

Wolf & Badger — Order Ready to Ship Webhook

Overview

Wolf & Badger sends an outbound webhook when an order is accepted and all shipping documents are available. This allows brands and their fulfilment or warehouse management systems to receive order data and shipping documents automatically, without polling the W&B platform.

W&B defines the payload structure. Your system must expose an HTTPS endpoint that accepts POST requests matching this spec.

Trigger

The `order.ready_to_ship` event fires when an order is accepted and all shipping documents have been generated. Pending events are pushed on a **15-minute interval** and pushed repeatedly for 3 days from time of creation or until delivered successfully (i.e. 200 Response).

Endpoint Requirements

You provide W&B with the HTTPS endpoint URL you want order data delivered to.

Requirements:

- Must be HTTPS
- Must accept POST requests with Content-Type: application/json
- Must return 200 OK to acknowledge successful receipt

We provide a secret token which you can use to verify the payload signature.

Request Format

Headers

Header	Value
Content-Type	application/json
X-WB-SIGNATURE	BASE64(HMAC(<secret_token>, <json_string>, SHA-256))

The X-WB-SIGNATURE header contains a Base64-encoded HMAC-SHA256 signature of the raw request body, signed with your secret token. You should verify this signature on every incoming request — see Verifying the Signature below.

The request body is a **compact JSON string** using only `,` and `:` as separators (no extra whitespace).

Body

```
{
  "id": "event_550e8400-e29b-41d4-a716-446655440000",
  "webhook_version": "1.0",
  "event_type": "order.ready_to_ship",
  "created": "2026-05-21T14:30:00Z",
  "data": {
    "wb_order_number": 100,
    "shopify_order_id": "shopify_order_id",
    "shipping_label": {
      "filename": "shipping_label.pdf",
      "content": "<base64-encoded bytes>"
    },
    "packing_slip": {
      "filename": "packing_slip.pdf",
      "content": "<base64-encoded bytes>"
    },
    "waybill": {
      "filename": "waybill.pdf",
      "content": "<base64-encoded bytes>"
    },
    "gift_message": "Happy Birthday!",
    "courier_service": "dhl",
    "shipment_items": [
      { "sku": "ABC-123", "quantity": 1 }
    ]
  }
}
```

Field Reference

Envelope fields

Field	Type	Description
id	string	Unique event identifier in the format event_<uuid>
webhook_version	string	Webhook spec version
event_type	string	Always order.ready_to_ship for this event
created	string	UTC timestamp of when the event was created, in ISO 8601 format

`data` object

Field	Type	Required	Description
wb_order_number	integer	Yes	W&B's unique order identifier
shopify_order_id	string or null	No	Shopify order ID, if the brand is Shopify-integrated

Field	Type	Required	Description
shipping_label	object	Yes	Shipping label document — see Document object
packing_slip	object	Yes	Packing slip document — see Document object
waybill	object or null	No	Waybill document — see Document object. Present for some courier services only
gift_message	string or null	No	Customer gift message, if included with the order
courier_service	string or null	No	Courier to be used for shipment (e.g. dhl, ups uk)
shipment_items	array	Yes	List of SKUs and quantities in this shipment — see Shipment item object

Document object

Applies to shipping_label, packing_slip, and waybill.

Field	Type	Description
filename	string	File name including extension (e.g. shipping_label.pdf)
content	string	File content encoded as Base64 (UTF-8 characters). Decode to retrieve the raw bytes

Decoding example (Python):

```
import base64

raw_bytes = base64.b64decode(payload["data"]["shipping_label"]["content"])
with open("shipping_label.pdf", "wb") as f:
    f.write(raw_bytes)
```

Shipment item object

Field	Type	Description
sku	string	Product SKU as listed in W&B
quantity	integer	Number of units in this shipment

Response Codes

Status	Meaning
200	Event successfully received
400 or above	Event delivery failed

Your endpoint must return 200 OK for every successfully processed event. Any response with a status code of 400 or above is treated as a failed delivery.

Retry Behaviour

- Failed deliveries are retried on the same 15-minute push interval

Verifying the Signature

Every request includes an X-WB-SIGNATURE header. This is a Base64-encoded HMAC-SHA256 signature computed over the raw request body, using your secret token.

Verify this on every incoming request to confirm it originated from W&B and the payload has not been tampered with.

Verification example (Python):

```
import hmac
import hashlib
import base64

def verify_signature(secret_token: str, raw_body: bytes, signature_header: str) -> bool:
    expected = base64.b64encode(
        hmac.new(secret_token.encode(), raw_body, hashlib.sha256).digest()
    ).decode()
    return hmac.compare_digest(expected, signature_header)
```

Verification example (Node.js):

```
const crypto = require('crypto');

function verifySignature(secretToken, rawBody, signatureHeader) {
    const expected = crypto
        .createHmac('sha256', secretToken)
        .update(rawBody)
        .digest('base64');
    return crypto.timingSafeEqual(Buffer.from(expected), Buffer.from(signatureHeader));
}
```

Use a constant-time comparison (`hmac.compare_digest` in Python, `timingSafeEqual` in Node.js) to prevent timing attacks.

Webhook Registration

To register your endpoint, share your HTTPS URL with the W&B team. We will configure the webhook and send your secret token via email. Dashboard self-service registration is coming in a future release.

Notes

- The request body is a compact JSON string with only , and : as separators. Use the **raw body bytes** (before any parsing) when computing or verifying the HMAC signature.
- All Base64-encoded file content is transmitted as UTF-8 characters.
- waybill may be null — your implementation should handle this gracefully.
- shopify_order_id will be null for brands not connected to Shopify.
- The created field refers to when the event was created, not when it was delivered.
- The 15min event push interval is simple enough for the pilot phase; a backoff interval will be included in a later iteration.