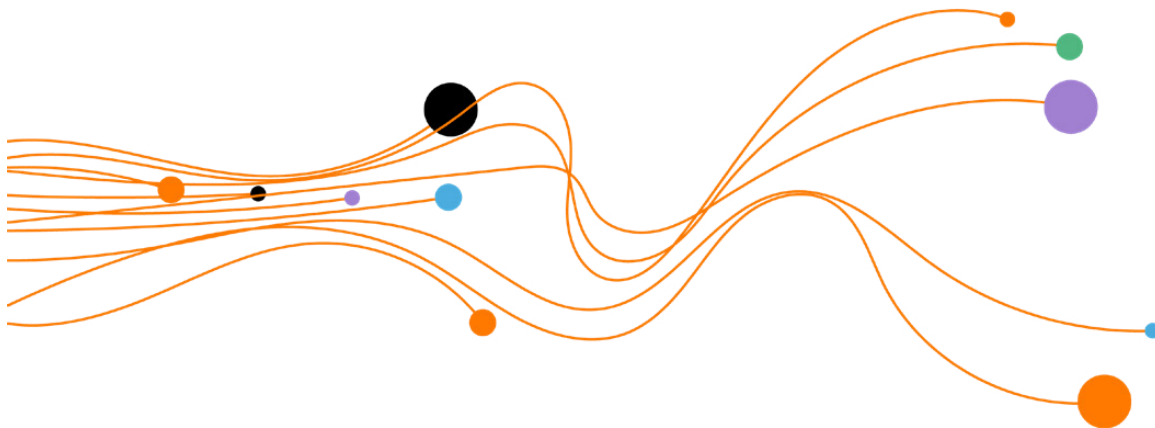


Configure Incident API in Postman

Contents

Prerequisite: before starting	2
Use case 1: Get all my Incidents	3
Use case 2: Filter the Incidents list.....	4
Use case 3: Pagination of the Incidents list.....	5
Use case 4: Get details of an Incident.....	6
Annexes: Configure API OAuth 2.0 authentication in Postman	7
Orange API Specific Terms for Maintenance API	Error! Bookmark not defined.



Prerequisite: before starting

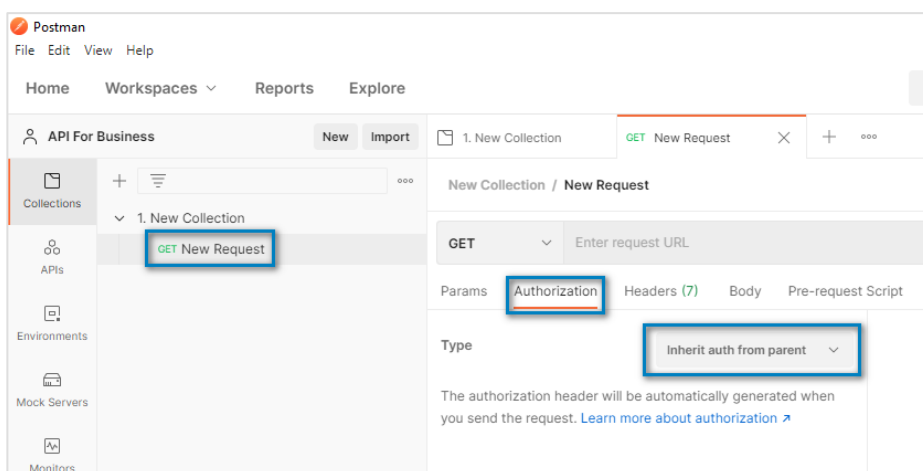
In the “Authorization” tab of the new request, select “Inherit auth from Parent”

Order for Business API is protected with OAuth 2.0.

- The objective is to get the `access_token`.
- The `access_token` must be added to HTTP headers of each API call.
- The Postman authentication configuration is detailed in the [annexes](#) of this document

The API also requires the generation of an **API-Key**

- Your Orange Business Service API contact can provide you this API key.
- The API key must be added to HTTP headers of each API call.
- The header is `x-api-Key`.



Use case 1: Get all my Incidents

List all my incidents without filter, sort, or paging

Use the new request created in the previous step

Request

- Method & URL: **GET** , <https://api.orange.com/incident/b2b/v1/incidents>
- Header : add **x-api-key**

Response

"HTTP Status code = 200" for a successful answer Body

Header you will receive 2 specific headers:

- **X-Result-Count** header, the number of Incidents
- **X-Total-Count** header, the number of all Incidents matching criteria.

The screenshot shows the Orange Developer interface for a REST client. The request is a GET to `https://api.orange.com/incident/b2b/v1/incidents` with an `x-api-key` header. The response body is a JSON array of incident objects.

KEY	VALUE
<input checked="" type="checkbox"/> x-api-key	{{API-KEY}}
Key	Value

```
1  {
2    {
3      "id": "2103M44704",
4      "status": "VALIDATED",
5      "impact": "HIGH",
6      "priority": "P1",
7      "urgency": "HIGH",
8      "serviceCondition": "INTERUPTED",
9      "type": "FAILURE",
10     "origin": "CUSTOMER",
11     "shortDescription": "CLOT_AUTO_ATQxxx positionné",
12     "description": "fggf",
13     "servicePoint": {
14       "serviceLevelManagement": {}
15     },
16     "customer": {
17       "id": "33-01257052",
18       "name": "BULLES'0"
19     },
20   },
21 }
```

Use case 2: Filter the Incidents list

You have the possibility to filter a collection by adding to the URI some attributes.

Example: retrieve incidents occurred in France and with a priority "P1".

Request

Add the following parameters the previous request:

- `location.address.country=FR`
- `priority=P1`

Response

List of your filtered incidents in a json format

The screenshot shows an API client interface for the endpoint `https://api.orange.com/incident/b2b/v1/incidents?location.address.country=FR&priority=P1`. The parameters section is active, showing two checked parameters: `location.address.country` with value `FR` and `priority` with value `P1`. The body section is also active, displaying a JSON response in 'Pretty' format. The response is a single incident object with the following fields:

```
1 {
2   "id": "2103M44704",
3   "status": "VALIDATED",
4   "impact": "HIGH",
5   "priority": "P1",
6   "urgency": "HIGH",
7   "serviceCondition": "INTERUPTED",
8   "type": "FAILURE",
9   "origin": "CUSTOMER",
10  "shortDescription": "CLOT_AUTO_ATQxxx positionné",
11  "description": "fggf",
12  "servicePoint": {
13    "serviceLevelManagement": {}
14  },
15  "customer": {
16    "id": "33-01257052",
17    "name": "BULLES'O"
18  }
19 }
```

Use case 3: Pagination of the Incidents list

The API provides a pagination mechanism with the following query parameters

- **offset**: The index of the first element to retrieve. Zero is the first item of the collection.
- **limit**: The maximum number of items to return.
- **sort**: The comma-separated list of field names to sort the result. Prefixing a field name with a "-" sign will indicate a descending order.

Thanks to these 3 query parameters, you can retrieve Incidents page per page.

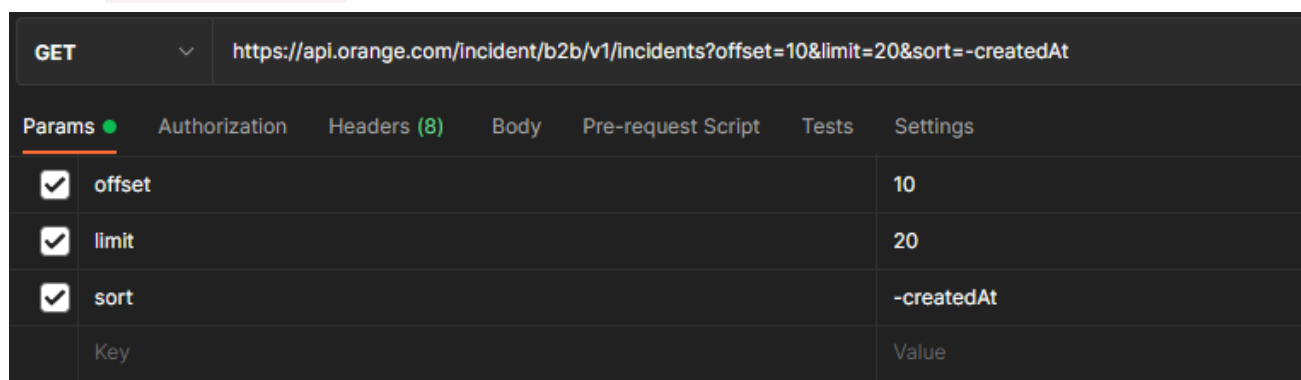
Example: to get the next 20 Incidents from the 10th Incident of the global list and ordered by the creation date.

Request

Uncheck all the previous parameters in the "Query Params" list

Add the following parameters in the "Query Params" list

- **offset=10**
- **limit=20**
- **sort=-createdAt**



The screenshot shows a REST client interface with a GET request to the URL `https://api.orange.com/incident/b2b/v1/incidents?offset=10&limit=20&sort=-createdAt`. The 'Params' tab is active, displaying a table of query parameters:

Key	Value
<input checked="" type="checkbox"/> offset	10
<input checked="" type="checkbox"/> limit	20
<input checked="" type="checkbox"/> sort	-createdAt
Key	Value

Response

List of 20 incidents in a Json format

Use case 4: Get details of an Incident

With the Incidents for Business API, you can retrieve all information about a specific incident. You only need the id of the incident (you can retrieve it from the incidents list).

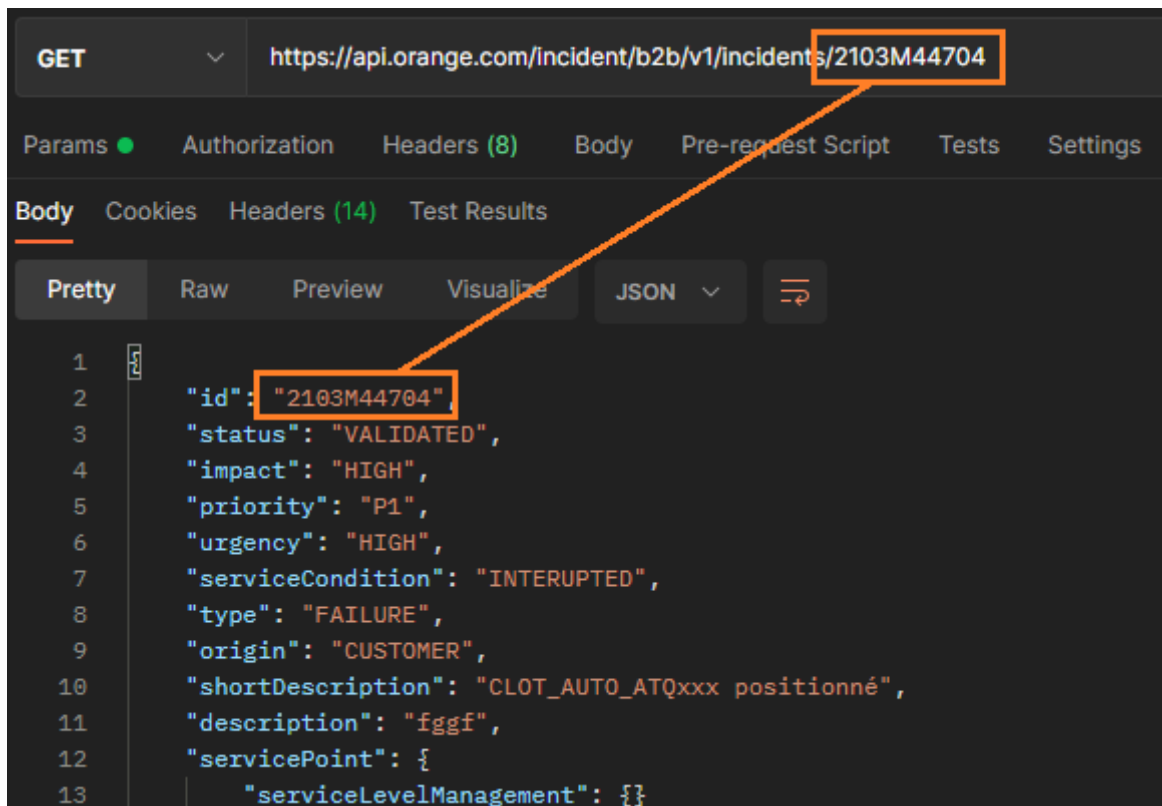
Request

Uncheck all the parameters in the "Query Params" list

Add the id of the incident (you can retrieve it from the incidents list) to the URL : `/{incident_id}`

Response

Your incident in a json format



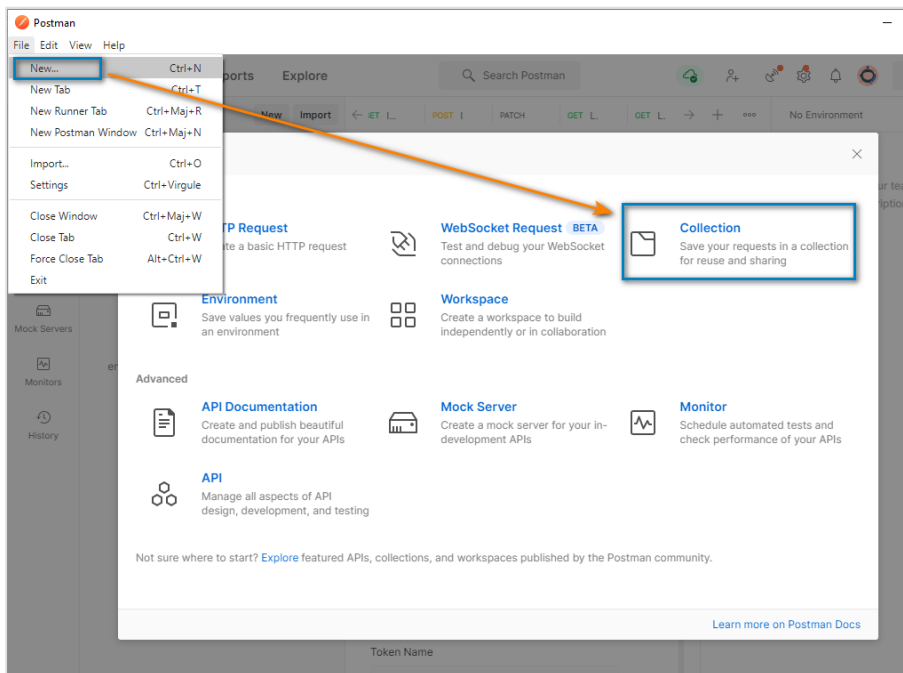
The screenshot shows a REST client interface. At the top, the method is set to 'GET' and the URL is `https://api.orange.com/incident/b2b/v1/incidents/2103M44704`. The 'Body' tab is selected, and the response is displayed in 'Pretty' JSON format. The response is a JSON object with the following fields: 'id', 'status', 'impact', 'priority', 'urgency', 'serviceCondition', 'type', 'origin', 'shortDescription', 'description', 'servicePoint', and 'serviceLevelManagement'. The 'id' field value is '2103M44704'. An orange arrow points from the 'id' field in the response to the '2103M44704' part of the URL in the request.

```
1  {
2    "id": "2103M44704",
3    "status": "VALIDATED",
4    "impact": "HIGH",
5    "priority": "P1",
6    "urgency": "HIGH",
7    "serviceCondition": "INTERUPTED",
8    "type": "FAILURE",
9    "origin": "CUSTOMER",
10   "shortDescription": "CLOT_AUTO_ATQxxx positionné",
11   "description": "fggf",
12   "servicePoint": {
13     "serviceLevelManagement": {}
```

Annexes: Configure API OAuth 2.0 authentication in Postman

Step 1: Create a new Collection

File > New > Select “Collection”



Step 2: Configure OAuth 2.0 authentication

In the “Authorization” tab :

- Type : OAuth 2.0

In the configure “New Token Options” section :

- Token Name : Set a name to your Token
- Grant Type : Clients Credentials
- Access Token URL : <https://api.orange.com/oauth/v3/token>
- Client ID : info provided from your Orange Developer Account
- Client Secret : info provided from your Orange Developer Account
- Scope : Ø
- Client Authentication : Send as Basic Auth header

New Collection Watch 0

Authorization Pre-request Script Tests Variables

This authorization method will be used for every request in this collection. You can override this by specifying one in the request.

Type OAuth 2.0

The authorization data will be automatically generated when you send the request.
[Learn more about authorization](#)

Add auth data to Request Headers

Current Token

Access Token Available Tokens
 Access Token

Header Prefix Bearer

Configure New Token Configuration Options Advanced Options

Token Name Token API4B

Grant Type Client Credentials

Access Token URL https://api.orange.com/oauth/v2/toker

Client ID

Client Secret

Scope e.g. read:org

Client Authentication Send as Basic Auth header

Clear cookies

Get New Access Token

Step 3: Get Access Token

Click on “Use Token”

MANAGE ACCESS TOKENS

All Tokens Delete

Token NSE 04/21

Token API4B

Use Token

Token Details

Token Name Token API4B

Access Token

Token Type Bearer

expires_in 7776000

The new token is displayed on the previous page in the “Current Token” section

Current Token

This access token is only available to you. Sync the token to let collaborators on this request use it.

Access Token Available Tokens ▾

✕

Header Prefix ⓘ Bearer

Step 4: Create a new Request and use the OAuth 2.0 authentication

Create a new request into the Collection you have created previously

In the “Authorization” tab of the new request, select “Inherit auth from Parent”

