order\_pools/{poolId}/pay/**  
Performs an atomic stock check on each supplier, decrements inventory, sets the bulk order to COMPLETED, and confirms success to the client.

**That’s the full story, step‑by‑step, with every endpoint and its role.**