

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

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](#)

GET ▼ `{{dataAPIBaseUrl}}/companies?companyName=CARREFOUR PROXIMATE FRANCE&countryCode=FR`

Params ● Authorization Headers (9) Body Pre-request Script Tests Settings

<input type="checkbox"/>	companyId	397fb600-414c-3a55-a41a-d66bb6cfb36c	C
<input checked="" type="checkbox"/>	companyName	CARREFOUR PROXIMATE FRANCE	C
<input type="checkbox"/>	postalCode	75019	P
<input type="checkbox"/>	city	PARIS	C
<input checked="" type="checkbox"/>	countryCode	FR	C
<input type="checkbox"/>	range	10-25	P
<input type="checkbox"/>	externalId	statistic-number%23810684000	
	Key	Value	D

Body Cookies Headers (17) Test Results

Pretty Raw Preview Visualize JSON ▼ ≡

```

1  [
2    {
3      "name": "CARREFOUR PROXIMATE FRANCE",
4      "countryCode": "FR",
5      "address": {
6        "address1": "Route de Paris Zi",
7        "postalCode": "14120",
8        "city": "MONDEVILLE",
9        "countryCode": "FR"
10     },
11     "externalIds": [
12       {
13         "id": "33343531333034383840706e703d46522f373039",
14         "repositorySlug": "giid"
15       }
16     ]
17   },

```

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%2333343531333034383840706e703d46522f373039](#)

GET ⌵ {{dataAPIBaseUrl}}/companies?countryCode=FR&externalId=gliid#2333343531333034383840706e703d46522f373039

Params ●

Authorization

Headers (9)

Body

Pre-request Script

Tests ●

Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	countryCode	FR
<input checked="" type="checkbox"/>	externalId	gliid#33343531333034383840706e703d46522f373039
	Key	Value

Body	Cookies	Headers (15)	Test Results
Pretty	Raw	Preview	Visualize
JSON ⌵			
1	[
2	{		
3	"id": "6eaf5bc8-c91d-3816-9957-23a2e477ebcb",		
4	"name": "CARREFOUR PROXIMATE FRANCE",		
5	"lastMajorUpdate": "2024-02-14T01:00:00.000+01:00",		
6	"countryCode": "FR",		
7	"naceCodes": [
8	{		
9	"code": "7740",		
10	"label": "Leasing of intellectual property and similar products, except copyrighted works"		
11	}		
12],		

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](#)

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

Params
Authorization
Headers (9)
Body
Scripts
Tests
Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	externalId	easy%2300011640409946
<input checked="" type="checkbox"/>	countryCode	RO
	Key	Value

Body
Cookies
Headers (16)
Test Results

Pretty
Raw
Preview
Visualize
JSON

```

1  [
2    {
3      "id": "b2ceac5b-e249-3fb7-abc2-f4d2d1b40013",
4      "name": "NORDIS HOTEL S.R.L.",
5      "lastMajorUpdate": "2024-08-27T02:00:00.000+02:00",
6      "countryCode": "RO",
7      "naceCodes": [
8        {
9          "code": "5510",
10         "label": "Hotels and similar accommodation"
11       }
12     ],
13     "legalForm": {
14       "code": "300",
15       "label": "Limited Liability Company"
16     },
17     "address": {
18       "address1": "ALEEA ALEXANDRU NR.21",
19       "address2": "CAMERA 4, ET.1",
20       "postalCode": "011821",
21       "city": "BUCURESTI, SECTOR 1",
22       "countryCode": "RO"
23     },
24     "externalIds": [
25       {
26         "id": "J40/13703/2019",
27         "repositorySlug": "national-id"
28       },
29       {
30         "id": "00011640409946",
31         "repositorySlug": "easy"
32       },
33       {
34         "id": "J40/13703/2019",
35         "repositorySlug": "registration-number"
36       },
37       {
38         "id": "136748695",
39         "repositorySlug": "icon"
40       }
41     ]
42   }
43 ]

```

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:

The screenshot shows a REST client interface with a POST request to `{{dataAPIBaseUrl}}/urba360/orders`. The request body is a JSON object with the following fields: `companyId`, `externalId`, `countryCode`, `customerReference`, and `language`. The response is a JSON object with the following fields: `id`, `orderDate`, `expectedDeliveryDate`, `customerReference`, `status`, and `publications`. The `publications` field is an array of three objects, each containing `id`, `status`, `expectedDeliveryDate`, and `assessmentSlug`.

```

1 {
2   "companyId": "6eaf5bc8-c91d-3816-9957-23a2e477ebcb",
3   "externalId": "icon#125227953",
4   "countryCode": "FR",
5   "customerReference": "GF ordering URBA360 1st",
6   "language": "EN"
7 }

```

```

1 {
2   "id": "190c2141-597c-4f8f-9e5c-3d26da9250b5",
3   "orderDate": "2024-02-14T08:29:49.275+01:00",
4   "expectedDeliveryDate": "in-preparation",
5   "customerReference": "GF ordering URBA360 1st",
6   "status": "in-preparation",
7   "publications": [
8     {
9       "id": "6e28914a-fbfc-4e88-ac01-90b53f6982ea",
10      "status": "in-preparation",
11      "expectedDeliveryDate": "in-preparation",
12      "assessmentSlug": "debtor-risk-assessment-package"
13    },
14    {
15      "id": "bbb3e6b7-0660-4d84-9081-ae9ea106cb7b",
16      "status": "in-preparation",
17      "expectedDeliveryDate": "in-preparation",
18      "assessmentSlug": "advanced-opinion"
19    },
20    {
21      "id": "0e0aa702-3b9a-4d91-897a-949d31203f5c",
22      "status": "in-preparation",
23      "expectedDeliveryDate": "in-preparation",
24      "assessmentSlug": "country-risk-assessment"
25    }
26  ]
27 }

```

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).

The screenshot shows a REST client interface with a GET request to the endpoint `{{dataAPIBaseUrl}}/urba360/orders/:id`. The path variable `id` is set to `feaf012d-e2d9-472a-b001-5d34e960c9d9`. The response is a JSON array of order objects, each containing an `id`, `status`, `assessmentSlug`, and `backendOrderId`.

```
1 [
2   {
3     "id": "feaf012d-e2d9-472a-b001-5d34e960c9d9",
4     "companyId": "b2ceac5b-e249-3fb7-abc2-f4d2d1b40013",
5     "externalId": "icon136748695",
6     "orderDate": "2024-10-24T12:26:12+02:00",
7     "countryCode": "RO",
8     "customerReference": "GF URBA360 ordering for Tests",
9     "orderStatus": "ready",
10    "channelDelivery": "Urba API",
11    "publications": [
12      {
13        "id": "96ef6b17-cdf5-490f-a821-0e25d08d58af",
14        "status": "ready",
15        "assessmentSlug": "debtor-risk-assessment-package",
16        "backendOrderId": "4-20873690"
17      },
18      {
19        "id": "70909719-b900-4e6e-a2b8-2bccd4726754",
20        "status": "ready",
21        "assessmentSlug": "late-payment-index",
22        "backendOrderId": "4-20873694"
23      },
24      {
25        "id": "398005be-86df-4d28-87a7-429e9dbba21f",
26        "status": "ready",
27        "assessmentSlug": "advanced-opinion",
28        "backendOrderId": "4-20873691"
29      },
30      {
31        "id": "bed081b5-fd8a-4179-a617-a73c006e75c8",
32        "status": "ready",
33        "assessmentSlug": "country-risk-assessment",
34        "backendOrderId": "4-20873692"
35      },
36      {
37        "id": "7b76a172-fa5b-4f22-bb74-b17a81951b9e",
38        "status": "ready",
39        "assessmentSlug": "sector-risk-assessment",
40        "backendOrderId": "4-20873693"
41      },
42      {
43        "id": "NULL",
44        "status": "ready",
45        "assessmentSlug": "financials"
46      }
47    ]
48  }
49 ]
```

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 from 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, countryRiskAssessment, SectorRiskAssessment, latePaymentIndex, financials, creditOpinion).

The screenshot shows a REST client interface with a GET request to the endpoint `{{dataAPIBaseUrl}}/urba360/content?id=a784668e-62d4-4bff-a526-6b5a528d3841`. The query parameter `id` is set to `a784668e-62d4-4bff-a526-6b5a528d3841`. The response body is displayed in JSON format, showing a complex object with various fields like `companyGeneralInformation`, `productStatus`, `productDetails`, and `financials`.

```
1  {
2    "companyGeneralInformation": { ...
48  },
49  "productStatus": { ...
80  },
81  "expectedDeliveryDates": { ...
88  },
89  "productDetails": {
90    "score": [ ...
236  ],
237    "creditOpinion": [ ...
273  ],
274    "sectorRiskAssessment": [ ...
815  ],
816    "countryRiskAssessment": [ ...
1159  ],
1160    "latePaymentIndex": [
1161      {
1162        "value": 4,
1163        "description": "No negative experience",
1164        "updatedAt": "20240222",
1165        "newUpdate": false,
1166        "customerReference": "GF ordering URBA360 2nd"
1167      }
1168    ],
1169    "financials": { ...
1541  }
1542 }
1543 }
```

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:

GET
{{dataAPIBaseUrl}}/publications?format=json&id=3e4f796c-16f0-4e2e-818f-64060479259e

Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings

Query Params

<input type="checkbox"/>	Key	Value
<input type="checkbox"/>	language	EN
<input checked="" type="checkbox"/>	format	json
<input checked="" type="checkbox"/>	id	3e4f796c-16f0-4e2e-818f-64060479259e
	Key	Value

body
Cookies
Headers (15)
Test Results

Pretty
Raw
Preview
Visualize
JSON

```

1  [
2    {
3      "releaseDate": "2024-02-22",
4      "assessmentSlug": "late-payment-index",
5      "companyId": "557c8bf3-a413-379d-a415-2183b7739a57",
6      "paymentIndexValue": 4,
7      "updateDate": "2024-02-22",
8      "descriptions": [
9        {
10         "label": "Late Payment Index (LPI)",
11         "full": "No negative experience",
12         "language": "en",
13         "short": "LPI"
14       }
15     ]
16   }
17 ]

```

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:

POST

{{dataAPIBaseUrl}}/urba360/monitorings/orders

Params

Authorization

Headers (11)

Body

Scripts

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```

1 {
2   "externalId": "icon#103594340",
3   "countryCode": "PL",
4   //"companyId": "8097ef7b-0f0d-3093-a904-e6a3d3ba760a",
5   "customerReference": "GF place monitoring order for Urba360",
6   "language": "EN"
7   //"callbackUrl": "string",
8   //"notificationCallbackUrl": "string"
9 }

```

Body

Cookies

Headers (16)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```

1 {
2   "id": "358e4679-84de-449c-86f6-8a8f7b36f378",
3   "companyId": "8a47e836-01f6-32ab-8500-d73bfc35abcc",
4   "externalId": "icon#103594340",
5   "countryCode": "PL",
6   "monitorings": [
7     {
8       "id": "4755fa9b-4101-4301-b2e4-01c15de33035",
9       "initialDeliveryId": "4755fa9b-4101-4301-b2e4-01c15de33035",
10      "startDate": "2024-11-04",
11      "endDate": "2026-11-04",
12      "assessmentSlug": "debtor-risk-assessment-package",
13      "expectedDeliveryDate": "in-preparation",
14      "status": "in-preparation"
15    },
16    {
17      "id": "85272fd4-e6c2-4366-9f2a-ba5bf0f408ee",
18      "initialDeliveryId": "85272fd4-e6c2-4366-9f2a-ba5bf0f408ee",
19      "startDate": "2024-11-04",
20      "endDate": "2026-11-04",
21      "assessmentSlug": "advanced-opinion",
22      "expectedDeliveryDate": "in-preparation",
23      "status": "in-preparation"
24    },
25    {
26      "id": "fdb8622a-8554-4a3c-a6c3-7c813b06dae3",
27      "initialDeliveryId": "fdb8622a-8554-4a3c-a6c3-7c813b06dae3",
28      "startDate": "2024-11-04",
29      "endDate": "2026-11-04",
30      "assessmentSlug": "country-risk-assessment",
31      "expectedDeliveryDate": "in-preparation",
32      "status": "in-preparation"
33    },
34    {
35      "id": "7ec5e9da-f66e-47fe-ad05-edae5b476082",
36      "initialDeliveryId": "7ec5e9da-f66e-47fe-ad05-edae5b476082",
37      "startDate": "2024-11-04",
38      "endDate": "2026-11-04",
39      "assessmentSlug": "sector-risk-assessment",
40      "expectedDeliveryDate": "in-preparation",
41      "status": "in-preparation"
42    }
43   ]
44 }

```

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"

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `{{dataAPIBaseUrl}}/urba360/monitorings/orders/:id`
- Headers:** 9 headers are listed, including Authorization, Headers, Body, Scripts, Tests, and Settings.
- Query Params:** A table with 3 columns: Key, Value, and Description. The first row shows 'Key' and 'Value'.
- Path Variables:** A table with 3 columns: Key, Value, and Description. The first row shows 'id' with the value 'dcaf4640-b633-468b-8b10-ef0636da761a'.
- Status:** 202 Accepted (2.44 s)
- Response Body:** A JSON object with the following structure:

```
{  "statusCode": 202,  "url": "/urba360/monitorings/orders/dcaf4640-b633-468b-8b10-ef0636da761a",  "fishtag": "a606b6ab-09c1-41b1-8a51-63b0a08f07b9",  "messages": [    {      "code": "order_not_finished",      "message": "Order is not yet finished"    }  ]}
```

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.

GET `{{dataAPIBaseUrl}}/urba360/monitorings/orders/id`

Params Authorization Headers (9) Body Scripts Tests Settings

Query Params

Key	Value	Description
Key	Value	Description

Path Variables

Key	Value	Description
id	358e4679-84de-449c-86f6-8a8f7b36f378	Description

ody Cookies Headers (16) Test Results 200 OK

Pretty Raw Preview Visualize JSON

```

1  {
2    "id": "358e4679-84de-449c-86f6-8a8f7b36f378",
3    "companyId": "8a47e836-81f6-32ab-8500-d73bfc35abcc",
4    "externalId": "icon8103594340",
5    "countryCode": "PL",
6    "orderDate": "2024-11-04T09:11:00+01:00",
7    "startDate": "2024-11-04",
8    "endDate": "2026-11-04",
9    "customerReference": "GF place monitoring order for Urba360",
10   "urbaStatus": "ready",
11   "channelDelivery": "Urba API",
12   "monitorings": [
13     {
14       "id": "4755fa9b-4101-4301-b2e4-01c15de33035",
15       "status": "ready",
16       "assessmentSlug": "debtor-risk-assessment-package",
17       "backendOrderId": "4-17395168",
18       "backendMonitOrderId": "4-17395169"
19     },
20     {
21       "id": "8bcda711-ce69-4a80-bc83-669fa6953eaf",
22       "status": "ready",
23       "assessmentSlug": "late-payment-index",
24       "backendOrderId": "4-17395176",
25       "backendMonitOrderId": "4-17395177"
26     },
27     {
28       "id": "fdb8622a-8554-4a3c-a6c3-7c813b06dae3",
29       "status": "ready",
30       "assessmentSlug": "country-risk-assessment",
31       "backendOrderId": "4-17395174",
32       "backendMonitOrderId": "4-17395175"
33     },
34   ]
35 }

```

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`

The screenshot shows a REST client interface with a GET request to the endpoint `{{dataAPIBaseUrl}}/urba360/notifications/12977790/content`. The response is displayed in JSON format, showing a list of notification details. The JSON structure includes fields for release date, assessment slug, company ID, max credit amount, currency, value, and decision details.

```
1 [
2   {
3     "releaseDate": "2024-11-28",
4     "assessmentSlug": "advanced-opinion",
5     "companyId": "7faa1896-d96e-3c3a-9b12-287b75d5a53c",
6     "maxCreditAmount": {
7       "currency": "GBP",
8       "value": 0.0000
9     },
10    "decisionDetails": [
11      {
12        "code": "A889",
13        "descriptions": [
14          {
15            "label": "Decision Details",
16            "full": "Weak Profitability",
17            "language": "en",
18            "short": "AO"
19          }
20        ]
21      }
22    ]
23  }
24 ]
```

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](#)

Params • Authorization Headers (10) Body Scripts Tests Settings		
Query Params		
<input type="checkbox"/>	Key	Value
<input type="checkbox"/>	channelDelivery	Urba API
<input checked="" type="checkbox"/>	companyId	7faa1086-d96e-3c3a-9b12-287b75d5a53c
	Key	Value

Body Cookies (1) Headers (15) Test Results ↻		
Pretty Raw Preview Visualize JSON ↕		
1	[
2	{	
3	"id": "08e7e59e-7e99-4ada-ab5d-9000578a51ac",	
4	"companyId": "7faa1086-d96e-3c3a-9b12-287b75d5a53c",	
5	"externalId": "icon#5651140",	
6	"orderDate": "2024-03-12T12:42:59+01:00",	
7	"countryCode": "PL",	
8	"startDate": "2024-03-12",	
9	"endDate": "2025-03-12",	
10	"urbaStatus": "ready",	
11	"channelDelivery": "Urba Front Office"	
12	}	
13]	

- 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)

GET

[[dataAPIBaseUrl]]/urba360/content?id=08e7e59e-7e99-4ada-ab5d-9000578a51ac

Params • Authorization Headers (9) Body Scripts Tests Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description
<input checked="" type="checkbox"/>	id	08e7e59e-7e99-4ada-ab5d-9000578a51ac	
	Key	Value	Description

Body Cookies Headers (15) Test Results 200 OK

Pretty Raw Preview Visualize JSON

```
1 {
2   > "companyGeneralInformation": { --
123   },
124   > "productStatus": { --
155   },
156   > "expectedDeliveryDates": { --
163   },
164   > "productDetails": {
165     > "score": { --
239     },
240     > "creditOpinion": { --
267     },
268     > "sectorRiskAssessment": { --
809     },
810     > "countryRiskAssessment": { --
1172     },
1173     > "latePaymentIndex": {
1174       {
1175         "value": 4,
1176         "description": "No negative experience",
1177         "updatedAt": "20240312",
1178         "newUpdate": false,
1179         "customerReference": ""
1180       },
1181       "financials": {}
1182     }
1183   }
1184 }
```

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.

GET `{{dataAPIBaseUrl}}/urba360/orders?channelDelivery=Urba API&orderedFrom=2024-10-22`

Params • Authorization Headers (9) Body Scripts Tests Settings

Key	Value	Description
<input checked="" type="checkbox"/> channelDelivery	Urba API	
<input type="checkbox"/> companyId	35e4fadd-adf4-3850-a8fe-6dfd5ab3279c	
<input type="checkbox"/> externalid	icon#140938300	
<input type="checkbox"/> countryCode	ES	
<input checked="" type="checkbox"/> orderedFrom	2024-10-22	
<input type="checkbox"/> orderedTo	2024-10-22	
Key	Value	Description

Body Cookies Headers (19) Test Results 206 Partial Content

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": "852fa880-5fd6-4258-958a-ed2d41bc937b",
4     "companyId": "31c8b2eb-b782-383f-b25f-11d8f4ca3df8",
5     "externalId": "icon#151144327",
6     "orderDate": "2024-11-28T21:16:18+01:00",
7     "countryCode": "DE",
8     "customerReference": "urba test 360 KH",
9     "urbaStatus": "partially_ready",
10    "channelDelivery": "Urba API"
11  },
12  {
13    "id": "2422312c-bcec-4c52-9b96-c4452ca58d82",
14    "externalId": "icon#154058465",
15    "orderDate": "2024-11-28T20:53:58+01:00",
16    "countryCode": "SE",
17    "customerReference": "urba test 360 KH",
18    "urbaStatus": "partially_ready",
19    "channelDelivery": "Urba API"
20  },
21 ]
```

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.

ICON Data API ToStay / URBA360 / GET All URBA360 monitoring orders

GET ((dataAPIBaseUri)) /urba360/monitorings/orders

Params Authorization Headers (9) Body Scripts Tests Settings

Key	Value	Description
<input type="checkbox"/> channelDelivery	Urba API	
<input type="checkbox"/> companyId	7faa1086-d96e-3c3a-9b12-287b75d5a53c	
<input type="checkbox"/> externalId	icon#140938300	
<input type="checkbox"/> countryCode	ES	
<input type="checkbox"/> orderedFrom	2024-10-22	
<input type="checkbox"/> orderedTo	2024-10-22	

Body Cookies Headers (19) Test Results 206 Partial Content

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": "0c39266e-3fb7-4998-8133-858f2986fd31",
4     "companyId": "86e748ea-983c-37b3-822d-286962b0d26a",
5     "externalId": "icon#154823968",
6     "orderDate": "2024-12-02T14:41+01:00",
7     "countryCode": "SE",
8     "startDate": "2024-12-02",
9     "endDate": "2025-12-02",
10    "urbaStatus": "partially_ready",
11    "channelDelivery": "Urba Front Office"
12  },
13  {
14    "id": "02dfbf41-f88c-4d5d-98de-9d706ee168dd",
15    "externalId": "icon#154848838",
16    "orderDate": "2024-11-28T16:02:53+01:00",
17    "countryCode": "DE",
18    "startDate": "2024-11-28",
19    "endDate": "2025-11-28",
20    "customerReference": "IG URBA KH",
21    "urbaStatus": "ready",
```

