Advanced Guide: How to get channel insights from Measure API V2

The goal of this article is to provide an example on how to retrieve channel insights consistently using the measure public api v2.

Prerequisites

1. Get an api key from: Settings-> Integrations & API -> API keys.
2. Know the brandwatch global request rate limits. Check them here.
   The AppQuota and CompanyQuota are renewed every month on the 11-th day.
3. Read the documentation - section Measure API V2.

Get channel insights

Goal: Take all metrics for all instagram channels and write them to a file iteratively.

Note: It is recommended to work with only one network at a time. If you need insights for more than one network take them network by network

1. GET all instagram channel uuids

```
curl 'https://api.falcon.io/channels?apikey={{apikey}}'
```

Cycle through all the items and get all channels that have *status="online"*, *network="instagram"* fields and take their **uuids**. If your company has more than 100 channels increase the limit by adding &limit={numChannels}.

Example Result:

```
[119f0444be58424aba702d725e4cbd77, f8f86293c996462dbb0dd0ae5a91e9bd,
56e6326493cc45b8b5662ed815f8331b, ca1a7be7d48045cebfdcde60c0f82cee,
e89c38ae4af947a68197185e8b538b8c, b80c7f331dba4d1d82bfa667439507fb,
cbbf50f37a734acc8e4860b41732d2f0, 68e0cc9aa9274397b39092f5c5ec95d0,
8c9ea00a1bb840b585f9c81546662d67, b5e5a59b1c794381833a4d01dbad4647,
0e0dd4a213834e28bffc6386e2156fe9, f777cceb81ef4ff4ab82e4fe4369ce50,
9cf7b0372aaf423dbf9bf215cf96556f, c4a853bcabdf47d78f2e945937248ada,
71117369a42046e7b7fc793065d3b0ba, a070d425ffdb4eba95143d1ed33fbd7c]
```

2. GET all instagram metricIds

```
curl
'https://api.falcon.io/measure/v2/metrics/channel?apikey={{apikey}}&networks=instagram'
```

Cycle through all the items and take their ids, optionally take also the breakdown ids.

Example Result:

```
[channel/fans/day, channel/stories/day, channel/incoming_private_messages/day,
channel/phone_call_clicks/day, channel/story_replies/day, channel/directions_clicks/day,
channel/impressions/day, channel/reach/day, channel/text_message_clicks/day,
channel/cta_clicks/day, channel/story_mentions/day, channel/net_fans/day,
channel/frequency/day, channel/website_clicks/day, channel/direct_messages/day,
channel/fans_increase/day, channel/views/day, channel/fans_by_gender_age/day,
channel/fans_by_country/day, channel/fans_online_by_hour/total/day,
channel/fans_online_by_hour/0/day, channel/fans_online_by_hour/1/day,
channel/fans_online_by_hour/10/day, channel/fans_online_by_hour/11/day,
channel/fans_online_by_hour/12/day, channel/fans_online_by_hour/13/day,
channel/fans_online_by_hour/14/day, channel/fans_online_by_hour/15/day,
channel/fans_online_by_hour/16/day, channel/fans_online_by_hour/17/day,
channel/fans_online_by_hour/18/day, channel/fans_online_by_hour/19/day,
channel/fans_online_by_hour/2/day, channel/fans_online_by_hour/20/day,
channel/fans_online_by_hour/21/day, channel/fans_online_by_hour/22/day,
channel/fans_online_by_hour/23/day, channel/fans_online_by_hour/3/day,
channel/fans_online_by_hour/4/day, channel/fans_online_by_hour/5/day,
channel/fans_online_by_hour/6/day, channel/fans_online_by_hour/7/day,
channel/fans_online_by_hour/8/day, channel/fans_online_by_hour/9/day]
```

3. Request channel insights

Keep the Measure API V2 request limits in mind:

Note: The request limits in the API docs are the most up to date limits. The ones below are just example limits.

```
METRICS_LIMIT = 20;
CHANNELS_LIMIT = 15;
UNTIL - SINCE < 3 months range
```

Note: Maximum range between since and until is 3 months but you can request data up to two years back.

We have 16 channel uuids and 44 metricIds.
Partition the channel uuids array into multiple arrays with max CHANNELS_LIMIT elements.

Example Result:

```
[

[f8f86293c996462dbb0dd0ae5a91e9bd, 119f0444be58424aba702d725e4cbd77,
56e6326493cc45b8b5662ed815f8331b, ca1a7be7d48045cebfdcde60c0f82cee,
e89c38ae4af947a68197185e8b538b8c, b80c7f331dba4d1d82bfa667439507fb,
cbbf50f37a734acc8e4860b41732d2f0, 68e0cc9aa9274397b39092f5c5ec95d0,
8c9ea00a1bb840b585f9c81546662d67, b5e5a59b1c794381833a4d01dbad4647,
0e0dd4a213834e28bffc6386e2156fe9, f777cceb81ef4ff4ab82e4fe4369ce50,
9cf7b0372aaf423dbf9bf215cf96556f, c4a853bcabdf47d78f2e945937248ada,
71117369a42046e7b7fc793065d3b0ba],

[a070d425ffdb4eba95143d1ed33fbd7c],
]
```

Partition the metricIds array into multiple arrays with max METRICS_LIMIT elements.

Example Result:

```
[
[channel/fans/day, channel/stories/day, channel/incoming_private_messages/day,
channel/phone_call_clicks/day, channel/story_replies/day, channel/directions_clicks/day,
channel/impressions/day, channel/reach/day, channel/text_message_clicks/day,
channel/cta_clicks/day, channel/story_mentions/day, channel/net_fans/day,
channel/frequency/day, channel/website_clicks/day, channel/direct_messages/day,
channel/fans_increase/day, channel/views/day, channel/fans_by_gender_age/day,
channel/fans_by_country/day, channel/fans_online_by_hour/total/day],

[channel/fans_online_by_hour/0/day, channel/fans_online_by_hour/1/day,
channel/fans_online_by_hour/10/day, channel/fans_online_by_hour/11/day,
channel/fans_online_by_hour/12/day, channel/fans_online_by_hour/13/day,
channel/fans_online_by_hour/14/day, channel/fans_online_by_hour/15/day,
channel/fans_online_by_hour/16/day, channel/fans_online_by_hour/17/day,
channel/fans_online_by_hour/18/day, channel/fans_online_by_hour/19/day,
channel/fans_online_by_hour/2/day, channel/fans_online_by_hour/20/day,
channel/fans_online_by_hour/21/day, channel/fans_online_by_hour/22/day,
channel/fans_online_by_hour/23/day, channel/fans_online_by_hour/3/day,
channel/fans_online_by_hour/4/day, channel/fans_online_by_hour/5/day],
[channel/fans_online_by_hour/6/day, channel/fans_online_by_hour/7/day,
channel/fans_online_by_hour/8/day, channel/fans_online_by_hour/9/day]

]
```

We have two arrays. The channel uuids array has 2 subarrays and the metric ids array has 3 subarrays.

Create requests with every combination between the two arrays. This makes a total of six (2*3) requests.

Create requests like:

```
curl --location --request POST
'https://api.falcon.io/measure/v2/insights/channel?apikey={{apikey}}' --header 'Content-Type:
application/json' --data-raw '{
  "since": "2022-10-01",
  "until": "2022-10-05",
  "metricIds": {{metricIdsSubarray}},
  "channelIds": {{channelIdsSubarray}]
}'
```

Execute every request **synchronously one by one and wait 30 seconds between each request**, otherwise you may hit the request limits quickly.

Note: The 30 second wait is very conservative and makes sure the limits will not be hit whatever the use case. For the majority of use cases you can try to decrease it to 10 seconds or less. If you are ok to let the code run for more time and gradually load the data it is a safe bet to leave it to 30 seconds. That way you can leave the code running for days and backfill all your data from two years back and be sure you will not hit any limits.

For each executed request do:

Example request:

```
curl --location --request POST
'https://api.falcon.io/measure/v2/insights/channel?apikey={{apikey}}' --header 'Content-Type:
application/json' --data-raw '{
  "since": "2022-10-01",
  "until": "2022-10-05",
  "metricIds":
["channel/fans/day","channel/stories/day","channel/incoming_private_messages/day","channel
/phone_call_clicks/day","channel/story_replies/day","channel/directions_clicks/day","channel/i
mpressions/day","channel/reach/day","channel/text_message_clicks/day","channel/cta_clicks/
day","channel/story_mentions/day","channel/net_fans/day","channel/frequency/day","channel/
website_clicks/day","channel/direct_messages/day","channel/fans_increase/day","channel/vie
ws/day","channel/fans_by_gender_age/day","channel/fans_by_country/day","channel/fans_onli
ne_by_hour/total/day"],
  "channelIds":
["119f0444be58424aba702d725e4cbd77","f8f86293c996462dbb0dd0ae5a91e9bd","56e63264
93cc45b8b5662ed815f8331b","ca1a7be7d48045cebfdcde60c0f82cee","e89c38ae4af947a681
97185e8b538b8c","b80c7f331dba4d1d82bfa667439507fb","cbbf50f37a734acc8e4860b41732
d2f0","68e0cc9aa9274397b39092f5c5ec95d0","8c9ea00a1bb840b585f9c81546662d67","b5e
5a59b1c794381833a4d01dbad4647","0e0dd4a213834e28bffc6386e2156fe9","f777cceb81ef4
ff4ab82e4fe4369ce50","9cf7b0372aaf423dbf9bf215cf96556f","c4a853bcabdf47d78f2e94593
7248ada","71117369a42046e7b7fc793065d3b0ba"]
}'
```

Example Response:

```
{"insightsRequestId":"fc795a92-3c4c-44e0-885e-58f3c861cea0"}
```

Take the insightsRequestId **wait 10 seconds** and call

```
curl
'https://api.falcon.io/measure/v2/insights/fc795a92-3c4c-44e0-885e-58f3c861cea0?apikey={
{apikey}}'
```

Example Response:

```
{"status":"IN_PROGRESS"}
```

Status is IN_PROGRESS. Wait 5 seconds and check again.

```
curl
'https://api.falcon.io/measure/v2/insights/fc795a92-3c4c-44e0-885e-58f3c861cea0?apikey={
{apikey}}'
```

Note: If you see that you are regularly getting status IN_PROGRESS you may increase the wait time so you don't waste requests.

Example Response:

```
{
  "status": "READY",
  "data": {
    "insights": {"truncated for clarity"},
    "paging":{"nextPage":2}
  }
}
```

Read data.insights and store it.

If data.paging is not empty, then get data.paga.nextPage value and create a request to take the second page.

```
curl
'https://api.falcon.io/measure/v2/insights/fc795a92-3c4c-44e0-885e-58f3c861cea0?apikey={
{apikey}}&page=2'
```

Example Response:

```
{
  "status": "READY",
  "data": {
    "insights": {"truncated for clarity"},
    "paging":{"nextPage":3}
  }
}
```

Read data.insights and store it. Cycle through the pages until paging is empty.