Webhook Based Parcel Tracking Guide *(v1.4)*

Introduction

Partners need timely insight into events related to their parcels, and relying solely on time-based requests for updates is not an effective approach. Our Webhook functionality is designed to provide near real-time delivery of parcel state changes directly into partners' internal systems and applications. This supports partners who prefer not to depend on the https://boxnow.bg/en tracking page or who want to present tracking updates through their own mediums and channels.

Through the Webhook, partners receive notifications for all parcel status changes and events. Any events that are not relevant to a partner's workflow can be filtered and ignored on their side, as selective subscription to specific event types is not currently supported.

The available types of webhook events are:

- **New - Parcel has been registered to the BOX NOW system.**
- **Delivered - Parcel has been delivered to end-customer.**
- **Expired-return – Parcel has expired and it was returned to the sender.**
- **Returned - Parcel sent by end-customer, has been returned to the sender.**
- **In-transit – Parcel is being transferred to a specific APM.**
- **in-depot - Parcel is at a BOX NOW Depot.**
- **In-Final-Destination - Parcel has reached its final destination, it's in APM, waiting for pickup.**
- **Cancelled - Parcel has been canceled by the sender and was never given to BOX NOW.**
- **Cancelled-return - Parcel order had been cancelled by the sender and returned to him.**
- **wait-for-load – Parcel is waiting in a specific APM to be taken over by BOX NOW courier for either to be returned to the sender or transferred to another APM.**

Hook Management

During the initial phase, BOX NOW will assist partners in configuring their webhook to the appropriate URL. Each partner receives a secret key to securely validate messages received from BOX NOW through data signing if needed.

Please ensure that webhook destination URLs are publicly accessible and explicitly use SSL Certificates from a public Certificate Authority (CA). If additional authentication is required, this can be managed through request headers.

Each partner can register only one webhook URL. If multiple internal systems need to receive webhook data, we recommend implementing a secure internal mechanism to relay the information after it has been received from BOX NOW.

Webhook Description

Although “**parcelState”** and “**event”** JSON properties appear similar, please rely on the “**event”** JSON property when parsing the status. This is required to ensure proper alignment with what end customers see on the tracking page.

The Webhook request schema conforms to the **CloudEvents** specification. This means that in a request you can find:

- **specversion** - The CloudEvents specification version (e.g., "1.0").
- **id** - A unique identifier for the event.
- **source** - A URI-reference indicating the event producer.
- **type** - A string describing the type of event.
- **time** - The time of webhook dispatch.
- **data.time** - Time when the parcel event was created.
- **datasignature** - HMAC SHA256 digests of the request data that enables the recipient to verify integrity of received data. A partner must first obtain a secret key from us for decryption and verification.

In case multiple similar webhook requests are received in a short period, the client must use “**data.time**” to identify and filter out any invalid or duplicate updates.

Webhook Delivery

System makes N attempts to deliver a webhook request to your specified URL until the recipient responds with a successful message 200 OK.

Our **Retry Policy** uses exponential backoff with variable delays.

These variable delays are as follows:

1. After 5 Minutes
2. After 10 Minutes
3. After 1 Hour
4. After 2 Hours

Last attempt to deliver the hook is made 24 hours after event creation.

Webhook events are generated at specific points in the application through dedicated code snippets that automatically publish messages to Pub/Sub message queues. This mechanism allows the system to emit structured event notifications whenever key operations occur, enabling partners systems to consume and process those events asynchronously.

Problem Mitigation

If we notice repeated unsuccessful responses to webhook requests, our team will reach out to inform you and help identify the issue. If the problem cannot be resolved promptly, we may temporarily disable the Webhook to give you space to address the situation.

All unsuccessful webhook requests are placed in a queue and delivered in order. This ensures that when a retry occurs, you receive the earlier pending requests first, following the process outlined above.

Webhook Request Example:

```
POST {uri} HTTP/1.1
Content-Type: application/json

{
"specversion" : "1.0",
"type" : "bg.boxnow.parcel_event_change",
"source" : "https://boxnow.bg/api/v1/webhooks/{id}",
"subject" : "{ Parcel ID }",
"id" : "{ Message ID }",
"time" : "2022-09-16T11:06:04.296Z",
"datacontenttype" : "application/json",
"datasignature": "{ Box Now server event data integrity signature }"
"data" : {
"parcelId" : "{ Parcel ID }",
"parcelState" : "{ Parcel State }",
"parcelReferenceNumber" : "{ Parcel Reference Number }",
"parcelName" : "{ Parcel Name }",
"orderNumber" : "{ Order Number }",
"event" : "{ Parcel Event }",
"eventLocation" : {
"displayName" : "{ Parcel location in a time of event }",
"postalCode" : "{ Parcel location in a time of event }"
},
"customer" : {
"name" : "{ Customer Name }",
"email" : "{ Customer Email }",
"phoneNumber" : "{ Customer Phone Number }",
},
"additionalInformation" : "{ Delivery Request additionalInformation }",
"time": "2022-09-16T11:06:04.458Z"
}
}
```

Some webhook requests may include additional attributes depending on the triggering event. For instance, the **eventLocation** JSON object is not present in “**new**” or “**delivered**” events.
We recommend adjusting your parsers to avoid expecting every attribute in every

request. After performing the necessary security checks, make your implementation more flexible in handling optional fields to ensure smooth processing.

Testing Webhooks

As with **Partner API** integrations, Webhook integrations should be initially tested in the Staging environment. Each environment will have its own Webhook Secret key (explained above), so please ensure you use the correct key when testing if you check the requests integrity.

BOX NOW is also available to support you in exploring full workflows and testing all relevant variables and events. This helps ensure everything functions as expected before moving to production.

Cross-referencing Webhook Events with the Tracking Page

For your convenience, here is a table that depicts how a customer would see the status of a parcel in the official BOX NOW Tracking page in comparison to the Webhook requests "**event"** types.

Webhook event	Status (EN)
New	Parcel has been registered to the BOX NOW system.
Delivered	Parcel has been delivered to end-customer.
Expired-return	Parcel has expired and it was returned to the sender.
Returned	Parcel sent by end-customer, has been returned to the sender.
In-transit	Parcel is being transferred to a specific APM.
In-depot	Parcel is at a BOX NOW Depot.
In-Final-Destination	Parcel has reached its final destination, it's in APM, waiting for pickup.
Cancelled	Parcel has been canceled by the sender and was never given to BOX NOW.
Cancelled-return	Parcel order had been cancelled by the sender and returned to him.
Wait-for-load	Parcel is waiting in a specific APM to be taken over by BOX NOW courier for either to be returned to the sender or transferred to another APM.