



Intigo API Documentation

Unlock powerful parcel management through our easy-to-use API.

This document guides you through creating, tracking, and managing your parcels efficiently.

- **Authentication:**

An API key is required in the request header under the "Authorization" field with the following format:

Authorization: { apiKey: API_KEY }

Find your API_KEY in your account on my.intigo.tn or dev-my.intigo.tn under the developers section!

- **API Base URL:**

Production: <https://external-api.intigo.tn/secure-api>

Development: <https://dev-external-api.intigo.tn/secure-api>

- **Errors:**

- 400:** Bad Request: Invalid request data
- 401:** Unauthorized: Invalid or missing API key
- 403:** Forbidden: User does not have permission
- 500:** Internal Server Error: Unexpected error occurred

Table of Contents

Page	Description	Method	Route
3.	Create a new parcel	POST	/parcels
5.	Create new parcels	POST	/parcels/batch
6.	Parcels Creation Timeouts Webhook	POST	On your choice
8.	Get a parcel	GET	/parcels/parcel
9.	Get one shipping voucher (bordereau)	GET	/bordereaux/<nid>
10.	Get shipping vouchers (bordereaux)	GET	/bordereaux
11.	Get parcel history	GET	/parcels/<nid>/history
12.	Create a ticket	POST	/tickets
14.	Get a ticket	GET	/tickets/<number>
15.	Intigo Webhook Setup	POST	On your choice

Create a new parcel

- **Method:** POST
- **Route:** /parcels
- **Purpose:** Creates a new parcel in the delivery system.
- **Request Body:**

Field	Description	Type	Format (if applicable)
cid	Client ID provided in the request body	string	5-20 characters
name	Recipient name	string	3-20 characters
phone	Recipient phone number (8 digits)	string	8 digits, numbers only
amount	Parcel price	number	0-3000
city	Recipient city	string	See Additional Notes
subDivision	Recipient subdivision	string	See Additional Notes
address	Recipient address	string	3-100 characters
size	Parcel size (1: Normal, 4: Large, or 5: Extra Large)	number	1, 4, or 5 only
isFragile	True if parcel is fragile	boolean	true or false
isExchange	True if parcel is an exchange	boolean	true or false
description	Describe the contents of the parcel	string	Max 500 characters
additionalInformation	Additional information for delivery	string	Max 500 characters
pickUpAddress	Where the parcel will be picked up from	string	3-500 characters
pickUpCity	The city from which the parcel will be picked up	string	See Additional Notes
pickUpSubDivision	The subdivision or area within the pickup city from which the parcel will be picked up	string	See Additional Notes

- **Additional Notes:**

- All fields are required except **isFragile** and **isExchange**, which have a default value of **false**.
- The API will reject any request containing additional, undeclared fields in the body.
- For city, pickUpCity, subDivision, and pickUpSubDivision fields, choose from the predefined list available at external-api.intigo.tn/secure-api/regions. Exact spelling and capitalization are mandatory.

- **Response:**

On success (status code 201), the response body will be a JSON object containing the following details:

status:	Indicates the parcel status
nid:	Unique identifier for the new parcel
cid:	Client ID provided in the request body
secretCode:	Parcel secret code (used when the scan code is damaged)

Create new parcels

- **Method:** POST
- **Route:** /parcels/batch
- **Purpose:** Creates new parcels in the delivery system.

- **Request Body:**

This endpoint expects an array of objects conforming to the same structure as the single parcel definition used in the `/parcels` request. Please refer to the previous route `/parcels` documentation for detailed information on the required fields and their data types.

- **Response:**

On success (status code 201), the response body will be an array of JSON objects containing the following details:

status:	Indicates the parcel status
nid:	Unique identifier for the new parcel
cid:	Client ID provided in the request body
amount:	Parcel price
city:	Recipient city
subDivision:	Recipient subdivision
isExchange:	True if parcel is an exchange
nbTentative:	The number of delivery attempts made for the parcel

- **Important:**

- During peak periods with high user activity, creating a large number of parcels simultaneously might exceed the allocated processing time, resulting in a request timeout.
- While the timeout occurs, the platform continues processing parcel creation in the background.
- To handle this case, please follow up the [Parcels Creation Timeouts Webhook](#)

Parcels Creation Timeouts Webhook

- **Overview:**

- This document explains how to set up and use the Parcels Creation Timeouts Webhook to receive a notification once your parcels have been created.

- **Requirements:**

- A registered Intigo user account
- A route on your server that can accept POST requests and **must return the exact it receives**

- **Setting Up the Webhook:**

1. Log in to your Intigo account at my-rfa.intigo.tn or dev-my.intigo.tn.
2. Navigate to the "Development settings" section.
3. In the "**Webhook for batch parcel creation**" section:
 - a. Enter your webhook URL in the "Endpoint" field.
 - b. (Optional) You can enter a custom token in the "Token" field. If you do, we'll use it in the `Authorization` header of our webhook requests with the format `Bearer YOUR_TOKEN`. Otherwise, we'll automatically use your API key from the previous section.
4. Click "Test" to send a test notification. The button is disabled until you enter a valid URL.
5. Once the test succeeds, click "Save" to store your URL and token (if used). You'll then receive a notification once your parcels have been created.

- **Notification Process:**

- Once parcel creation is complete, the platform will send a POST request to your configured webhook URL.

- The request body will contain an array of objects, each representing a created parcel:

```
[
  {
    "status": number,
    "nid": string,
    "cid": string,
    "scanCode": string,
    "secretCode": string
  },
  ... (additional parcels)
]
```

- **Object Fields:**
- **status:** A numerical code indicating the parcel status.
- **nid:** The unique identifier assigned to the parcel.
- **cid:** The customer identifier associated with the shipment.
- **secretCode:** Parcel secret code (used when the scan code is damaged).

Cet a parcel

- **Method:** GET
- **Route:** /parcels/parcel
- **Purpose:** This route allows you to retrieve detailed information about a specific parcel using either its unique identifier (nid) or the associated customer ID (cid).
- **Request Query:**

Field	Description	Type	Format (if applicable)
cid	Client ID provided in the request body	string	3-20 characters
nid	The unique identifier of the parcel	string	5-9 characters

- **Additional Notes:**
 - Either **cid** or **nid** is required to identify the specific parcel. Providing both is not recommended and may lead to unexpected behavior.
- **Response:**

On success (status code 201), the response body will be a JSON object containing the following details:

 - status:** Indicates the parcel status
 - nid:** Unique identifier for the new parcel
 - cid:** Client ID provided in the request body
 - secretCode:** Parcel secret code (used when the scan code is damaged)

Get one shipping voucher (bordereau)

- **Method:** GET
- **Route:** /bordereaux/<nid>
- **Purpose:** This route generates and serves a PDF bordereau (identification sheet) for a specific parcel identified by its unique ID (nid).

- **Request Params:**

Field	Description	Type	Format (if applicable)
nid	The unique identifier of the parcel	string	5-9 characters

- **Additional Notes:**

- **nid** is required to identify the specific parcel.
- Providing a valid **nid** in the URL path is mandatory for successfully retrieving the parcel's bordereau

- **Response:**

On success (status code 201), the response body will contain the following details:

- **Content-Type:** application/pdf
- **Body:** The generated PDF file containing bordereau

Get shipping vouchers (bordereaux)

- **Method:** GET
- **Route:** /bordereaux
- **Purpose:** This route allows you to generate and download a single PDF file containing bordereaux for multiple parcels.
- **Request Query:**

Field	Description	Type	Format (if applicable)
nids	Multiple query parameters named nids, each containing a unique parcel identifier (nid).	array	Each 5-9 characters

- **Example:**
 - Download bordereaux for parcels with nids 1097367 and 1097368:
`/bordereaux?nids=1097367&nids=1097368`
- **Additional Notes:**
 - At least two **nids** parameters are required.
- **Response:**

On success (status code 201), the response body will containing the following details:

 - **Content-Type:** application/pdf
 - **Body:** The generated PDF file containing bordereaux

Get parcel history

- **Method:** GET
- **Route:** /parcels/<nid>/history
- **Purpose:** This route allows you to retrieve the history of actions and events for a specific parcel.

- **Request Params:**

Field	Description	Type	Format (if applicable)
nid	The unique identifier of the parcel	string	5-9 characters

- **Additional Notes:**

- **nid** is required to identify the specific parcel.
- Providing a valid **nid** in the URL path is mandatory for successfully retrieving the parcel's history

- **Response:**

On success (status code 201), the response body will be a JSON object containing the following details:

- status:** Indicates the parcel status
- date:** The date of the action
- centrale:** Indicates whether the action occurred at a central location

Create a ticket

- **Method:** POST
- **Route:** /tickets
- **Purpose:** This route enables the creation of a ticket, allowing you to submit inquiries, requests, or issues for further assistance or resolution.

- **Request Body:**

Field	Description	Type	Format (if applicable)
nid	The unique identifier of the parcel	string	5-9 characters
ticketType	See Additional Notes	Number	1, 2, 3, 4 or 5
service	See Additional Notes	Number	1, 2 (optional)
title	Ticket title (required for type 1)	string	Max 50 characters
description	Ticket description (required for type 1)	string	Max 500 characters
newAmount	New requested amount (required for type 2)	Number	0-3000
newPhone	New phone number (required for type 4)	String	8 digits, numeric
newCity	New delivery city (required for type 3)	string	See Additional Notes
newSubDivision	New delivery subdivision (required for type 3)	string	See Additional Notes
newAddress	New delivery address (required for type 3)	string	Max 500 characters

- **Additional Notes:**

- For newCity and newSubDivision fields, choose from the predefined list available at external-api.intigo.tn/secure-api/regions. Exact spelling and capitalization are mandatory.
- **nid** is required to identify the specific parcel.

- **Ticket Types:**

ID	Ticket Type	Required Fields	Description
1	Standard	Title, Description	This type of ticket is used for any general question or issue.
2	Price Change	newAmount	Request a change in the delivery price.
3	Address Change	newAddress, newSubDivision and newCity	Request a change in the delivery address.
4	Phone Number Change	newPhone	Request a change in the phone number associated with the parcel.
8	Relance du colis	nid	This type of ticket is used to relaunch a parcel

- **Response:**

On success (status code 201), the response body will be a JSON object containing the following details:

number: The unique identifier of the ticket

Get a ticket

- **Method:** GET
- **Route:** /tickets/<ticketNumber>
- **Purpose:** This route allows you to retrieve detailed information about a specific ticket using its unique identifier.

- **Request Params:**

Field	Description	Type	Format (if applicable)
ticketNumber	The unique identifier of the ticket	number	5-9 characters

- **Additional Notes:**

- **ticketNumber** is required to identify the specific parcel.
- Providing a valid **ticketNumber** in the URL path is mandatory for successfully retrieving the ticket

- **Response:**

On success (status code 201), the response body will be a JSON object containing the following details:

title: Indicates the ticket title
status: Indicates the ticket status
nid: The unique identifier of the parcel
ticketNumber: The unique identifier of the ticket

- **Ticket Statuses:**

- **1 Pending**
- **2 In progress**
- **3 Treated**
- **4 Transferred to administration**

Intigo Webhook Setup

This document explains how to set up and use the Intigo webhook to receive real-time updates on parcel statuses.

- **Overview:**

- The Intigo webhook sends notifications whenever the status of a parcel changes. This allows you to automate tasks, update systems, and keep users informed about their shipments.

- **Requirements:**

- A registered Intigo user account
- A route on your server that can accept POST requests and **must return the exact it receives**

- **Setting Up the Webhook:**

6. Log in to your Intigo account at my.intigo.tn or dev-my.intigo.tn (for development).
7. Navigate to the "Development settings" section.
8. In the "**Webhooks**" section:
 - a. Enter your webhook URL in the "Endpoint" field.
 - b. (Optional) You can enter a custom token in the "Token" field. If you do, we'll use it in the `Authorization` header of our webhook requests with the format `Bearer YOUR_TOKEN`. Otherwise, we'll automatically use your API key from the previous section.
9. Click "Test" to send a test notification. The button is disabled until you enter a valid URL.
10. Once the test succeeds, click "Save" to store your URL and token (if used). You'll then receive real-time parcel updates.

- **Webhook Data Format:**

The webhook sends a POST request with this JSON data:

```
{
  "status":      number,
  "message":     string,
  "date":        date,
  "parcels":     array: [
    {
      "nid": string,
      "cid": string
    },
    {
      "nid": string,
      "cid": string
    },
    ... (additional parcels)
  ]
}
```

- **status:** A numerical code indicating the update's overall status (see Status Meanings section).
- **message:** A human-readable message providing context (see Status Meanings section).
- **date:** The date when the action was performed, formatted in **ISO 8601 format**.
- **parcels:** An array of objects, each representing an affected parcel.
 - **nid:** The unique parcel identifier.
 - **cid:** The customer identifier associated with the shipment.

- **Expected Response:**

Your endpoint should return a 200 OK response to acknowledge receipt. You can optionally include information in the response body, but Intigo won't process it.

- **Security Considerations:**

Secure your endpoint and only allow authorized access. Never share your API key or token. Intigo uses HTTPS, but further security measures are recommended.

- **Status Meanings:**

The webhook will send updates with the following statuses and messages:

Status	Message	Description
2	Colis assigné au livreur	Parcel assigned to a delivery person
6	Colis livré	Parcel delivered
8	Colis annulé par admin	Parcel canceled by admin
10	Colis en cours de livraison	Parcel out for delivery
15	Colis en retour provisoire	Parcel temporarily returned
16	Colis en retour définitif	Parcel permanently returned
17	Colis en retour vendeur	Parcel returned to seller
20	Colis en transfert vers <central name>	Parcel transferred to central hub
21	Colis perdu	Parcel lost
25	Colis entré au <central name>	Parcel received at central hub
27	Colis en transfert vers <central name>	Parcel in transit
28	Vérification	Parcel require verification
29	Relance	Parcel will be relaunched

By following these guidelines, you can effectively use the Intigo webhook to improve your business workflows and keep users informed about their parcels.

