

Google Analytics 4 & Meltano



Outline

- I. Why are we migrating to GA4?
- II. GA4 vs GA3
- III. New ETL process
- IV. Dimension & Metrics
- V. Data structures

Why?

1. **Forced sunset:** in July 2023, Google is deprecating GA3
2. **Consolidate into MWAA:** GA3 orchestrated with Windows scheduler
3. **Data Lake:** centralize data in s3 (external table into Redshift)

II. GA3 vs GA4

GA3



1. **Custom code:** our current process was built from the ground up
2. **viewID:** A reporting view is the level in an Analytics account where you can access reports and analysis tools
3. **Universal analytics:** metrics are based on sessions and pageviews

GA4



1. **Repeatable framework:** replicable architecture that can be containerized and managed from a config
2. **propertyID:** a new **property** designed for the future of measurement:
 - Collects both website and app data to better understand the customer journey
 - Uses event-based data instead of session-based
 - Includes privacy controls such as cookieless measurement, and behavioral and conversion modeling
3. **Cannot capture “site speed”:** GA4 metrics track user activities as events

III. New ETL Process

GA3: Custom ETL

- Extract - [ga_helpers.py](#)
 - Dimensions/Metrics/View Id (--config)
 - Start date / Days ago (--start_date/--days_ago)
 - Service Account
- Load - [ga_helpers.py](#)
 - 1) Raw CSV files
 - 2) S3 Bucket
 - 3) [Redshift] Tables

GA4: Meltano

- Extractor - [tap-google-analytics-v4](#)
 - JSON reports (reports)
 - Property Id (property_id)
 - Start date (start_date)
 - Service Account (key_file_location)
- Loader - [target-s3-jsonl](#)
 - 1) Raw JSONL files [S3 Data Lake Bucket]
 - 2) [Redshift] External Tables
 - 3) [Redshift] Views

III. New ETL Process

GA3

- Custom ETL -
 - Extract & Load - [ga_helpers.py](#)

```
def ga_query_to_df(view_id, start_date, end_date, dimensions:list, metrics:list, segments:list, filters:str=None):
    print(f'Reached ga_query_to_df. Start: {start_date}, end: {end_date}')

    analytics = initialize_analyticsreporting()

    view_id = view_id_dict[view]

    ga_input_tab = get_ga_data(view_id, start_date, end_date, analytics, dimensions, metrics, segments, None, filters)

    # pagination...

    for report in ga_input_tab.get('reports', []):
        # set column headers
        next_page_token = report.get('nextPageToken', None)
        if next_page_token is not None:
            print(f'Next page: ' + next_page_token)
            traffic_response = get_ga_data(view_id, start_date, end_date, analytics, dimensions, metrics, segments, None, filters)
            ga_df = print_response(ga_input_tab)
            if next_page_token is not None:
                print(f'Next page: ' + next_page_token)
                while next_page_token is not None:
                    # get ga_data
                    re_response = get_ga_data(view_id, start_date, end_date, analytics, dimensions, metrics, segments, page=next_page_token,
                    filters=filters)

                    # converts segment re-response object to df
                    re_df = print_response(re_response)
                    ga_df = ga_df.append(re_df.reset_index(drop=True))
                    for report in re_response.get('reports', []):
                        # set column headers
                        next_page_token = report.get('nextPageToken', None)
                        if next_page_token is not None:
                            break

    # print the df
    p_print(ga_df.head(), "HEAD of resulting Dataframe", "")

    return ga_df
```

GA4

- Meltano -
 - Extractor - [tap-google-analytics-v4](#)

```
plugins:
  extractors:
    - name: tap-google-analytics-v4
      namespace: tap_google_analytics_v4
      pip_url: git+https://github.com/ /tap-google-analytics-v4.git
      executable: tap-google-analytics
      capabilities:
        - catalog
        - discover
        - state
        - about
        - stream-maps
      settings:
        - name: key_file_location
        - name: reports
        - name: property_id
        - name: start_date
          kind: date_iso8601
        - name: end_date
          kind: date_iso8601
      config:
        start_date: '2022-12-01'
        end_date: '2022-12-02'
        key_file_location: $MELTANO_PROJECT_ROOT/.meltano/meltano_GA4_client_secrets.json
```

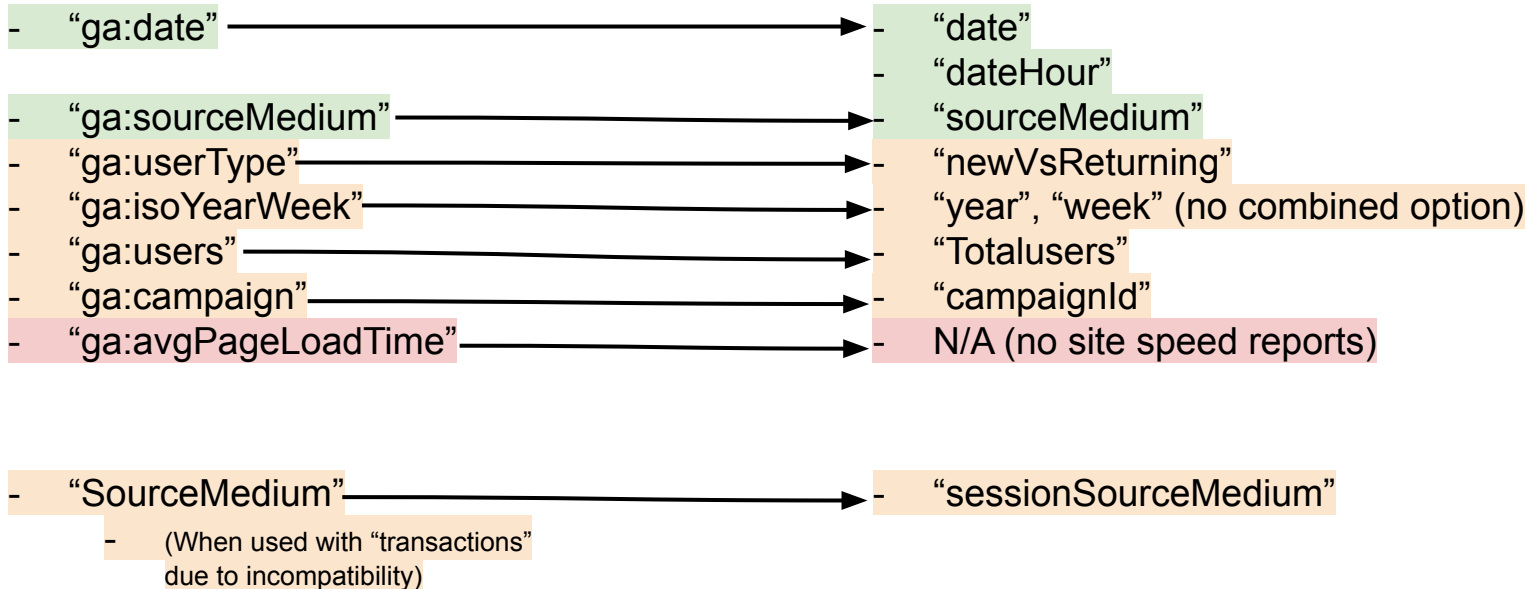
- Loader - [target-s3-jsonl](#)



IV. Dimensions & Metrics

GA3

GA4



V. Data structures

GA3

- Request Body- JSON
 - serializable, class dict
- Response from API -
 - Returning value from batchget(), class dict, is iterable.

GA4

- Request Body - class
 - Google.analytics.data_v1beta, created with RunReportRequest(), Not JSON serializable
- Response from API -
 - Returning Value from run_report(), type Google.analytics.data_v1beta, is not iterable

V. Data structures

GA3

```
{
  "kind": "analytics#gaData",
  "id": string ↗,
  "selfLink": string ↗,
  "containsSampledData": boolean ↗,
  "query": {
    "start-date": string ↗,
    "end-date": string ↗,
    "ids": string ↗,
    "dimensions": [
      string ↗
    ],
    "metrics": [
      string ↗
    ],
    "samplingLevel": string ↗,
    "include-empty-rows": boolean ↗,
    "sort": [
      string ↗
    ],
    "filters": string ↗,
    "segment": string ↗,
    "start-index": integer ↗,
    "max-results": integer ↗
  },
  "itemsPerPage": integer ↗,
  "totalResults": integer ↗,
  "previousLink": string ↗,
  "nextLink": string ↗,
  "profileInfo": {
    "profileId": string ↗,
    "accountId": string ↗,
    "webPropertyId": string ↗,
    "internalWebPropertyId": string ↗,
    "profileName": string ↗,
```

GA4

```
{
  "reports": [
    {
      "columnHeader": {
        "metricHeader": {
          "metricHeaderEntries": [
            {
              "name": "ga:users",
              "type": "INTEGER"
            }
          ]
        }
      },
      "data": {
        "isDataGolden": true,
        "maximums": [
          {
            "values": [
              "98"
            ]
          }
        ],
        "minimums": [
          {
            "values": [
              "98"
            ]
          }
        ],
        "rowCount": 1,
        "rows": [
          {
            "metrics": [
              {
                "values": [
                  "98"
                ]
              }
            ]
          }
        ]
      }
    }
  ]
}
```