

Integration Plan: 1C:Trade Management with Jushuitan API and OpenShop

This document describes the integration architecture between the company's internal systems:

- **1C:Trade Management** (1C UT)— an ERP system for managing goods, orders, and stock.
- **OpenShop** — online stores platform, an e-commerce system built with Node.js + Vue.

And the external API — **Jushuitan API**

Objective:

To automate data exchange between 1C UT (the core system), Jushuitan API (for ordering, logistics and inventory), and OpenShop (online stores). 1C UT acts as the central hub.

1. Data for Synchronization

Reference Data (Master Data)

Object	Sync Direction	Key Fields
Products (Catalog)	Jushuitan → 1C UT → OpenShop	Code, Name, Article Number, Barcode, Unit of Measure, Prices (retail/purchase), Weight, Dimensions, Active Status
Customers (Clients)	OpenShop → 1C UT → Jushuitan	Full Name, Phone, Email, Delivery Address, Type (Individual/Legal), Discount Group
Warehouses	1C UT ↔ Jushuitan	Warehouse Code, Name, Address, Primary/Virtual Flag
Order Statuses	OpenShop ↔ 1C UT ↔ Jushuitan	Status Code, Name (e.g., "Accepted", "Packed", "Shipped")

Documents (Operational Data)

Object	Sync Direction	Key Fields
Orders	OpenShop → 1C UT → Jushuitan	Order Number, Date, Customer, Products (code, quantity), Amount, Status, Delivery/Payment Method
Stock Balances	1C UT ↔ Jushuitan →	Warehouse, Product, Quantity, Reserved Amount

Object	Sync Direction	Key Fields
	OpenShop	
Sales/Shipments	Jushuitan → 1C UT	Invoice Number, Shipment Date, Products, Source Warehouse
Payments	OpenShop → 1C UT → Jushuitan	Amount, Date, Payment Method, Order Reference

2. Business Processes

Main Data Flows

1. Product Upload to OpenShop:

- Jushuitan → 1C UT → OpenShop: Update product catalog (description and images), prices, stock.
- **Frequency:** Real-time or every 15–30 minutes.

2. Order Reception from OpenShop:

- OpenShop → 1C UT: New orders are automatically sent to 1C UT for item reservation. Payment verification from customer: 1C UT → Bank. Delivery address is determined using YandexMaps.
- **Action in 1C:** Create “Customer Order” and “Order Payment” documents

3. Order Transmission to Jushuitan:

- 1C UT → Jushuitan: Orders are sent via Jushuitan API for logistics (packing, delivery).
- **Jushuitan APIs used:** orders.upload (order upload), orders.status.update (status updates)

4. Stock Synchronization:

- Jushuitan → 1C UT: Updated stock after shipment/receipt.
- 1C UT → OpenShop: Update stock in the online store.
- **Frequency:** Hourly or on change.

5. Order Status Updates:

- Jushuitan → 1C UT → OpenShop: Order statuses (“In Transit”, “Delivered”) are synchronized with OpenShop to inform customers.
-

3. Technical Implementation

Integration Components

- **For 1C UT:** Use **HTTP connections** (external handlers) or **REST adapters** to work with Jushuitan API.
- **For OpenShop:** Use **standard 1C integration module** (CommerceML) for synchronizing product data and orders.
- **For Jushuitan:** Key methods:
 - stock.query (stock balances),
 - orders.upload (order upload),
 - orders.status.update (status updates)

Data Exchange Diagram



4. Data Requirements

- **Unique Identifiers:** Product, order, and customer identifiers must match across all systems (e.g., product article in 1C = SKU in OpenShop).
- **Error Handling:**
 - Data validation before sending (e.g., required fields check).
 - Error logging in a dedicated journal in 1C.
- **Security:**
 - Use HTTPS and Jushuitan authorization tokens.
 - Restrict API access rights.

5. Implementation Stages

1. **Setup 1C ↔ OpenShop exchange** (via CommerceML)
2. **Develop 1C ↔ Jushuitan connector:**
 - Implement order export from 1C to Jushuitan
 - Import stock balances from Jushuitan to 1C
3. **Testing:**
 - Trial sync of test orders/stock
 - Verify status accuracy
4. **Go Live:** Gradual rollout of processes
5. **Monitoring:** Daily review of exchange logs

