# PAMS B2B Push PO API

Partner Integration Guide

2026-06-09

## Contents



## PAMS B2B Push PO API — Partner Integration Guide

**Version:** 1.0 **Last updated:** 2026-06-08 **Base URL:** `https://b2b.pamsauto.com`

This document describes how to push purchase orders into PAMS programmatically. It is the contract between PAMS and partner systems (CCC, PartsTrader, APU-style, or any other system that already knows what part it wants to buy from us). If you are integrating against PAMS for the first time, start here.

---

### 1. Getting an API key

API keys are issued by PAMS. Request a key by emailing **pamsadmin@pamsauto.com** with:

- Your company name
- The PAMS customer account number you'll be billing against
- Whether you want a **Sandbox** key, a **Production** key, or both

PAMS will return the key(s) to you in a one-time secure message. **Save the key immediately** — PAMS does not store the raw key, only a hash, and cannot retrieve it for you later. If you lose a key, request a new one (the old one will be revoked).

A customer account may have at most one active key per environment at a time. Generating a new key automatically invalidates the previous active key for that environment.

### Sandbox vs Production

- **Sandbox keys** look like `pams_test_<64 hex chars>`. They route to the PAMS test database (`pams-sql3`), which is restored from Friday's production backup every Saturday night.
- **Production keys** look like `pams_live_<64 hex chars>`. They route to the live PAMS database.
- The **URL is the same** for both. The environment is determined entirely by which key you send. This makes it impossible to accidentally hit production with a sandbox key (or vice versa) by tweaking a URL.

## 2. Authentication

Every request must include the key in the `Authorization` header using the standard `Bearer` scheme:

```
Authorization: Bearer pams_live_a1b2c3d4e5f6...
```

A missing, malformed, or revoked key returns **HTTP 401** with `{"error":"UNAUTHORIZED", ...}`.

---

## 3. Endpoints

| Method | URL | Purpose |
|---|---|---|
| POST | `/B2BPushPO.ashx` | Push a new purchase order |
| POST | `/B2BPushPOLabels.ashx?workOrder=<WorkOrderNumber>` | Attach shipping labels to an existing WO |
| GET | `/B2BPushPOStatus.ashx?workOrder=<WorkOrderNumber>` | Poll a WO's current status |

All requests and responses are `application/json`. Errors are also JSON — never HTML or plain text.

---

## 4. POST `/B2BPushPO.ashx` — push a PO

Creates a new work order in PAMS from your purchase order. Everything happens in a single round-trip: pricing is validated, the WO is created, and any labels you send inline are attached.

### Request body

```
{
  "poNumber": "YOUR-PO-123",
  "shipTo": {
    "name": "Joe Shop",
    "street1": "123 Main St",
    "street2": null,
    "city": "Somewhere",
    "state": "OH",
    "zip": "44256",
    "phone": "555-555-5555"
  },
  "lineItems": [
    {
      "inventoryId": 5482552,
      "stockTicketNumber": "N1234A",
      "unitPrice": 575.00,
      "shippingAmount": 47.50
    }
  ],
```

```
  "shippingLabels": [
    {
      "inventoryId": 5482552,
      "trackingNumber": "1Z999AA10123456784",
      "carrier": "UPS",
      "mimeType": "application/pdf",
      "labelData": "<base64-encoded label bytes>"
    }
  ]
}
```

### Field reference

Split into four small tables so long field names don't overflow.

#### Top-level fields

| Field | Type | Required | Notes |
|---|---|---|---|
| `poNumber` | string | yes | Your PO number. Free-form. PAMS accepts duplicate PO numbers across requests — each push creates a new WO. |
| `shipTo` | object | yes | Ship-to address. Fields below. |
| `lineItems` | array | yes | At least one line. Each line is exactly one unit — to order two of a part, send two line items. |
| `shippingLabels` | array | no | Inline shipping labels (see §7). You can also push labels later via `/B2BPushPOLabels.ashx`. |

#### `shipTo` fields

| Field | Type | Required | Notes |
|---|---|---|---|
| `name` | string | recommended | Recipient's name or company. Appears on the work order's ship-to card. |
| `street1` | string | yes | Street is required. |
| `street2` | string | no | Apt / suite / etc. |

| Field | Type | Required | Notes |
|---|---|---|---|
| `city` | string | recommended | If you omit it we derive from the ZIP. |
| `state` | string | optional | Two-letter state code. If omitted, derived from the ZIP. If sent, we trust you and skip the lookup. |
| `zip` | string | yes | 5-digit ZIP. PAMS does not validate or geocode the street, but the ZIP is used to derive state and county. |
| `phone` | string | recommended | Recipient phone. |

**`lineItems[]` fields (each line is one part)**

| Field | Type | Required | Notes |
|---|---|---|---|
| `inventoryId` | integer | yes | PAMS InventoryID. Identifies the specific physical part. |
| `stockTicketNumber` | string | yes | Must match the inventory row — catches mis-identified parts before they ship. |
| `unitPrice` | decimal | yes | All-in price you are paying for the part. Includes any core charge. See §6 for the math. |
| `shippingAmount` | decimal | depends | **Required (and $> 0$)** if your account pays shipping. **Forbidden** if your account provides labels. See §5. |

**`shippingLabels[]` fields (each label is one shipment)**

| Field | Type | Required | Notes |
|---|---|---|---|
| `inventoryId` | integer | yes | The `lineItems[].inventoryId` this label covers. Each label is assigned to one specific part — required so PAMS can carry the assignment through to the eventual invoice. Must match an `inventoryId` on this PO. |
| `trackingNumber` | string | no | Carrier tracking number. Extracted from the label image if omitted. |
| `carrier` | string | no | Carrier code (`UPS`, `FedEx`, etc.). Extracted from the label if omitted. |
| `mimeType` | string | yes | One of: `application/pdf`, `image/png`, `image/jpeg`, `image/jpg`, `image/bmp`. |
| `labelData` | string | yes | Base64-encoded label bytes. |

### Success response (HTTP 200)

```
{
  "workOrderNumber": 1234567,
  "poNumber": "YOUR-PO-123",
  "labelsAccepted": 1,
  "totalPrice": 575.00,
  "shippingCharge": 47.50
}
```

`workOrderNumber` is the user-facing PAMS work order number. Use this to attach labels later or poll status.

### Failure response (HTTP 4xx)

PAMS uses **one-fail-all-fail**: if any line item or shipping check fails, the entire PO is rejected and no work order is created. The response details every issue found in a single call so you can fix them all and retry in one round-trip.

```
{
  "error": "PO_VALIDATION_FAILED",
  "message": "PO failed validation. See details for the specific issues - fix all of them and resubmit.",
```

```
  "details": [
    {
      "lineItemIndex": 0,
      "inventoryId": 5482552,
      "stockTicketNumber": "N1234A",
      "issue": "UNIT_PRICE_TOO_LOW",
      "submittedValue": 500.00,
      "expectedMinimumValue": 575.00
    }
  ]
}
```

The error and issue codes are stable. See §9 for the full catalog.

---

## 5. Shipping modes

Every PAMS B2B customer account is configured as one of:

- **Customer provides labels.** You always ship your own labels. None of your line items may carry a `shippingAmount`. The work order is created with no shipping charge on any line.
- **Customer pays for shipping.** Every line item carries its own `shippingAmount` (`> 0`). Per-line because a multi-line PO can ship as multiple shipments at different rates — allocate however you like. PAMS independently quotes the **whole PO as one freight bundle** (the rate engine quotes a bundle, not per part) and validates: **the sum of your line `shippingAmount` values must equal or exceed PAMS's quote.** Lower is a failure (one error against the whole PO). Higher is accepted — each line is billed the amount you sent.

The mode is set on your account during onboarding. If you need to switch, contact PAMS.

---

## 6. Pricing rules

Your `unitPrice` on every line must be **greater than or equal to** PAMS's expected all-in price for the part. Lower is a failure. Higher is accepted — you'll be billed what you sent.

The expected all-in price is computed from PAMS's inventory data and equals:

```
expectedAllInPrice = retailPrice
                   + applicableCoreCharge
                   + hazmatSurcharge
```

Where: - `retailPrice` is the value PAMS published on the daily GoParts feed (`rprice` column) — no floor or markup. - `applicableCoreCharge` is the part's core charge, but only if it exceeds a threshold: **$50** for non-LTL parts, **$200** for LTL parts. Cores at or below the threshold are absorbed by PAMS and contribute zero. **If you send `coreCharge` on the line item, your value is used instead — see the field reference in §4.** - `hazmatSurcharge` is **$75** for hazmat part types: **210 (seat belts)** and **253 (airbags)**. Zero for all other part types. This is a shipping surcharge for hazmat materials; it's billed as part of the sales price, not as separate shipping.

### Worked example

PAMS holds an airbag with retail price $55 and no core. It is non-LTL, part type 253.

- `retailPrice = 55`
- Core = $0 → `applicableCoreCharge = 0`
- Part type 253 (hazmat) → `hazmatSurcharge = 75`
- `expectedAllInPrice = 55 + 0 + 75 = 130`

| Your `unitPrice` | Result |
|---|---|
| 100 | `UNIT_PRICE_TOO_LOW` — PO rejected |
| 130 | Accepted (exact match) |
| 175 | Accepted (you overpaid by $45) |

---

## 7. Shipping labels

PAMS accepts these formats for shipping labels:

- `application/pdf`
- `image/png`
- `image/jpeg`
- `image/jpg`
- `image/bmp`

Send the bytes base64-encoded in `labelData`. Carrier and tracking number are optional — if you don't send them, PAMS attempts to extract them from the label image itself. `inventoryId` is **required** on every label — each label is assigned to one specific part on the order. (Multi-part shipments on a single label aren't currently supported; if you need that, contact PAMS.)

You can attach labels either:

1. **Inline on the PO push** — include `shippingLabels` on the `/B2BPushPO.ashx` request body.
2. **Later, after the WO is created** — POST to `/B2BPushPOLabels.ashx?workOrder=<WorkOrderNumber>` with body `{"labels":[ ... same shape ... ]}`. Useful when labels aren't ready at PO submission time. The `inventoryId` on each label must point at a part that is still on that work order.

A work order may sit indefinitely waiting for labels — PAMS will not ship until labels are present.

The per-part assignment carries through to the invoice when the WO is promoted, **provided the inventory still aligns** — same parts, no inventory swaps, no dropped lines. If the work order changes between label upload and invoice promotion (PAMS staff substitutes a part, etc.), the label is preserved on the invoice but left unassigned for PAMS staff to attach manually.

---

## 8. POST `/B2BPushPOLabels.ashx?workOrder=<n>` — add labels later

### Request body

```
{
  "labels": [
    {
      "inventoryId": 5482552,
      "trackingNumber": "1Z999AA10123456784",
      "carrier": "UPS",
      "mimeType": "application/pdf",
      "labelData": "<base64>"
    }
  ]
}
```

`inventoryId` is required — must match a part on the work order you're attaching to.

### Success (HTTP 200)

```
{ "workOrderNumber": 1234567, "labelsAccepted": 1 }
```

You may only attach labels to work orders owned by your account. Attempting to attach to a WO you don't own returns 404.

---

## 9. GET `/B2BPushPOStatus.ashx?workOrder=<n>` — poll status

Returns a minimal status snapshot. Poll this on a schedule that suits your system (no fixed PAMS rate limit, but be reasonable — every 5–15 minutes is plenty).

**Success (HTTP 200)**

```
{
  "workOrderNumber": 1234567,
  "createdDate": "2026-06-08T14:23:01",
  "lineItemCount": 1,
  "labelCount": 1
}
```

You may only query work orders owned by your account.

---

## 10. Cancellation

**The API does not support PO cancellation.** Once a PO is accepted and a work order is created, the work order can only be cancelled by contacting PAMS directly (phone or email). This is intentional — to prevent partial-fulfillment race conditions.

---

## 11. Error and issue code reference

**Top-level error codes (`error` field)**

| Code | HTTP | Meaning |
|---|---|---|
| `UNAUTHORIZED` | 401 | API key missing, malformed, or inactive. |
| `BAD_REQUEST` | 400 | Request structure is invalid (missing required fields, malformed JSON, etc.). |
| `PO_VALIDATION_FAILED` | 400 | Business validation failed. See `details` for the per-line issues. |
| `CUSTOMER_NOT_FOUND` | 400 | Your account no longer exists in PAMS. Contact PAMS. |
| `SHIPPING_QUOTE_FAILED` | 400 | PAMS could not compute a shipping quote for this PO. |
| `WORK_ORDER_CREATE_FAILED` | 500 | Validation passed but PAMS could not persist the work order. Contact PAMS with the response. |
| `INTERNAL_ERROR` | 500 | An unexpected error. PAMS support has been notified automatically. |

**Per-line issue codes (`details[].issue`)**

| Code | Meaning |
|---|---|
| `INVENTORY_NOT_FOUND` | The `inventoryId` doesn't exist in PAMS. |
| `STOCK_TICKET_MISMATCH` | The `stockTicketNumber` doesn't match the inventory row. |
| `PART_NOT_AVAILABLE` | The part is no longer in stock or is blocked from online sale. |
| `UNIT_PRICE_TOO_LOW` | Your `unitPrice` is below PAMS's expected all-in price. See `expectedMinimumValue`. |
| `SHIPPING_AMOUNT_TOO_LOW` | The sum of your per-line `shippingAmount` values is below PAMS's freight quote for the PO. |

| Code | Meaning |
|---|---|
| `SHIPPING_AMOUNT_REQUIRED` | Your account pays for shipping; this line is missing `shippingAmount` or sent it as 0. |
| `SHIPPING_AMOUNT_NOT_ALLOWED` | Your account provides its own labels; this line included a `shippingAmount`. |
| `LABEL_FORMAT_INVALID` | One of the labels has an unsupported `mimeType`. See §7. |
| `MISSING_FIELD` | A required field is missing on a line item or label. |

---

## 12. Status callbacks (webhooks)

PAMS does not currently push status callbacks to partners. Use the polling endpoint in §9.

---

## 13. Support

Questions, issues, or API key requests: **pamsadmin@pamsauto.com**

When reporting an issue, include the full request body and the response body. PAMS keeps a server-side log of every request and can look up the exact failure given the timestamp and your customer account.