

Topic Data Pull

Retrieve the Bulk-Data Details for a Topic

Request

Request

Headers

Request Path

Parameters

Sample Request

Response

Response Body

Sample Response

Error Codes

Before Using PubMatic APIs...

First generate the API Token. For more information, see [Getting Started with PubMatic APIs](#).

Retrieve the Bulk-Data Details for a Topic

If your data size is very large, the report takes too long to process, or the report generation is timing out, use this API method. This allows you to register the topic (query pattern) and get the data using the topicId generated by the registration process, when the data has been prepared and can manage large data sets more efficiently.

This method is limited to a single topic per pull. If you need to pull all topics, see [Webhook Data Pull](#).

Request

URI	<code>http://{domainName}/topic/data/{topicId}/{day}</code>
HTTP Method	GET

Request Headers

Header Name	Type	Required	Description
Authorization	string	Yes	Send your generated access token for authentication in place of <code>{access_token}</code> for App/User. Format, "Bearer + Space + Token". Example: <i>"Bearer 9c7xxx3c"</i>
Accept-Encoding: gzip	string	Yes	Compresses data before network transfer. Client receives compressed data in bytes. Example: <i>154 megabytes file can be compressed to 17 megabytes</i>

Request Path Parameters

Parameter Name	Type	Required	Validations	Description
topicId	integer	Yes	Publishers can only query their own data	Unique topic ID
day	string	Yes		YYYY-MM-DD

Sample Request

Response

Response Header Name	Type	Description
Content-Type	application/json	Json response

Response Body

The following table lists parameters returned in API responses.

Response Body Parameter	Type	Description
alert	string	Alert message from server. In most cases, it will be null. Example: <pre>"alert": null</pre>
columns	string[]	Dimension and metrics columns. Please see the relevant reference section for a complete list of dimensions and metrics and their limitations. Example: <pre>"columns": ["dealMetaId", "revenue", "paidImpressions", "ecpm"],</pre> <p>dealMetaId is a special dimension in query for additional columns to be added on returned response. Example:</p> <p><i>Query on dealMetaId, revenue, paidImpressions, and ecpm, adds one more column, publisherdealId, to the response. The rows section will also include the new column value.</i></p> <pre>"columns": ["dealMetaId", "publisherdealId", "revenue", "paidImpressions", "ecpm"],</pre>

dataFreshness	object	<p>The date on which the settings of the dataset were last updated. Example:</p> <pre>"dataFreshness": { "dataFreshnessHour": "2015-03-12T04", "timeZone": "PST" },</pre>
displayValue	object	<p>The display value of the "ID." Example:</p> <pre>dealMetaId "13497" <i>has a description (name) of</i> "xxx"</pre> <pre>"displayValue": { "dealMetaId": { "13497": "xxx", "16083": "yyy",</pre>

<p>rows</p>	<p>array of array</p>	<p>The main table-like data structure, in which each column in the row corresponds to the columns definition in the same sequence. Example:</p> <p><i>The records below would be presented in columns, ["dealMetaId", "publisherDealId", "revenue", "paidImpressions", "ecpm"]</i></p> <pre>"rows": [["16651", "PM- XXX339", 83365.114 596, 10126407. 0, 8.232448], </pre> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <p>When using the dealMetaId dimension, you must associate the publisherDealId.</p> </div> <p>Example:</p> <pre>"columns": ["dealMetaId", "publisherDealId", "revenue"], "rows": [["10534", "PM- SUU63339", 0], </pre>
--------------------	-----------------------	---

Sample Response

JSON

```
{
  "alert": null,
  "columns": [
    "dealMetaId",
    "revenue",
    "paidImpressions",
    "ecpm"
  ],
  "currency": "USD",

```

```

    "dataFreshness": {
      "dataFreshnessHour": "2015-03-12T04",
      "timeZone": "PST"
    },
    "displayValue": {
      "dealMetaId": {
        "13497":
"Criteo_Criteo_ALL_ROS_ALL_0708_3.50",
        "16083": "Target_RON_4.
00_MediaMath",
        "16091":
"Target_Women'sLifestyle_7.00_MediaMath",
        "16651": "Netflix_Entertainment_8.
00_DBM",
        "17698": "AT&T_RON_6.
50_MEC_AppNexus",
        "3504": "P&G_RON_6.
50_AudienceScience",
        "3740": "MaxPoint
Interactive_Maxpoint
Interactive_ALL_FLNEWS_ALL_6.00",
        "3743": "MaxPoint
Interactive_Maxpoint
Interactive_ALL_FLTODAY_ALL_6.00",
        "6189": "Invite_Geico_RON_ALL_2.
00",
        "8763": "MaxPoint
Interactive_Maxpoint Interactive_ALL_RON_ALL_2.
00"
      }
    },
    "rows": [
      [
        "16651",
        83365.114596,
        10126407,
        8.232448
      ],
      [
        "8763",
        77102.258791,
        38541211,
        2.000515
      ],
      [
        "3504",
        73710.9945,
        11340153,
        6.5
      ],
      [
        "3740",
        60776.184,

```

```

        10129364,
        6
    ],
    [
        "17698",
        54480.049734
        8096994,
        6.728429
    ],
    [
        "13497",
        45204.601177,
        11099020,
        4.072846
    ],
    [
        "6189",
        34021.467339,
        16561790,
        2.054214
    ],
    [
        "16091",
        30320.486,
        4331498,
        7
    ],
    [
        "3743",
        20962.92,
        3493820,
        6
    ],
    [
        "16083",
        10852.705205,
        2364376,
        4.590093
    ]
    ]
}

```

Error Codes

Error Code	Error Description
500	Internal Server Error
400	Bad Request
401	Unauthorized
403	Access Denied

404	Data Not Found
503	Service Not Available or too many requests

[« API Data Pull](#) [Webhook Data Pull](#) »

[Top](#)