

Analytics 2.0 API real-time reports

The Analytics 2.0 real-time report API endpoint allows you to access real-time data programmatically through Adobe Developer. The real-time data reported is less than two minutes latent and auto-updates on a minute-by-minute basis. See the [Real-time reporting overview](#) for more information.

The endpoint described in this guide is routed through analytics.adobe.io. To use it, you will need to first create a client with access to the Adobe Analytics Reporting API. For more information, refer to [Getting started with the Analytics API](#).

Adobe may add optional request and response members (name/value pairs) to existing API objects at any time and without notice or changes in versioning. Adobe recommends that you refer to the API documentation of any third-party tool you integrate with our APIs so that such additions are ignored in processing if not understood. If implemented properly, such additions are non-breaking changes for your implementation. Adobe will not remove parameters or add required parameters without first providing standard notification through release notes.



This guide includes the following sections:

- [Real-time report API endpoint](#)
- [Real-time reports with relative date ranges](#)
- [Real-time reports with an additional dimension](#)
- [Real-time breakdown reports](#)
- [Limiting the number of dimension return values per period](#)
- [Status codes](#)
- [Trouble-shooting](#)
- [More information](#)

Real-time report API endpoint

Use the following endpoint to generate a real-time report for the data requested in a POST body.

POST

```
https://analytics.adobe.io/api/{GLOBAL_COMPANY_ID}/reports/realtime
```

You can find your global company ID by using the [Discovery API](#).



Request and response examples

Click the **Request** tab in the following example to see a cURL request for this endpoint. Click the **Response** tab to see a successful JSON response for the request.

Request Response

Copy

```
1 curl -X POST "https://analytics.adobe.io/api/{GLOBAL_COMPANY_ID}/reports/realtime"
2 -H "accept: application/json" \
3 -H "x-api-key: {CLIENT_ID}" \
4 -H "Authorization: Bearer {ACCESS_TOKEN}" \
5 -d '{
6   "rsid": "exemplarsid",
```

```
7  "globalFilters": [
8    {
9      "type": "dateRange",
10     "dateRange": "YYYY-MM-24T09:00:00/YYYY-MM-24T09:30:00"
11   }
12 ],
13 "metricContainer": {
14   "metrics": [
15     {
16       "columnId": "0",
17       "id": "metrics/occurrences"
18     }
19   ]
20 },
21 "dimensions": [
22   {
23     "id": "variables/daterangeminute",
24     "dimensionColumnId": "0"
25   }
26 ],
27 "settings": {
28   "realTimeMinuteGranularity": 10,
29   "limit": 20
30 }
31 }'
```

Request example details

The above example creates a real-time report request for the following:

- To show data for the dimension `daterangeminute` and the metric `occurrences` for the `rsid` `examplesid`.

To show data every 20 minutes starting from `YYYY-MM-24T09:00:00` to `YYYY-MM-24T09:30:00`, or on the same day--the 24th between the time from `09:00` to `09:30`. Note: the start date cannot be earlier than 20 hours from the time the request is made, according to the time zone specified for the report suite.

- To show data at a granularity of `10` minutes, as specified in the value of `realTimeMinuteGranularity`. The default granularity is one minute.

Request parameters

The POST real-time reports endpoint includes the following request parameters:

| PARAMETER | REQ/OPT | TYPE | DESCRIPTION |
|----------------------------|----------|-----------|--|
| <code>rsid</code> | required | string | report suite ID |
| <code>locale</code> | optional | string | The specified language |
| <code>globalFilters</code> | optional | container | Contains the <code>type</code> and <code>dateRange</code> parameters |
| <code>type</code> | optional | dateRange | The type of filter to be applied |
| <code>dateRange</code> | optional | string | The start and end dates for the report. The format is <code>YYYY-DD-DDT00:00:00/YYYY-MM-DDT00:00:00</code> and is based on the timezone of the <code>rsid</code> . |

| PARAMETER | REQ/OPT | TYPE | DESCRIPTION |
|----------------------------------|----------|-----------|--|
| <code>metricContainer</code> | required | container | Contains the <code>metrics</code> container. One metric is required for each real-time report. |
| <code>columnId</code> | required | string | The column ID. One metric column ID is required for each real-time report. |
| <code>id</code> | required | string | The metric or dimension ID. One metric ID is required for each real-time report. |
| <code>dimensions</code> | required | container | Contains the <code>id</code> and <code>dimensionColumnId</code> of the dimensions to be included in the report. For real-time reports, the <code>variables/dateRange</code> is required. |
| <code>dimensionColumnId</code> | required | string | The dimension column ID |
| <code>settings</code> | optional | container | Contains the <code>settings</code> object members for the specified real-time report, including <code>realTimeMinuteGroup</code> |
| <code>realTimeMinuteGroup</code> | optional | | The number of minutes between the reporting of the specified data |
| <code>limit</code> | optional | \$int32 | The maximum number of items to show per page |

Response example details

The above JSON response example shows the following details:

- The number of `occurrences` from 9:00 to 9:10 on the 24th is `2183`.
- The number of `occurrences` from 9:10 to 9:20 on the 24th is `2256`.
- The number of `occurrences` from 9:20 to just before 9:30 on the 24th is `2034`.
- The total number of `occurrences` from 9:00 to 9:30 on the 24th is `6473`.

Response parameters

The POST real-time reports endpoint includes the following response parameters:

| PARAMETER | TYPE | DESCRIPTION |
|-------------------------------|-----------|--|
| <code>totalPages</code> | \$int32 | The total pages for the report |
| <code>firstPage</code> | boolean | Whether the response includes the first page of report |
| <code>lastPage</code> | boolean | Whether the response includes the last page of report |
| <code>numberOfElements</code> | \$int32 | The total number of elements included in the report |
| <code>rows</code> | container | Contains the <code>row</code> parameters <code>itemIds</code> , <code>values</code> , <code>data</code> , <code>value</code> , and <code>itemId</code> . |

| PARAMETER | TYPE | DESCRIPTION |
|-------------|-----------|---|
| itemIds | string | Processing numbers used to associate time values |
| values | string | The values--or in this case the tim--associated with the data |
| data | Sdouble | The data--or in this case the occurrences--associated with the specified item and value |
| value | string | The values--or in this case the tim--associated with the data |
| itemId | string | Processing number used to associate time values |
| summaryData | container | Contains objects providing summary for report |
| totals | Sint32 | The total for all data results in the report |

Real-time reports with relative dates

The POST real-time reports endpoint also supports requests with relative date ranges. Many users find these values easier to specify for real-time querying. Relative date ranges differ from absolute date range values in how they are specified. Relative dates use the format from /to to specify the ranges. The following examples show the difference:

Absolute date range example:

```
"dateRange": "YYYY-10-01T00:00:00.000/YYYY-10-01T00:20:00.000", //Note: 20 minute range
```

Relative date range example:

```
"dateRangeRelative": "30 minutes ago/10 minutes ago" //Note: 20 minute range
```

Relative date range in JSON:

Copy

```
1 {
2   "rsid": "exemplersid.suiteall",
3   "globalFilters": [
4     {
5       "type": "dateRange",
6       "dateRangeRelative": "30 minutes ago/10 minutes ago"
7     }
8   ],
9 }
```

Possible values for from and to:

- minutes, for example: 20 minutes ago
- hours, for example: 1 hour ago
- now
- midnight
- noon
- hour time with AM/PM, for example: 3:45 PM or 3 PM
- yesterday hour time with AM/PM, for example: yesterday 12:30 PM or yesterday 12 PM

Real-time reports with an additional dimension

The following POST request example includes an additional dimension (`clickmaplinkbyregion`), along with the `daterangeminute` dimension shown in the first example above. The result is additional data values for the added dimension and combined dimension `occurrences`.

Request and response examples

Click the **Request** tab in the following example to see a cURL request for this endpoint. Click the **Response** tab to see a successful JSON response for the request.

Request Response

Copy

```
1  curl -X POST "https://analytics.adobe.io/api/{GLOBAL_COMPANY-ID}/reports/realtime
2  -H "accept: application/json" \
3  -H "x-api-key: {CLIENT_ID}" \
4  -H "Authorization: Bearer {ACCESS_TOKEN}" \
5  -d '{
6    "rsid": "examplesid",
7    "globalFilters": [
8      {
9        "type": "dateRange",
10       "dateRange": "YYYY-MM-25T09:00:00/YYYY-MM-25T09:30:00"
11      }
12    ],
13    "metricContainer": {
14      "metrics": [
15        {
16          "columnId": "0",
17          "id": "metrics/occurrences"
18        }
19      ]
20    },
21    "dimensions": [
22      {
23        "id": "variables/daterangeminute",
24        "dimensionColumnId": "0"
25      },
26      {
27        "id": "variables/clickmaplinkbyregion",
28        "dimensionColumnId": "1"
29      }
30    ],
31    "settings": {
32      "realTimeMinuteGranularity": 10,
33      "limit": 20
34    }
35  }
```

Request example details

The above example creates a real-time report request for the following:

- To show data for the dimensions `daterangeminute` and `clickmaplinkbyregion` according to the metric `occurrences` for the `rsid` `examplesid`.
- To show data over a 30-minute time period on from `YYYY-MM-25T09:00:00` to `YYYY-MM-25T09:30:00`, or on the same day--the 25th between the time from `09:00` to `09:30`. Note: the start date cannot be earlier than 20 hours from the time the request is made, according to the time zone specified for the report suite.
- To show data at a granularity of `10` minutes, as specified in the value of `realTimeMinuteGranularity`.

The request parameters shown in this example are described in the table above.

Response example details

The above JSON response example shows the following details:

- The number of clickmap `occurrences` for the `Save | BODY` region from 9:00 to 9:10 on the 25th is `53`.
- The number of clickmap `occurrences` for the `Discard | BODY` region from 9:00 to 9:10 on the 25th is `44`.
- The number of clickmap `occurrences` for the `Save | BODY` region from 9:10 to 9:20 on the 25th is `60`.
- The number of clickmap `occurrences` for the `Discard | BODY` region from 9:10 to 9:20 on the 25th is `45`.
- The number of clickmap `occurrences` for the `Save | BODY` region from 9:20 to 9:30 on the 25th is `214`.
- The number of clickmap `occurrences` for the `Discard | BODY` region from 9:20 to 9:30 on the 25th is `33`.
- The total number of clickmap `occurrences` from 9:00 to 9:30 on the 25th is `449`.

Real-time breakdown reports

Breakdown reports are useful when you want to see the cross-product of values from two different dimensions. When creating a real-time breakdown report, use the same endpoint as shown above. A real-time breakdown report includes the `variable/daterangeminute` dimension, a metric, an additional dimension, and a breakdown `itemId`. Within the payload of the request, use the `metricsFilters` parameter to specify the filter to be applied, as shown below:

Copy

```
1 "metricFilters":[
2   {
3     "id":"0",
4     "type":"breakdown",
5     "dimension":"variables/prop1",
6     "itemId":"2939795554"
7   }
8 ]
```

In this case, the `variables/prop1` dimension and its `itemId` are applied as a filter to produce the breakdown. To see an example of a breakdown request that specifies a different object within `metricFilters`, as well as general information on breakdown reports, see [Breakdown dimensions](#).

Real-time breakdown limitations

1. At a high level, real-time reports only support a single level of breakdown (i.e. one dimension broken down by another dimension). Realtime "report configuration" rows are set with one metric and 1-3 dimensions. Real-time breakdown reports (i.e. relating multiple dimensions in a single report) can only be run on dimensions that are configured as part of the same "report configuration" row in the data. Therefore, dimensions that exist in different "report configuration" rows cannot be executed together in a real-time report.
2. Only the first dimension in a report row is related or correlated to the other row dimensions. The second and third dimensions are not correlated to each other. As a result, any configuration that includes three dimensions can only show two possible breakdowns, as follows:
 - Valid: Dimension1 by Dimension2
 - Valid: Dimension1 by Dimension3
 - Not valid: Dimension2 by Dimension3

Note: Correlations work in both directions automatically. A single report on two dimensions together allows reporting breakdowns in either direction.

Limiting the number of dimension return values per period

In some instances, the number of dimension values returned per specified time period may be too high or excessive. To manage this, use the `realTimeValuePerPeriod` parameter in the `settings` object to specify the number of values you want returned. When using this parameter, you should specify a number of `10` (minimum) or higher or the setting will produce an error. The default is `10` values. For example, to have a certain dimension return only 12 values per time period (such as 30 minutes), the settings object would appear as follows:

Copy

```
1 "settings": {
2   "realTimeMinuteGranularity": 30,
3   "realTimeValuesPerPeriod": 12
4 }
```

Note: Any specified `limit` or `page` parameter settings will over-ride the `realTimeValuesPerPeriod` setting.

Status codes

Each API request returns an HTTP status code that reflects the result, as follows:

| HTTP CODE | MEANING | DESCRIPTION |
|-----------|------------------------|--|
| 200 | Success | The request is successful. |
| 400 | Bad Request | The request was improperly constructed, missing key information, and/or contained incorrect syntax. This error code could indicate a problem such as a missing required parameter or the supplied data did not pass validation. |
| 401 | Authentication failed | The request did not pass an authentication check. Your access token may be missing or invalid. Similarly, you may have attempted to access an object that requires administrator permissions. See the OAuth token errors section below for more details. |
| 403 | Forbidden | The resource was found, but you do not have the right credentials to view it. You might not have the required permissions to access or edit the resource for reasons not applicable to status code 401. |
| 404 | Not found | The requested resource could not be found on the server. The resource may have been deleted, or the requested path was entered incorrectly. |
| 500 | Internal server errors | This is a server-side error. If you are making many simultaneous calls, you may be reaching the API limit and need to filter your results. Wait for a moment before trying your request again, and contact your administrator if the problem persists. |

Troubleshooting

If you receive a 400 status code error response, make sure your requests comply with the following rules:

- Provide only valid metrics or dimensions that are [supported by real-time requests](#).
- Pass only one metric per request. Currently, the service returns 400 if more than one is passed.

- Always specify your first dimension as `variables/dateRangeMinute` since real time reports are always time bound and reported over minute granularity.
- Provide combinations of metrics and dimensions that qualify only as "overtime", "trended" or "breakdown" types.
- Supply search inputs for trended reports only.
- Specify the `realTimeMinuteGranularity` of at least `10` in the settings object.
- Do not pass `statistics`, `identityOverrides`, or `bulkExportSettings`. These are valid for normal reports but not for real-time.
- When using global filters, provide only a `dateRange` type entry.
- Do not include calculated metrics for real-time report requests.
- Do not include segment filtering for real-time report requests.
- Do not specify the start date of the report earlier than 20 hours ago (relative to report suite timezone).
- Do not specify the start date in the future.
- Request only 60 or fewer time periods per report.
- Always include a global date range in the format **StartDate/EndDate** (YYYY-MM-DDT00:00:00/YY-MM-DDT00:00:00). The start date cannot be earlier than the end date.
- Do not specify negative numbers for `page` parameters.

More information

For more information on real-time reports, see the following:

- [Adobe Analytics 2.0 API real-time report reference](#)
- [Adobe Analytics 2.0 API reporting guide](#)
- [Adobe Analytics API breakdown dimensions](#)
- [Adobe Analytics real-time reports configuration](#)
- [Adobe Analytics supported real-time metrics and dimensions](#)
- [Adobe Analytics real-time reporting overview](#)
- [Adobe Analytics Tech Notes: Real-time reports](#)

[← Reports](#)

[Breakdowns →](#)



Last updated 2024. 8. 7.

Was this helpful?

Yes

No