Contents

I. How to order URBA360 1

1. Searching a company: 1
2. Unique identification of the company/ searching via externalId: 2
3. Ordering URBA360 package 4
4. Checking the status of the order: 5
5. Getting the URBA360 content 7

II. How to order URBA360 Monitoring 10

1. Searching a company 10
2. Company identification/ search via externalId 10
3. Ordering URBA360 10
4. Ordering URBA360 Monitoring 10
5. Checking the status of the monitoring order: 11
6. Getting the URBA360 content 13
7. Getting the notifications 13

III. How to get URBA360 orders 16

I. How to order URBA360

The process is the same as for the other Coface product:

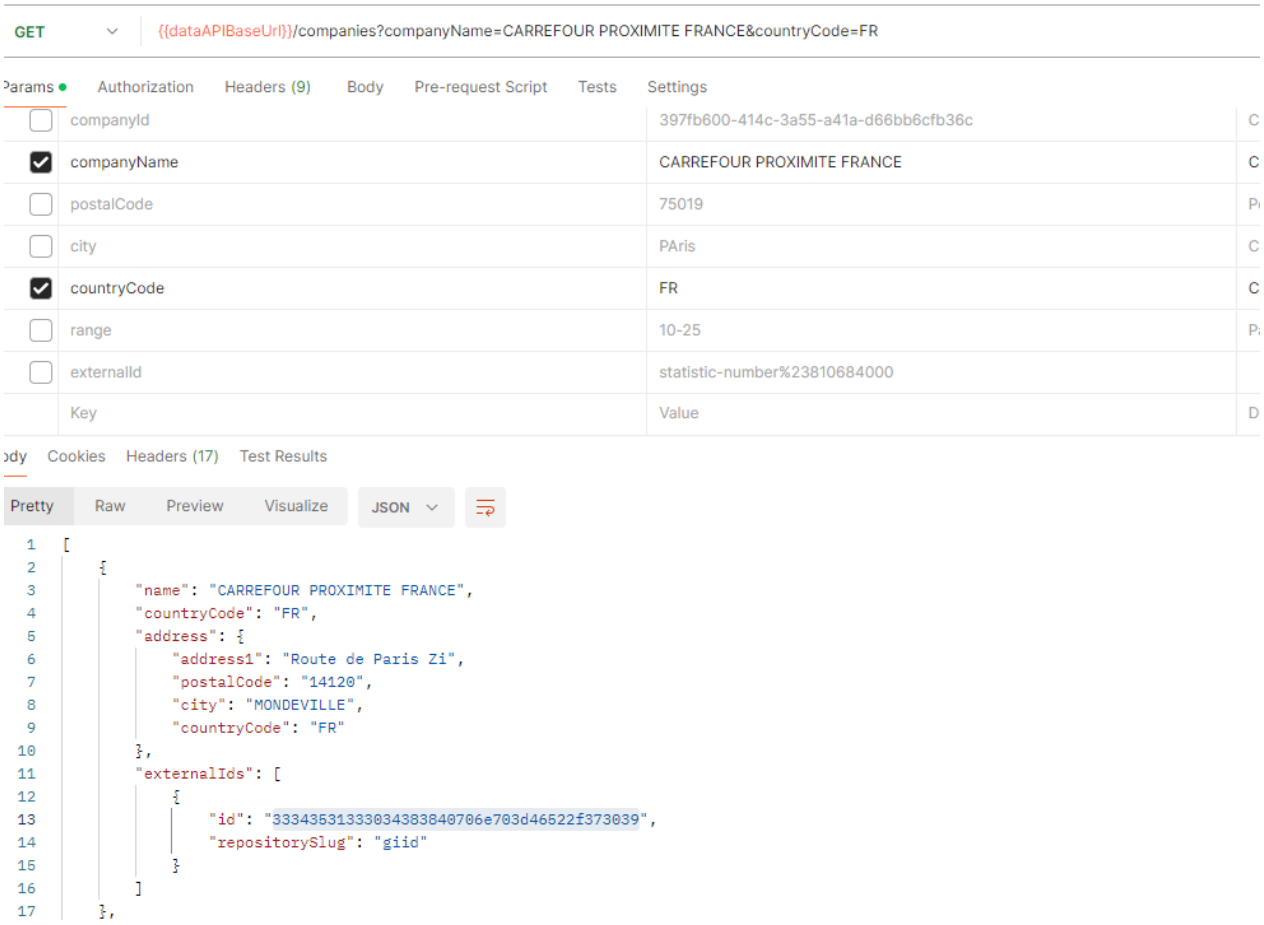
- you need first to search a company,
- unique identify that company,
- and then you can order URBA360 package.
- check the status of the order
- when status is ready => download the content.

1. Searching a company:

Endpoint: *GET /companies*

Parameters: countryCode and companyName

GET {{dataAPIBaseUrl}}/companies?companyName=CARREFOUR&countryCode=FR



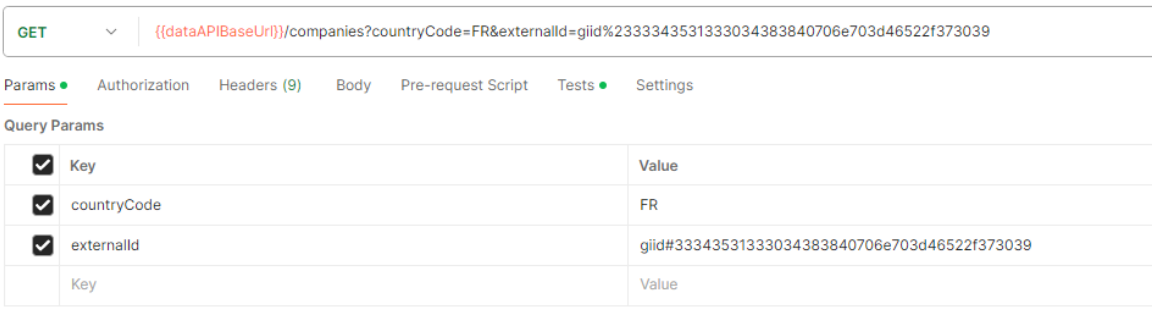
2. Unique identification of the company/ searching via externalId:

2.1 Via GIID returned when searched via Company Name and Country

Endpoint: *GET /companies*

Parameters: countryCode together with giid returned by previous call

GET /companies?countryCode=FR&externalId=giid%233334353133303438384 0706e703d46522f373039



Now, you have all identifiers (companyId OR iconNo OR easyNo or vatNo) needed for unique identification and you can order URBA360.

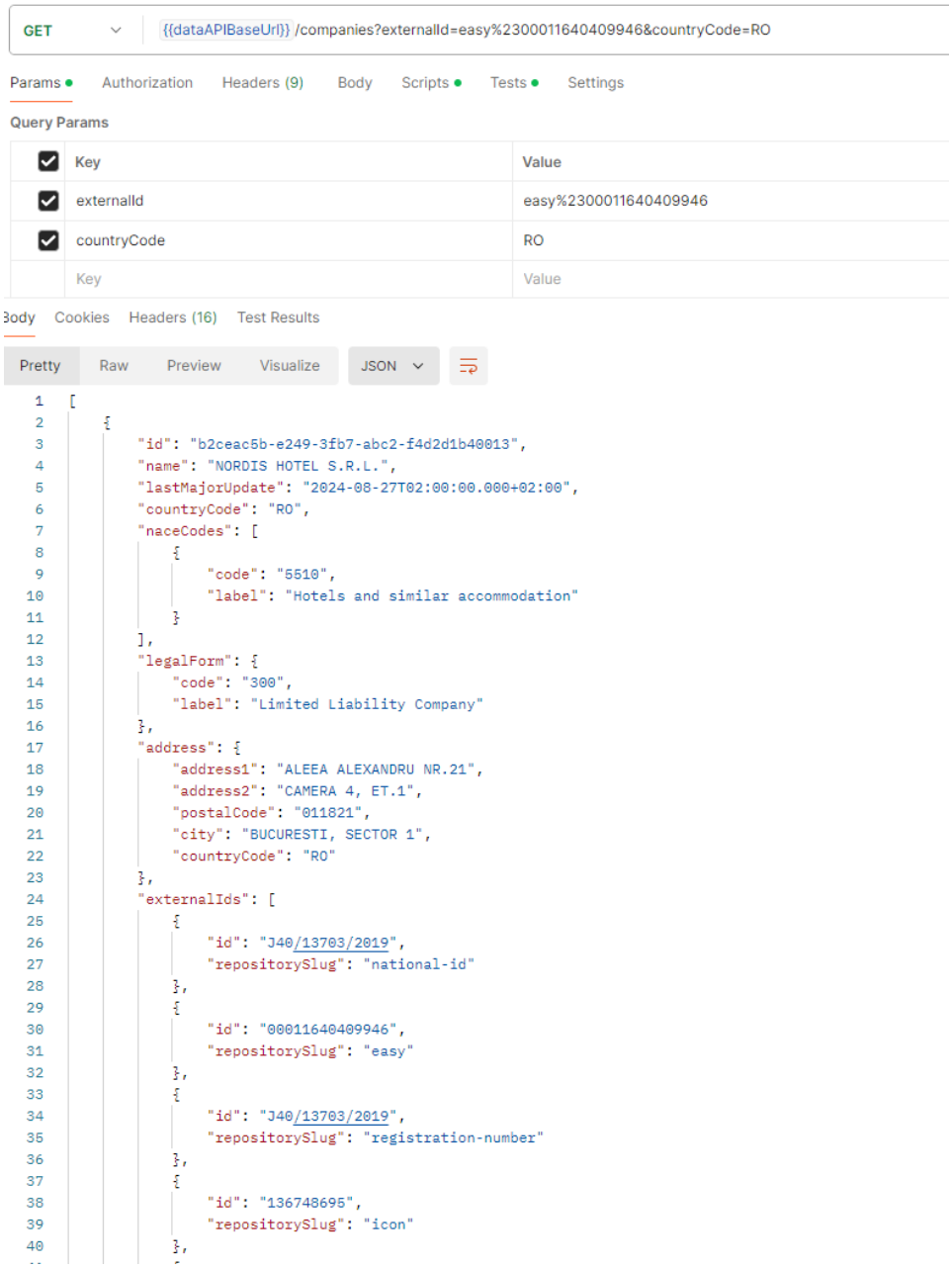
2.2 Via other externalId (e.g iconNo, easy number, registration-number, vat number)

In this moment there is no unique identifier across all the countries. Icon NO is the unique identifier used in this moment inside ICON Coface, but it also has the possibility of changing if Coface would change its external providers. Nevertheless, we are usually preparing workarounds to provide to our customers the new iconNO.

Therefore, is hard to guide to one unique identifier to be used in searching. Our recommendation is to always make the whole process: search the company giving most of the information: name, country, address, known legal identifiers.

Via easyNo:

GET /companies?externalId=easy%2300011640409946&countryCode=RO



3. Ordering URBA360 package

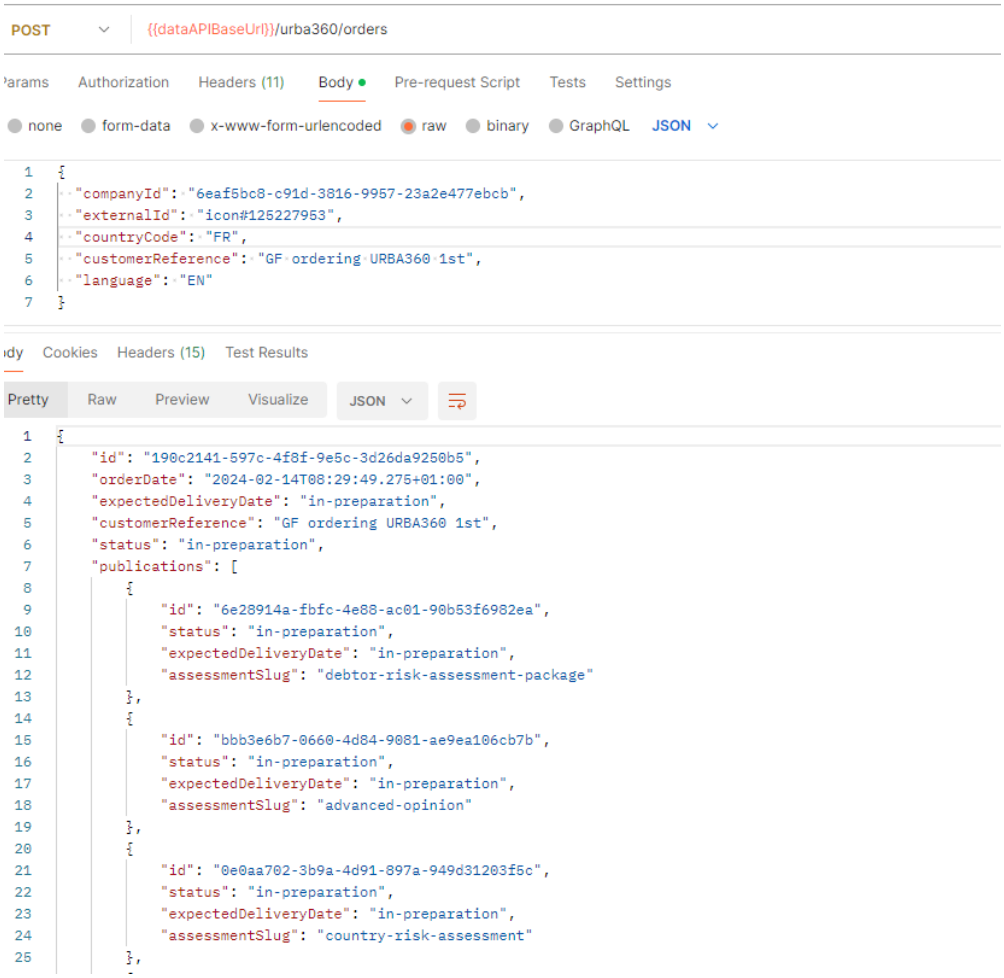
Endpoint: *POST {{dataAPIBaseUrl}}/urba360/orders*

Body Request:

```
{
  "companyId": "6eaf5bc8-c91d-3816-9957-23a2e477ebcb",
```

```
  "externalId": "icon#125227953",
  "countryCode": "FR",
  "customerReference": "GF ordering URBA360 1st",
  "language": "EN",
  "callbackUrl": "string"
}
```

Response:



4. Checking the status of the order:

There are two possibilities of checking the status of the order, depending on how the order was placed.

4.1 Callback mechanism was used

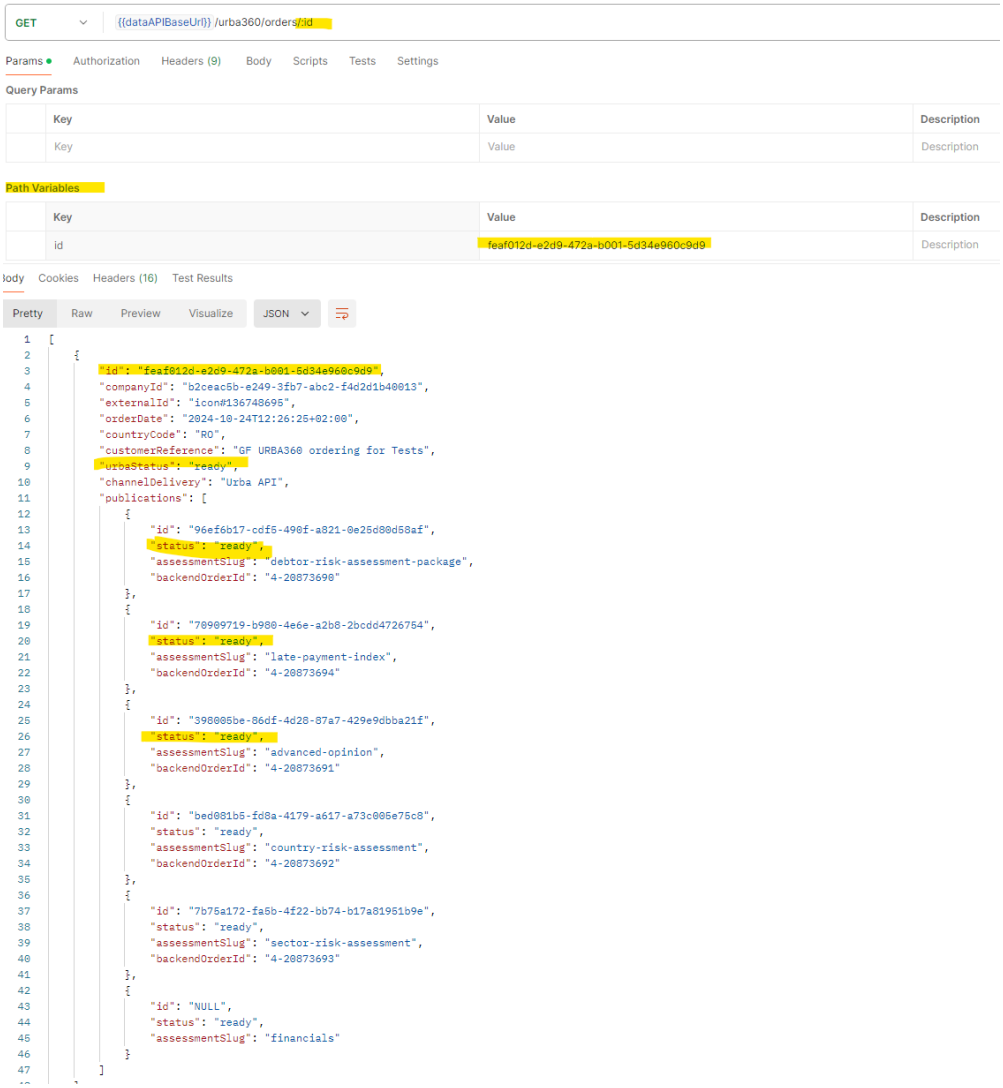
When all internal orders corresponding to all the URBA products(components) are having a final status (ready or cancelled) then one callback call is made to the URL provided in the body of the request when

posting the order. When ping is done and customer is receiving the call, he can download the content of URBA360 package via corresponding endpoint.

See section Download URBA360 content

4.2 No Callback mechanism

Endpoint: *GET {{dataAPIBaseUrl}}/urba360/orders/:id*
Path Variables: global id (first id from json that you received in the response of the POST method).



Response: it will contain the status of all orders that are created in our backend. The ids received can be used for getting parts of URBA separately, and the global Id can be used to get the whole JSON file.

Remark: The difference is that the big JSON contains also raw data corresponding to **peer groups and financials**, while if you use each individual id, you will get **only** corresponding data to those **products**: **late-payment-index**, **sector-risk-assessment**, **country-risk-assessment**, **advanced-opinion** and **debtor-risk-assessment**.

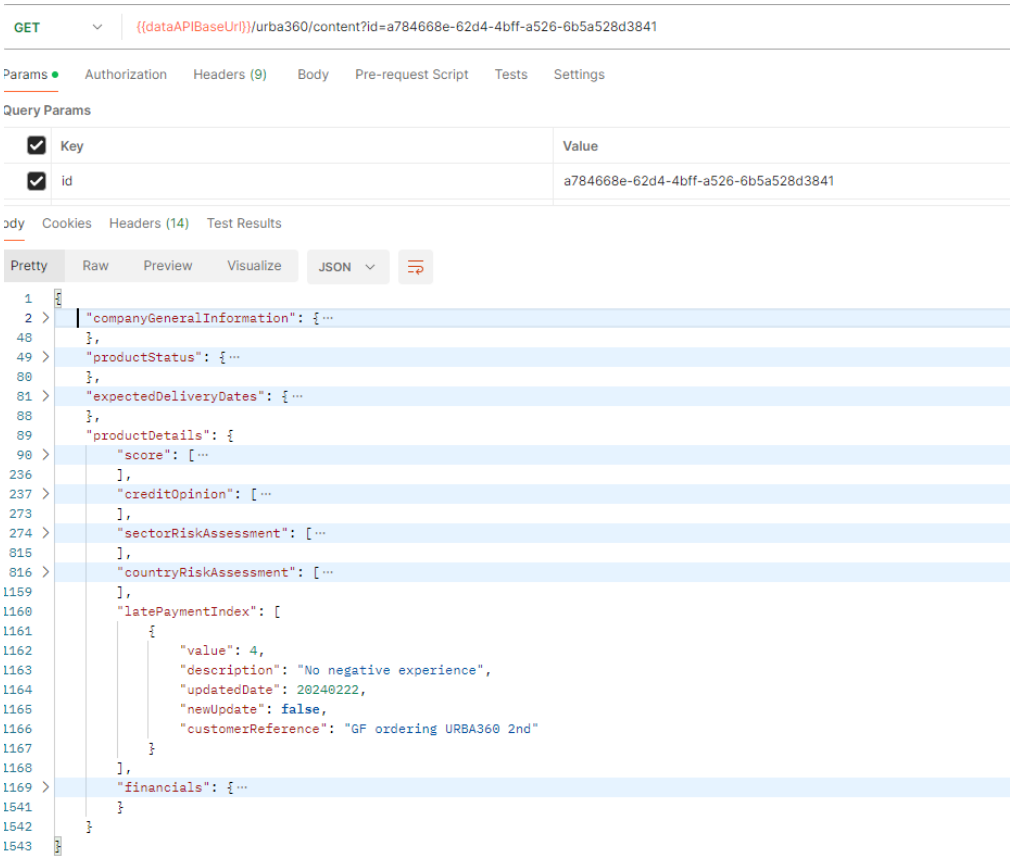
Important Remarks:

- Status can be different form one-suborder to another, as might be that some data takes a little bit longer to be available.
- As we are talking about several threads created for delivering the data, it might be that the systems is taking some time until the orders are created asynch in the backend. Therefore, if you are trying immediately (1 second after posting) to get the status of the URBA360 order you just have placed, you might be answered with a http status 202 Accepted, message: “Order is not yet ready”

5. Getting the URBA360 content

5.1 Getting the whole data in one file, including peer groups and financial info

Endpoint: *GET {{dataAPIBaseUrl}}/urba360/content?id=a784668e-62d4-4bff-a526-6b5a528d3841*
Query parameter: global id (first id from json that you received in the response of the POST method).
Response body: one JSON with all URBA360 information split in several sections: companyGeneralInformation, productStatus, productDetails (score, countryRiskAsessment, SectorRiskAssessment, latePaymentIndex, financials, creditOpinion).



5.2 Getting individual data for each of the URBA parts

Second option to get URBA data is to take product by product, using the ids received on POST call or from getting the status of the URBA order:

Endpoint for getting the status: *GET {{dataAPIBaseUrl}}/urba360/orders/:id*

Response:

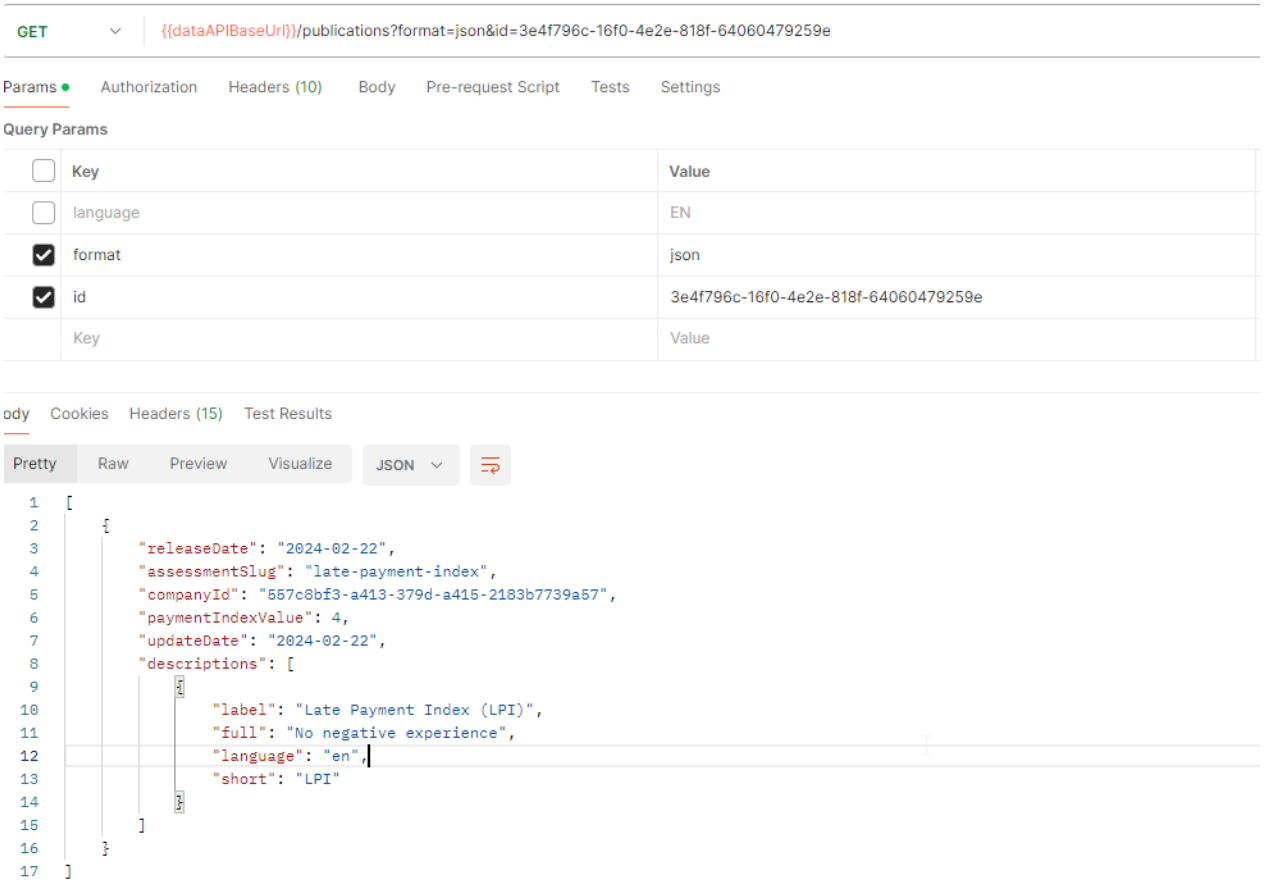

```
[{"id": "a784668e-62d4-4bff-a526-6b5a528d3841",
  "companyId": "557c8bf3-a413-379d-a415-2183b7739a57",
  "externalId": "icon#127524070",
  "orderDate": "2024-02-22T10:42:06+01:00",
  "customerReference": "GF ordering URBA360 2nd",
  "urbaStatus": "ready",
  "publications": [
            {   "id": "31e26a22-febc-47f4-b716-2dc256012805",
                "status": "ready",
                "assessmentSlug": "sector-risk-assessment"},
            {   "id": "3e4f796c-16f0-4e2e-818f-64060479259e",
                "status": "ready",
                "assessmentSlug": "late-payment-index"},
            {   "id": "66a60867-118d-47bc-8bbd-74852d3d4f76",
                "status": "ready",
                "assessmentSlug": "advanced-opinion"},
            {   "id": "9d2da226-9814-4e00-895f-797bf0c44716",
                "status": "ready",
                "assessmentSlug": "debtor-risk-assessment-package"},
            {   "id": "147b32c0-ecc9-408c-bf65-db7201a0f982",
                "status": "ready",
                "assessmentSlug": "country-risk-assessment"}
] }]
```


Now, you can take the content of any of the parts from URBA360.

For example, getting late-payment-index content:

Endpoint: *GET {{dataAPIBaseUrl}}/publications?format=json&id=3e4f796c-16f0-4e2e-818f-64060479259e*, where the id is the one from corresponding order.

Response: only the content of Late Payment index:



In the same way, all other contents can be extracted separately.

Except financials and peers group data which are available only in the big content of urba360 file.

Important remark: The whole content of URBA360 package can be downloaded when the status of the global order is “ready”. In case the status is “partial_ready” then it means that some of the internal orders are not finished yet. In this case the JSON will not be completed. Corresponding sections will be empty.

And in the case you want to download individually the content of the products, if corresponding orders are not ready, you will receive a Bad Request answer when trying to download the content.

II. How to order URBA360 Monitoring

Prerequisite: You will not be able to order an URBA360 Monitoring order until you have placed a simple URBA360 package order for that specific company

The process is the same as for the other Coface product or as for URBA360 package ordering, with small changes:

- search a company
- unique identify corresponding company (if case)
- order URBA360 package for corresponding company
- order URBA360 Monitoring for corresponding company
- check the status of the placed order
- when status is ready => download the content of the package
- listen for changes/ notifications
- when notifications are generated, read the content of the notification OR/AND download again the content of the package (it should be already updated)

1. Searching a company

Please check Searching a company section

2. Company identification/ search via externalId

Please check Company Identification section

3. Ordering URBA360

Please check Ordering UrbA360

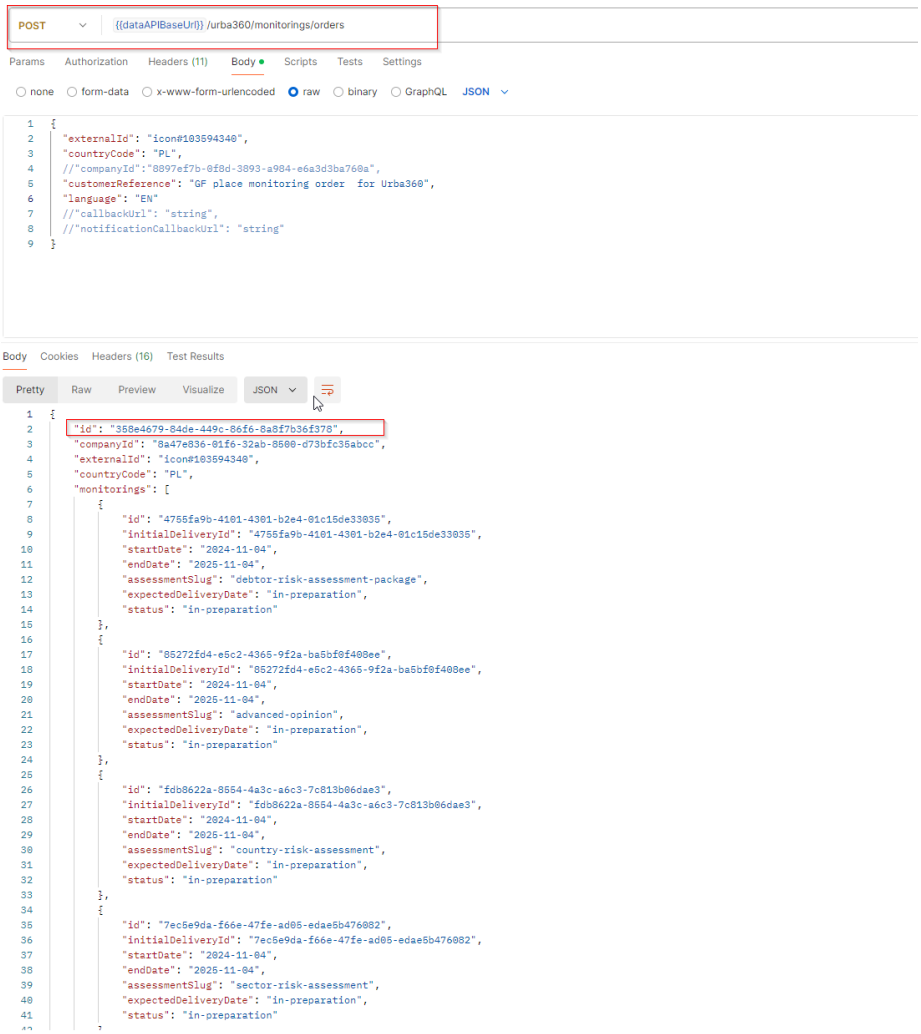
4. Ordering URBA360 Monitoring

Endpoint: *POST {{dataAPIBaseUrl}}/urba360/orders*

Body Request:

```
{
  "companyId": "6eaf5bc8-c91d-3816-9957-23a2e477ebcb",
  "externalId": "icon#125227953",
  "countryCode": "FR",
  "customerReference": " OUR TRACKING NUMBER #34234",
  "language": "EN",
  "callbackUrl": "string",
  "notificationCallbackUrl": "string"
}
```

Response:



5. Checking the status of the monitoring order:

There are two possibilities of checking the status of the order, depending on how the order was placed.

4.1 Callback mechanism was used

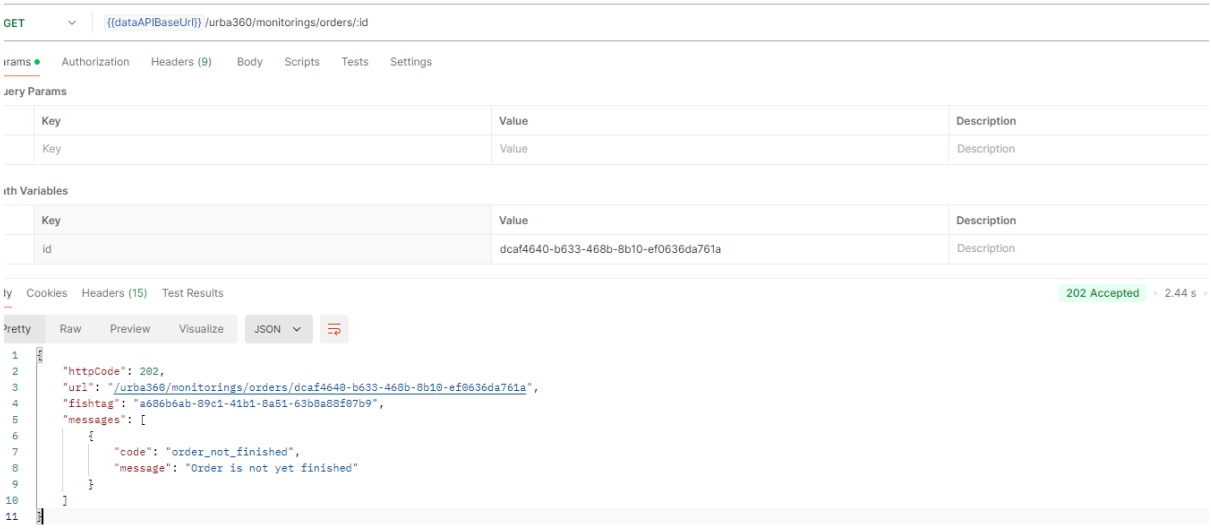
When all internal orders corresponding to all the URBA products(components) are having a final status (ready or cancelled) then one callback call is made to the URL provided in the body of the request when posting the order, in the field "**callbackUrl".** When ping is done and customer is receiving the call, he can download the content of URBA360 package via corresponding endpoint. It is the same endpoint as for a simple URBA360 package order.

See section Download URBA360 content

4.2 No Callback mechanism

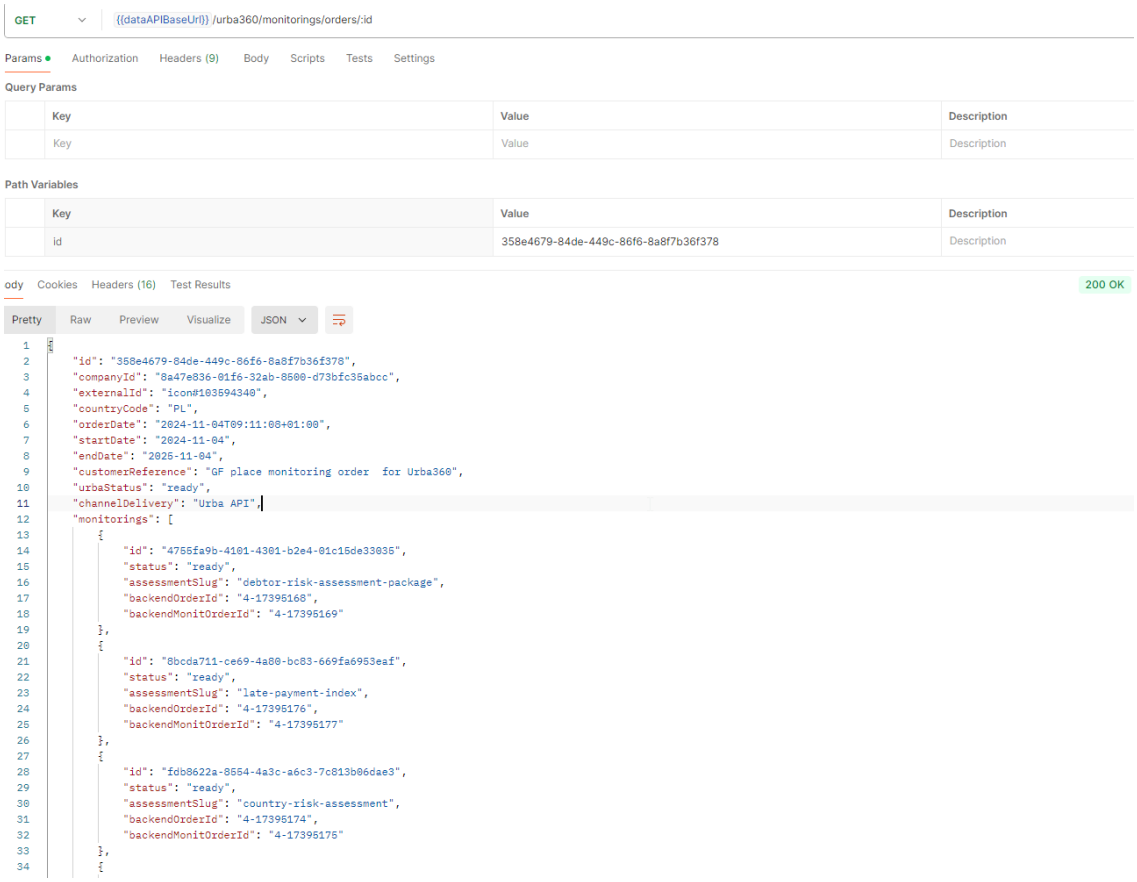
Important remarks:

1. The whole content of URBA360 package can be downloaded when the status of the global order is “ready”. In case the status is “partial_ready” then it means that some of the internal orders are not finished yet. In this case, the JSON will not be completed. Corresponding sections will be empty.
2. As we are talking about several threads created for delivering the data, it might be that the systems is taking some time until the orders are created asynch in the backend. Therefore, if you are trying immediately (1 second after posting) to get the status of the URBA360 monitoring order you just have placed, you might be answered with a http status 202 Accepted, message: “Order is not yet ready”



Endpoint: GET *{{dataAPIBaseUrl}}/urba360/monitorings/orders/:id*
Path Variables: global id (first id from json that you received in the response of the POST method).

Response: it will contain the status of all orders that are created in our backend. The ids received can be used for getting parts of URBA separately, and the global Id can be used to get the whole JSON file.



6. Getting the URBA360 content

It is the same process as for a simple URBA360 package order: with the id value returned by the POST command we can get the whole URBA content in JSON format.

Please check the details under the section [Getting the URBA360 package content]

7. Getting the notifications

As a customer which ordered monitoring (you have companies monitored) you will need to fetch the changes which are happening on these monitored companies.

Notifications are created per insight from URBA. Meaning: if a change happened to DRA, a new notification is generated. If a second change happened to LPI, a second notification is generated.

If you are already familiar with ordering standalone products (Reports, CRA, SRA, LPI, AO) then it is good to know that notifications are the same as for standalone products: notification for DRA, notification for LPI, etc.

There are 2 ways of getting the URBA360 notifications:

- using Callback mechanism or
- pulling regularly the notifications.

Difference between the 2 options: for the first one you will perform actions to get the notifications only after ping is sent to provided URL; while for the second method, you are doing actions without any trigger; when you want, depending on your need- You can do it regularly: daily, weekly, etc.

7.1 Notification Callback mechanism was used

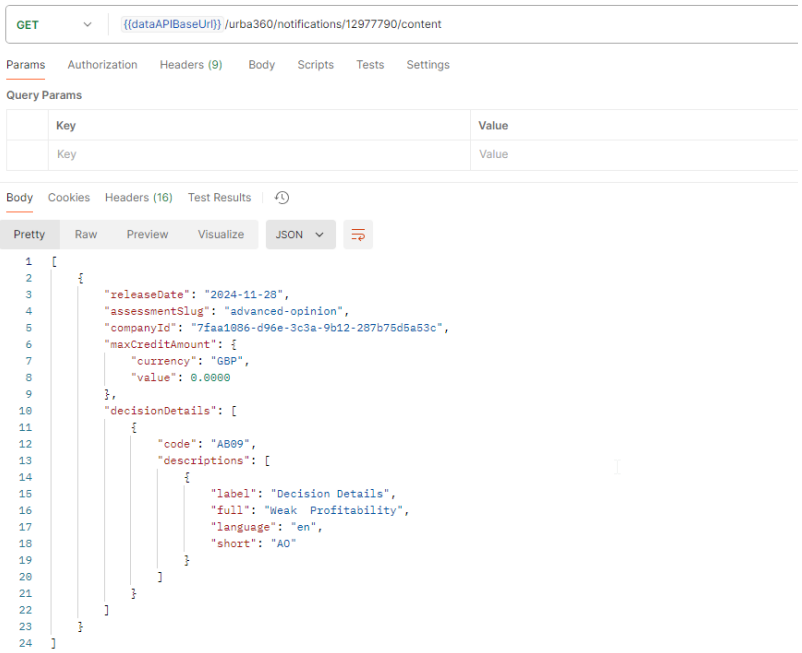
When POST- ing the URBA360 monitoring order, a callback url was provided inside the field `"notificationCallbackUrl": "string".` This means that with each and every change that is happening on any of the URBA360 parts which are monitored, a ping is sent to this specified URL.
It is just a simple ping call, we do not send any order info related to this ping.
Therefore, when you are receiving the ping, then following actions are needed:

a. check which notifications are new. You can do it using below endpoint
Endpoint: GET *{{dataAPIBaseUrl}}/urba360/notifications*

using corresponding filter based on your need:
GET *{{dataAPIBaseUrl}}/urba360/notifications?deliveryDateFrom=2024-07-29&deliveryDateTo=2024-07-29*
Result is a list of notifications received for the specified period, no matter the monitoring order / company monitored.

b. take the corresponding notificationId and download its content:

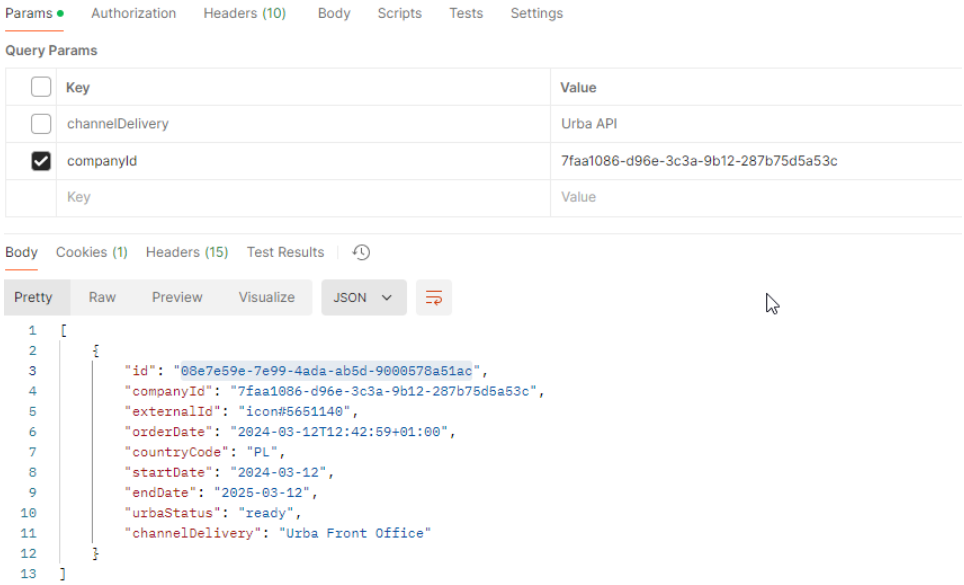
Endpoint: GET *{{dataAPIBaseUrl}}/urba360/notifications/12977790/content*



You can use the content of each notification to update data on your system, or you can download again the whole JSON file which is updated with each change.

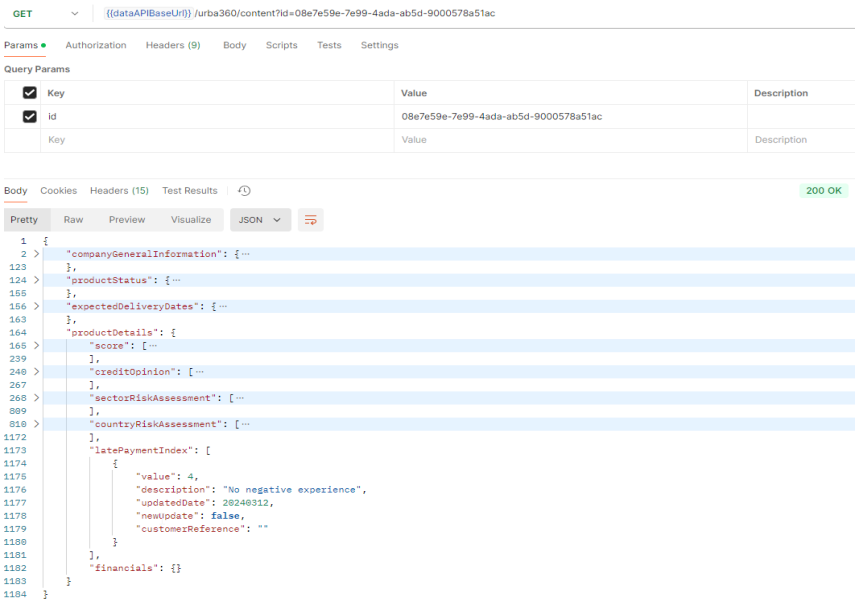
Identify the urba360 monitoring order based on notificationId:

- you can use the companyId field and find the corresponding URBA360 monitoring order:
  GET *{{dataAPIBaseUrl}}/urba360/monitorings/orders?companyId=7faa1086-d96e-3c3a-9b12-287b75d5a53c*



- With this id you can get the whole JSON content, updated with the changes corresponding to notifications:
  GET *{{dataAPIBaseUrl}}/urba360/content?id=7faa1086-d96e-3c3a-9b12-287b75d5a53c*

Response is the updated Json (is containing the changes from the notifications)



7.2 No Callback mechanism used

All actions from above (point 7.1) need to be done also in this case. The difference is that in this option, there is no ping which will trigger them, but you will run them based on your need: each day – to check and download the notifications daily, or at every 2 days, or weekly, etc.

III. How to get URBA360 orders

Via API you can see all orders performed under your customer profile independent of the channel: frontend application, API or via backend user.

1. Get URBA360 orders

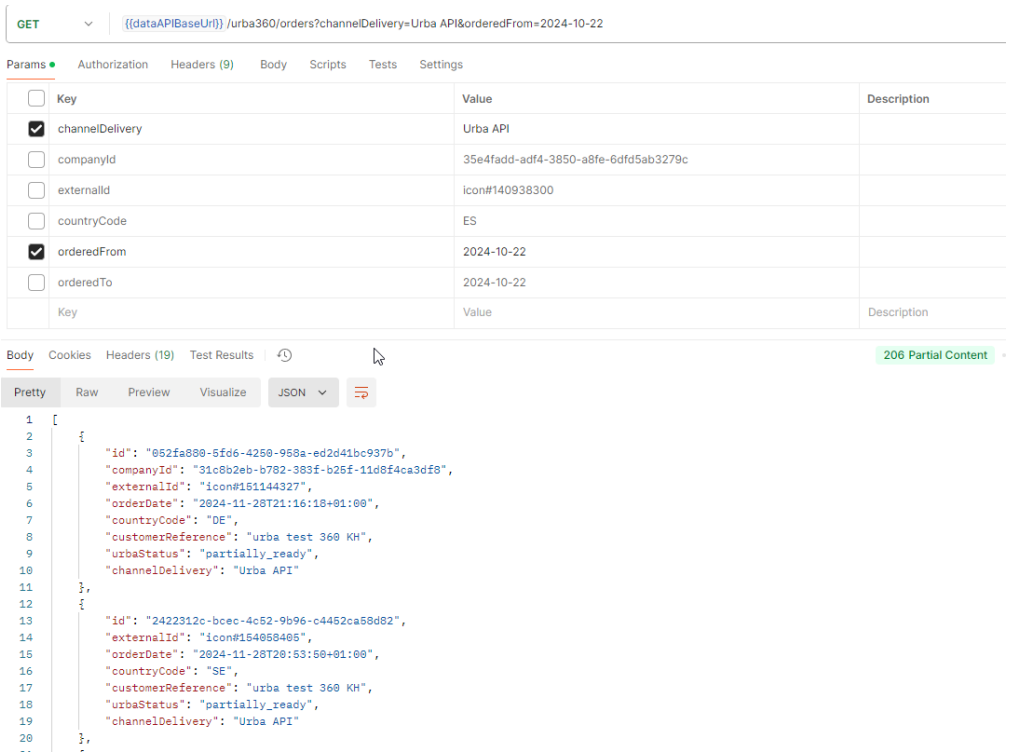
Endpoint *GET {{dataAPIBaseUrl}}/urba360/orders*

Response: all URBA360 orders independently of where they were placed.

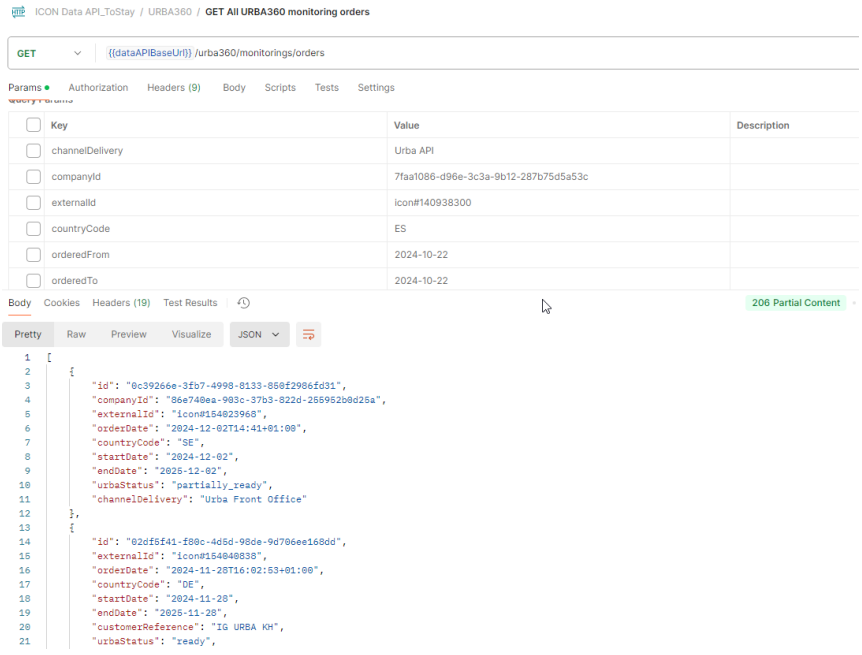
Response can be filtered using any of the following parameters: companyId, externalid +countryCode, orderedFrom, orderedTo, deliveredFrom, deliveredTo, customerReference, channelDelivery

Response is paginated and range parameter can be used for parsing the whole list. Syntax is offset- limit with a default and maximum value of the limit of 20 results per page. If multiple pages are returned, the http status code of the call is 206 Partial Content.

Details about each parameter can be found also in the swagger file.



2. GET URBA360 monitorings orders
   Endpoint *GET {{dataAPIBaseUrl}}/urba360/monitorings/orders*

Response: all URBA360 monitorings orders independently of channelused for placing them. Response can be filtered using any of the following parameters: companyId, externalId +countryCode, orderedFrom, orderedTo, deliveredFrom, deliveredTo, customerReference, channelDelivery
Response is paginated and range parameter can be used for parsing the whole list. Syntax is offset- limit with a default and maximum value of the limit of 20 results per page. If multiple pages are returned, the http status code of the call is 206 Partial Content.

Details about each parameter can be found also in the swagger file.