

Integration

Shelfbot integrates with your existing systems via secure API so products, inventory and orders stay in sync — and bins are presented accurately for picking and putaway.

Key features:

- WMS/ERP agnostic
 - JSON payloads
 - Designed for operational clarity
-

What data flows between systems

Shelfbot needs three things to run smoothly: product master data, inventory, and orders.

Products (SKU master)

Shelfbot imports SKU master data so items can be searched, stored, and picked reliably (SKU, description, barcodes; dimensions/weights preferred).

Inventory

Inventory movements (putaway, adjustments, cycle counts) are reflected between systems so Shelfbot knows what is available to pick.

Orders

Orders and line items flow from your WMS/ERP into Shelfbot, which then requests bins and confirms pick completion back to your system.

Minimum integration requirements

Unique identifiers

- Unique SKU (including variants)
- Unique order ID
- Unique order line ID

API capability

- A documented API with secure access (cloud-based or accessible from your network)
- Ability to create/update/delete products (or at minimum create/update)
- Ability to create/update inventory transactions
- Ability to create/update orders and receive order status updates (or accept status polling)

Operational discipline

- Operators scan every pick and putaway (to prevent stock drift)
- Stock on hand in the external system is accurate before enabling live picking

Practical note: Inventory accuracy depends on scanning discipline. If items are removed or added without scanning, inventory will drift between systems.

Integration approaches

If you haven't built an API yet, start with polling and move to webhooks/events as you mature.

1) Simple polling (fastest start)

Shelfbot periodically pulls orders and product/inventory updates from your system. Good when webhooks aren't available yet.

2) Webhooks / event-driven (recommended)

Your system pushes changes to Shelfbot as they happen (new order, inventory update, SKU update). Lower latency and less API load.

3) Middleware / iPaaS

Use an integration layer to map fields, transform payloads, and handle retries/monitoring. Helpful for multi-system environments.

Shelfbot API

The Shelfbot API documentation is shared with clients through a secure link during the integration process.

Integration process

Discovery

- Confirm your systems (WMS/ERP/e-comm) and integration constraints
- Agree on the "source of truth" for product, inventory, and orders
- Identify barcode standards and any edge cases (bundles, kitting, substitutions)

Mapping

- Map your SKU, order and inventory fields to Shelfbot's model
- Define status events and idempotency keys
- Define error handling (retries, dead-letter queue, human review)

Sandbox & testing

- Test with sample orders and known inventory
- Validate pick confirmation, partial picks, and putaway flows
- Run a short UAT with operators using real scanners and stations

Go-live

- Cutover plan and monitoring
- Operator training (pick, putaway, search, exceptions)
- Post-go-live tuning (slotting, batching, FIFO strategy as needed)

Start fast with a data dive

Before you write a full integration, we can size the system using SKU master data + outbound orders. That gives you confidence on bin count, aisle count, and robot/station sizing.

Quick checklist

- Do you have unique SKUs and barcodes for pick confirmation?
- Can your system export 30+ days of order history (ideally 12 months) for sizing?
- Do you have a stable "source of truth" for inventory on hand?
- Are there any DG / restricted items you plan to store (we'll confirm what's permitted)?
- Do you need one-way (orders → Shelfbot) or two-way (status + inventory sync) integration?

[Read the data dive specification](#)

Ready to map your system?

We'll keep it practical: agree on the source of truth, map the fields, test with real scanners, then go live.

[Talk to an engineer](#) • [Product overview](#) • [Safety & compliance](#)