Parcel events webhooks

Revisions

Version	Date	Author	Changes
1.0	21.10.2022	BoxNow BE team	Initial version
1.1	18.11.2022	BoxNow BE team	Extend hook data with reference numbers
1.2	23.11.2022	BoxNow BE team	Extend hook data with additinoalInformation and parcel name
1.3	24.2.2022	BoxNow BE team	Add customer information into webhook data

Introduction

Partners want to be aware of events related to their parcels. The use case could be mainly for near-life integration of state of parcel delivery into partners' customer portals. So this functionality extension focuses mainly on the partners that don't want to use the general https://boxnow.gr tracking page.

Partners can subscribe to the events on parcel. They will get every change about the state happening on the parcel. Events that are not relevant for the partner usage has to be filtered on the partner side. Therefore settings are either on or off for all and partner can't subscribe to just a set of events e.g. `new` and `delivered`.

Availability events in the moments are:

- new - Parcel has been registered to the system
- delivered - Parcel has been delivered
- expired - Parcel expired and will be returned to the sender
- returned - Parcel has been returned to the sender

- in-depot - Parcel is in one of our warehouses
- final-destination - Parcel has reached its final destination, waiting for pickup
- canceled - Parcel order had been canceled by the sender
- accepted-for-return - Parcel has been accepted from customer and will be returned to the sender
- missing - Box Now pickup courier was unable to obtain the parcel for delivery
- accepted-to-locker - Parcel has been accepted from customer and will be sent to the recipient

Hook management

V1: BoxNow management via CMS

In the first phase BoxNow will help Partners to set-up the webhook to the relevant URL. Partners will receive the integrity verification key, to safely recognize messages from BoxNow.

URL where the webhooks are pushed is either publicly accessible, or authentication is done via URL parameters.

V2: CRUD endpoints for partners to manage webhooks

In the latter stage, BoxNow will most likely provide a new API call to Partner API, where partners will be able to maintain their webhooks on their own.

- GET /api/v1/webhooks to list current webhooks
- POST /api/v1/webhooks to create a new webhook
- PUT /api/v1/webhooks/{id} to update a webhook
- DELETE /api/v1/webhooks/{id} to delete a webhook

Webhook

```
- id; string, required, unique identifier
- url; string, required,
- active; boolean, if false webhook will not fire
- type; string, enum={parcel_event_change}
```

- Partner may have multiple webhooks
- Partner receives data integrity verification key (see below)

Webhook description

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

{
  "specversion" : "1.0",
  "type" : "gr.boxnow.parcel_event_change",
  "source" : "https://boxnow.gr/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"
  }
}
```

Schema conforms CloudEvents spec

- datasignature - HMAC SHA256 digest of data for receiver to verify integrity of received data; receiver must first obtain the key from us for verification
- time - time of webhook dispatch
- data.time - time of when the parcel event created
- In case of multiple close updates, the client must use `data.time` to discard invalid updates

Webhook delivery

System makes N attempts to deliver a webhook until the receiver responds 200 OK.

Attempt retry policy uses exponential backoff delay with attempts around every 10 minutes. Last attempt to deliver the hook is made 24 hours after event creation.

Events are triggered programmatically in selected parts of the code via snippets to publish a message to Pub/Sub message queues.