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

The goal of this article is to provide an example on how to retrieve content 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 content insights

Goal: Take all content metrics for all facebook 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

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

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

Example Result:

```
[2251218615161967, 990064157851607, 536706626341028, 917291041741398,
101752822190239, 1669870896627278, 160595437616313, 107643346587002,
1781406488740652, 1095848597166952, 1897198880504263, 103235357690140,
1555238904713459, 635773539826876, 546650289020372, 693094307376969,
1769597826637807, 288721971540444, 351363318280238, 1592747851034711,
256080221451513, 1537723129793505, 507739059561724, 112169173973535,
189352594565351, 1469527876392921, 793022114202977, 802902563178983,
214061359369473, 103023681418539, 106141428896964, 108274572906948,
833871433343114, 691088441026650, 286453231765549, 100120228983145,
167206696767149, 510194999066526, 357050837716478, 329112521232092,
129797780413345, 883874651662050, 837473206610340, 108797198299090,
827344467315589, 125259194181035, 111825334631883, 1798690153684356,
1841285236189816, 2222109234724159, 1585464188351780, 350020615122850,
580802772014902, 418091191886049, 130511219203673, 157235471412609,
107992241839546, 942922802485311, 638175139572553, 102015409307512,
623115354801956, 114901099887654, 102375882615967, 100772054849803,
1654366434879540, 102236891148307, 957280054365105, 607693366063932,
111116254046939, 642777585839249, 109416885196165, 1730258937251354,
1834729560077207, 102065013546415, 109358685042722, 772866179504858,
104538374642095, 110302080422346, 1413973548644575, 330347807685270,
576845412647458, 333788233471842, 1111618002261533, 1665198663733258,
174463802679, 1622130091241781, 571524723001377, 316686608885869,
1525265274158370, 314600498625052, 1907970692757428, 142956609068298]
```

2. GET all facebook metricIds

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

Cycle through all the items and take their ids.

Example Result:

```
[content/engagement_rate_reach/lifetime, content/likes/lifetime,
content/impressions_paid/lifetime, content/engagement_rate_weighted/lifetime,
content/video_views_paid_complete/lifetime, content/video_views_autoplayed_30s/lifetime,
content/negative_feedback/lifetime, content/video_viewers_paid_complete/lifetime,
content/spam_rate/lifetime, content/ctr_link/lifetime, content/interactions/lifetime,
content/video_viewers_30s/lifetime, content/reactions_haha/lifetime,
content/reach_paid/lifetime, content/link_clicks/lifetime, content/clicks/lifetime,
content/impressions_nonviral/lifetime, content/video_views_clicked_to_play/lifetime,
content/video_views_clicked_to_play_30s/lifetime, content/photo_view_clicks/lifetime,
content/video_frequency/lifetime, content/comments_total/lifetime,
content/shares_total/lifetime, content/video_views_sound_on_10s/lifetime,
content/video_length/lifetime, content/reactions_like/lifetime,
content/report_spam_clicks/lifetime, content/ctr/lifetime, content/engagements/lifetime,
content/video_views_paid_30s/lifetime, content/video_viewers_organic/lifetime,
content/reach_viral/lifetime, content/hide_clicks/lifetime,
content/video_view_time_paid/lifetime, content/impressions/lifetime,
content/video_view_time_owned/lifetime, content/engagement_rate/lifetime,
content/frequency/lifetime, content/other_clicks/lifetime, content/negative_users/lifetime,
content/video_views_paid_10s/lifetime, content/engagements_weighted/lifetime,
content/video_views_sound_on/lifetime, content/video_view_time_organic/lifetime,
content/video_viewers/lifetime, content/engaged_fans/lifetime,
content/video_views_clicked_to_play_10s/lifetime, content/video_views_autoplayed/lifetime,
content/reach_organic/lifetime, content/video_play_clicks/lifetime,
content/video_retention_rate_10s/lifetime, content/video_viewers_organic_complete/lifetime,
content/video_views_organic_30s/lifetime, content/engaged_users/lifetime,
content/interaction_rate/lifetime, content/video_view_time_shared/lifetime,
content/engaged_users_rate/lifetime, content/video_views/lifetime,
content/video_views_autoplayed_10s/lifetime, content/video_avg_time_watched/lifetime,
content/video_views_organic_10s/lifetime, content/photo_view_rate/lifetime,
content/video_viewers_10s/lifetime, content/unlike_clicks/lifetime,
content/reactions_love/lifetime, content/impressions_viral/lifetime,
content/comments/lifetime, content/reactions_sorry/lifetime,
content/reach_nonviral/lifetime, content/video_view_time/lifetime,
content/video_view_rate/lifetime, content/reactions_total/lifetime,
content/impressions_fan/lifetime, content/engagement_rate_link/lifetime,
content/reactions_wow/lifetime, content/video_views_10s/lifetime, content/reach/lifetime,
content/reactions_anger/lifetime, content/interaction_rate_reach/lifetime,
content/video_views_organic/lifetime, content/impressions_organic/lifetime,
content/viral_amplification/lifetime, content/reactions/lifetime,
content/video_completion_rate/lifetime, content/video_views_organic_complete/lifetime,
content/video_viewers_paid/lifetime, content/reach_fan/lifetime,
content/video_views_paid/lifetime, content/shares/lifetime]
```

3. Request content ids

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;
CONTENTS_LIMIT = 300;
UNTIL - SINCE < 3 months
```

We have 92 channel ids and 89 metric ids.

Partition the channel ids array into multiple arrays with max CHANNELS_LIMIT elements.

Example Result:

```
[

[2251218615161967, 990064157851607, 536706626341028, 917291041741398,
101752822190239, 1669870896627278, 160595437616313, 107643346587002,
1781406488740652, 1095848597166952, 1897198880504263, 103235357690140,
1555238904713459, 635773539826876, 546650289020372], [693094307376969,
1769597826637807, 288721971540444, 351363318280238, 1592747851034711,
256080221451513, 1537723129793505, 507739059561724, 112169173973535,
189352594565351, 1469527876392921, 793022114202977, 802902563178983,
214061359369473, 103023681418539],

[106141428896964, 108274572906948, 833871433343114, 691088441026650,
286453231765549, 100120228983145, 167206696767149, 510194999066526,
357050837716478, 329112521232092, 129797780413345, 883874651662050,
837473206610340, 108797198299090, 827344467315589],

[125259194181035, 111825334631883, 1798690153684356, 1841285236189816,
2222109234724159, 1585464188351780, 350020615122850, 580802772014902,
418091191886049, 130511219203673, 157235471412609, 107992241839546,
942922802485311, 638175139572553, 102015409307512],
[623115354801956, 114901099887654, 102375882615967, 100772054849803,
1654366434879540, 102236891148307, 957280054365105, 607693366063932,
111116254046939, 642777585839249, 109416885196165, 1730258937251354,
1834729560077207, 102065013546415, 109358685042722],

[772866179504858, 104538374642095, 110302080422346, 1413973548644575,
330347807685270, 576845412647458, 333788233471842, 1111618002261533,
1665198663733258, 174463802679, 1622130091241781, 571524723001377,
316686608885869, 1525265274158370, 314600498625052],

[1907970692757428, 142956609068298]]
```

Partition the metric ids array into multiple arrays with max METRICS_LIMIT elements.

Example Result:

```
[

[content/engagement_rate_reach/lifetime, content/likes/lifetime,
content/impressions_paid/lifetime, content/engagement_rate_weighted/lifetime,
content/video_views_paid_complete/lifetime, content/video_views_autoplayed_30s/lifetime,
content/negative_feedback/lifetime, content/video_viewers_paid_complete/lifetime,
content/spam_rate/lifetime, content/ctr_link/lifetime, content/interactions/lifetime,
content/video_viewers_30s/lifetime, content/reactions_haha/lifetime,
content/reach_paid/lifetime, content/link_clicks/lifetime, content/clicks/lifetime,
content/impressions_nonviral/lifetime, content/video_views_clicked_to_play/lifetime,
content/video_views_clicked_to_play_30s/lifetime, content/photo_view_clicks/lifetime],

[content/video_frequency/lifetime, content/comments_total/lifetime,
content/shares_total/lifetime, content/video_views_sound_on_10s/lifetime,
content/video_length/lifetime, content/reactions_like/lifetime,
content/report_spam_clicks/lifetime, content/ctr/lifetime, content/engagements/lifetime,
content/video_views_paid_30s/lifetime, content/video_viewers_organic/lifetime,
content/reach_viral/lifetime, content/hide_clicks/lifetime,
content/video_view_time_paid/lifetime, content/impressions/lifetime,
content/video_view_time_owned/lifetime, content/engagement_rate/lifetime,
content/frequency/lifetime, content/other_clicks/lifetime, content/negative_users/lifetime],

[content/video_views_paid_10s/lifetime, content/engagements_weighted/lifetime,
content/video_views_sound_on/lifetime, content/video_view_time_organic/lifetime,
content/video_viewers/lifetime, content/engaged_fans/lifetime,
content/video_views_clicked_to_play_10s/lifetime, content/video_views_autoplayed/lifetime,
content/reach_organic/lifetime, content/video_play_clicks/lifetime,
content/video_retention_rate_10s/lifetime, content/video_viewers_organic_complete/lifetime,
content/video_views_organic_30s/lifetime, content/engaged_users/lifetime,
content/interaction_rate/lifetime, content/video_view_time_shared/lifetime,
content/engaged_users_rate/lifetime, content/video_views/lifetime,
content/video_views_autoplayed_10s/lifetime, content/video_avg_time_watched/lifetime],

[content/video_views_organic_10s/lifetime, content/photo_view_rate/lifetime,
content/video_viewers_10s/lifetime, content/unlike_clicks/lifetime,
content/reactions_love/lifetime, content/impressions_viral/lifetime,
content/comments/lifetime, content/reactions_sorry/lifetime,
content/reach_nonviral/lifetime, content/video_view_time/lifetime,
content/video_view_rate/lifetime, content/reactions_total/lifetime,
content/impressions_fan/lifetime, content/engagement_rate_link/lifetime,
content/reactions_wow/lifetime, content/video_views_10s/lifetime, content/reach/lifetime,
content/reactions_anger/lifetime, content/interaction_rate_reach/lifetime,
content/video_views_organic/lifetime],

[content/impressions_organic/lifetime, content/viral_amplification/lifetime,
content/reactions/lifetime, content/video_completion_rate/lifetime,
content/video_views_organic_complete/lifetime, content/video_viewers_paid/lifetime,
content/reach_fan/lifetime, content/video_views_paid/lifetime, content/shares/lifetime]

]
```

For each subarray of channel ids do:

1. Go through its elements one by one and invoke requests like.

```
curl
'https://api.falcon.io/publish/items?since={{since}}&until={{until}}&limit=100&channels={{cha
nnelId}}&apikey={{apikey}}&statuses=published'
```

Execute every request synchronously one by one and wait 10 seconds between each so you don't hit the limits.

Note: If you need more than 100 hundred posts for the period you can use paging and cycle through the pages to get all the posts. But don't take more than `CONTENTS_LIMIT` posts. You *should adjust the period (since and until values), so you are sure you don't have more than 300 posts per channel for this period.*

2. Store the items[*id] and items[channels]* in an object. Let's call the class of this object *Content*. At the end you should have content objects for all contentIds of all channels in the channel ids subarray.

Note: items[channels] contain the channel uuids that you need to request content insights.

Example Result:

```
[Content{id='c187e0368e26536c81be98818eb04f9a',channels=[b302fa0b5ad646dfba64183b2e34b001]},

Content{id='c8b01f3edcf859dfbe9347fae605cda8',channels=[b302fa0b5ad646dfba64183b2e34b001]},

Content{id='e89d4acc0a835242a6273c7c42ed16e3',channels=[b302fa0b5ad646dfba64183b2e34b001]},

Content{id='7317d0e7bdea548a9ff79e113499cf60',channels=[b302fa0b5ad646dfba64183b2e34b001]},

Content{id='357758fcfd7d50bbb6b5951eddcf254f',channels=[b302fa0b5ad646dfba64183b2e34b001]},

Content{id='16ff2ea4496c5857b59044549d2fd890',channels=[4fcc1ebaa4de4397aa204643d2ece33a]},

Content{id='91df62b3d8275c22b2ba50e562f11013',channels=[4fcc1ebaa4de4397aa204643d2ece33a]},

Content{id='b5cce81aa320419dbc1967802d7d4efb',channels=[4fcc1ebaa4de4397aa204643d2ece33a]},

Content{id='1f61a278629d5911a74a78b4d719b476',channels=[06c1fce7710644929604edbb547ff12a]},

Content{id='34fe772614965adcabe730e4ed5831a4',channels=[06c1fce7710644929604edbb547ff12a]},
Content{id='4c0a0175091e502890d745dca7edc2db',channels=[06c1fce7710644929604edbb
547ff12a]},

Content{id='6915e372f0ac5a0e8d92dc22ec74220c',channels=[06c1fce7710644929604edbb
547ff12a]},

Content{id='d8bd95852bde5227a92953adbd27bf51',channels=[06c1fce7710644929604edbb
547ff12a]},

Content{id='ea63082771745aca98e6f27729147381',channels=[06c1fce7710644929604edbb
547ff12a]},

Content{id='ffff629c7922552591c44d8cb74d057e',channels=[06c1fce7710644929604edbb5
47ff12a]},

Content{id='94fd21b89bce5ff0b7a645f9b399a783',channels=[06c1fce7710644929604edbb5
47ff12a]},

Content{id='3e9b0163651c5f74b5f91b07064e1422',channels=[06c1fce7710644929604edbb
547ff12a]},

Content{id='1a08b91f757e5c5bb9fcff497ab06196',channels=[06c1fce7710644929604edbb5
47ff12a]},

Content{id='9c7200883ead5868b49060b2e062183f',channels=[06c1fce7710644929604edbb
547ff12a]},

Content{id='2d3becd5dd825ca8a3ee7636456d5e57',channels=[06c1fce7710644929604edbb
547ff12a]},

Content{id='e6f41313276459d6af546d73de821015',channels=[bd6f0b79ead847ba9a0b20f2c
7fc35ba]},

Content{id='5d24f0b487915ddab730cc22839ee47a',channels=[962ba34287944b1f96757578
7119c4a4]},

Content{id='428a7f93f20858908cac439e919da02b',channels=[bfd3901d1bae4fbb86e4cff6b1
0b978e]},

Content{id='322fbd6af0ec5ccba64d0c1d5c5d429f',channels=[bfd3901d1bae4fbb86e4cff6b1
0b978e]}]
```

3. Group the content objects by channel uuid (the channel uuid is the field in the channels array of Content class) and call it “id”. Every channel id should have an associated array of content ids. Let’s call this structure *channelsArray*.

Example Result:

```
[{"id":"06c1fce7710644929604edbb547ff12a","contentIds":["1f61a278629d5911a7
4a78b4d719b476","34fe772614965adcabe730e4ed5831a4","4c0a0175091e502890d745d
ca7edc2db","6915e372f0ac5a0e8d92dc22ec74220c","d8bd95852bde5227a92953adbd27
bf51","ea63082771745aca98e6f27729147381","ffff629c7922552591c44d8cb74d057e"
,"94fd21b89bce5ff0b7a645f9b399a783","3e9b0163651c5f74b5f91b07064e1422","1a0
8b91f757e5c5bb9fcff497ab06196","9c7200883ead5868b49060b2e062183f","2d3becd5
dd825ca8a3ee7636456d5e57"]},{"id":"1ef191e639df488d9bc6a28c4b31a353","conte
ntIds":["426615bfdb89590aba9ad20a6aad7286"]},{"id":"962ba34287944b1f9675757
87119c4a4","contentIds":["5d24f0b487915ddab730cc22839ee47a"]},{"id":"4fcc1e
baa4de4397aa204643d2ece33a","contentIds":["16ff2ea4496c5857b59044549d2fd890
","91df62b3d8275c22b2ba50e562f11013","b5cce81aa320419dbc1967802d7d4efb"]},{
"id":"bfd3901d1bae4fbb86e4cff6b10b978e","contentIds":["428a7f93f20858908cac
439e919da02b","322fbd6af0ec5ccba64d0c1d5c5d429f"]},{"id":"bd6f0b79ead847ba9
a0b20f2c7fc35ba","contentIds":["e6f41313276459d6af546d73de821015"]},{"id":"
b302fa0b5ad646dfba64183b2e34b001","contentIds":["c187e0368e26536c81be98818e
b04f9a","c8b01f3edcf859dfbe9347fae605cda8","e89d4acc0a835242a6273c7c42ed16e
3","7317d0e7bdea548a9ff79e113499cf60","357758fcfd7d50bbb6b5951eddcf254f"]}]
```

5. Cycle through the partitioned metric ids array and for each element create a request like:

```
curl --location --request POST
'https://api.falcon.io/measure/v2/insights/content?apikey={{apikey}}'
--header 'Content-Type: application/json' --data-raw '{
  "since": {{since}},
  "until": {{until}},
  "metricIds": {{metricIdsSubarray}},
  "channels": {{channelsArray}}
}'
```

Example Request

```
curl --location --request POST
'https://api.falcon.io/measure/v2/insights/content?apikey={{apikey}}'
--header 'Content-Type: application/json' --data-raw '{
  "since": "2022-09-10T00:00:00.000Z",
  "until": "2022-10-10T00:00:00.000Z",
  "metricIds":
["content/engagement_rate_reach/lifetime","content/likes/lifetime","content
/impressions_paid/lifetime","content/engagement_rate_weighted/lifetime","co
ntent/video_views_paid_complete/lifetime","content/video_views_autoplayed_3
0s/lifetime","content/negative_feedback/lifetime","content/video_viewers_pa
id_complete/lifetime","content/spam_rate/lifetime","content/ctr_link/lifeti
me","content/interactions/lifetime","content/video_viewers_30s/lifetime","c
ontent/reactions_haha/lifetime","content/reach_paid/lifetime","content/link
_clicks/lifetime","content/clicks/lifetime","content/impressions_nonviral/l
ifetime","content/video_views_clicked_to_play/lifetime","content/video_view
s_clicked_to_play_30s/lifetime","content/photo_view_clicks/lifetime"],
  "channels":
[{"id":"06c1fce7710644929604edbb547ff12a","contentIds":["1f61a278629d5911a7
4a78b4d719b476","34fe772614965adcabe730e4ed5831a4","4c0a0175091e502890d745d
ca7edc2db","6915e372f0ac5a0e8d92dc22ec74220c","d8bd95852bde5227a92953adbd27
bf51","ea63082771745aca98e6f27729147381","ffff629c7922552591c44d8cb74d057e"
,"94fd21b89bce5ff0b7a645f9b399a783","3e9b0163651c5f74b5f91b07064e1422","1a0
8b91f757e5c5bb9fcff497ab06196","9c7200883ead5868b49060b2e062183f","2d3becd5
dd825ca8a3ee7636456d5e57"]},{"id":"1ef191e639df488d9bc6a28c4b31a353","conte
ntIds":["426615bfdb89590aba9ad20a6aad7286"]},{"id":"962ba34287944b1f9675757
87119c4a4","contentIds":["5d24f0b487915ddab730cc22839ee47a"]},{"id":"4fcc1e
baa4de4397aa204643d2ece33a","contentIds":["16ff2ea4496c5857b59044549d2fd890
","91df62b3d8275c22b2ba50e562f11013","b5cce81aa320419dbc1967802d7d4efb"]},{
"id":"bfd3901d1bae4fbb86e4cff6b10b978e","contentIds":["428a7f93f20858908cac
439e919da02b","322fbd6af0ec5ccba64d0c1d5c5d429f"]},{"id":"bd6f0b79ead847ba9
a0b20f2c7fc35ba","contentIds":["e6f41313276459d6af546d73de821015"]},{"id":"
b302fa0b5ad646dfba64183b2e34b001","contentIds":["c187e0368e26536c81be98818e
b04f9a","c8b01f3edcf859dfbe9347fae605cda8","e89d4acc0a835242a6273c7c42ed16e
3","7317d0e7bdea548a9ff79e113499cf60","357758fcfd7d50bbb6b5951eddcf254f"]}]
}'
```

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

Note: There is a 30 seconds wait between each insights post request. This 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 for two years back and be sure you will not hit any limits.

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/04276421-8891-4ae6-8ddb-3ca917dc3349?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.