 **Talend User Component tGoogleAnalytics4Input**

**Purpose**

This component addresses the needs of gathering Google Analytics 4 data for many properties and fine-grained detail data.

The component uses the Google Analytics API 1.0 beta and the Authentication API OAuth.

To provide the ability to run in multiple iterations the component has special capabilities to avoid multiple logins through iterations. Usually, automated processes should not use personal accounts. This requirement is addressed by using a service account, which are the only preferred way to login into Google Analytics for automated processes.

The account credentials are now only be taken from the standard json key file. This way you can only by changing the json file change the account, also the account type. No need for changing the job to switch between different account types like service account or application client Id.

**Talend-Integration**

This component can be found in the palette under Business->Google

This component provides an output flow and several return values.

**Precondition**

Take care your Google project is enabled for the Google Analytics Data API:

<https://developers.google.com/analytics/devguides/reporting/data/v1?hl=en_GB>

To enable this API, visit the Google Console: <https://console.cloud.google.com/apis/api/analyticsdata.googleapis.com/metrics>

To configure the access rights, visit the Google Analytics and go to the Admin section (see bottom left menu)

https://analytics.google.com/analytics/web/#/report-home/

**Parameters**

**Parameters to connect to Google Analytics (setup client)**

|  |  |  |
| --- | --- | --- |
| **Property** | **Content** | **Data types** |
| Key File (\*.json) | The Account Login works with private key file for authentication. In the process of creating a service account you download this file. Only for service accounts ***Required*** | String |
| GA4-Property-ID | GA4-Property-IDs are numbers. You can provide the value as number or String | String or Long |

If you key file is from a application client Id account than the component starts an approval process and expects on the first run an user interaction with the Google web page and after finishing the form to approve the access right you need to close the browser to let the component continue, otherwise the authentication process will not complete.

**Parameters to define the query**

|  |  |
| --- | --- |
| **Property** | **Content** |
| Start Date | All queries always need a time range (only date, not time).  The value must be a String with the pattern “yyyy-MM-dd”. ***Required!*** |
| End Date | Time range end. If you want gather data for one date, use start date as end date.  The value must be a String with the pattern “yyyy-MM-dd”. ***Required!*** |
| Dimensions | Dimensions are like group clauses. These dimensions will group the metric values.  See advise for notations below. Separate multiple dimensions with a comma.  ***Not required*** |
| Metrics | Things you want to measure. Separate multiple metrics with a comma.  See advise for notations below.  ***Required!*** |
| Dimension Filters | Contains all filters as concatenated string to filter rows by dimension values (cause an AND combination of the filters)  See advise for notation below |
| Metric Filters | Contains all filters as concatenated string to filter rows by metric values (cause an AND combination of the filters)  See advise for notation below |
| Sorts | Contains all sort criteria as concatenated string. Separate multiple dimensions/metrics with a comma.  See advise for notation below |
| Normalize Output | If true, the component normalizes the otherwise flat record into two normalized output flows (dimensions and metrics).  For every raw record with its columns for dimensions and metrics this option creates one record per raw-record and dimension / metric.  E.g. if the component in the flat mode would create 3 records with 4 dimensions and 2 metrics it will create for the dimensions-flow 3 x 4 records and for the metrics flow 3 x 2 records. |
| Exclude date dimension and provide value as return value | Set this to exclude the date dimension from the normalized output flow for dimension and instead set the date value in the globalMap as return value (available while the flow runs). |

**Explanation for the Normalized Output**

The normalized output as made for scenarios in which the job will be configured with metrics and dimensions at runtime. In this use case it is not possible to declare the appropriated schema for the flat output. The normalization creates 2 read only output schemas:

Dimensions

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Meaning** |
| ROW\_NUM | int | The row number from the original flat result row. It identifies the records, which belongs to together. |
| DIMENSION\_NAME | String | Name of the dimension |
| DIMENSION\_VALUE | String | Value of the dimension |

Metrics

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Meaning** |
| ROW\_NUM | int | The row number from the original flat result row. It identifies the records, which belongs together. |
| METRIC\_NAME | String | Name of the metric |
| METRIC\_VALUE | Double | Value of the metric |

**Advice for dimension and metric filter notation**

* For dimensions, metrics, filters and sorts you must use the notation from the former Google Core API and these filters will be translated into the new GA4 filters. Unfortunately, there is currently no other way to express the filters in a generic way like in UA because limitations of the current GA4-API.
* Combine all filter entries with OR or AND ( use comma for OR and semicolon for AND )
* In a filter you can currently only combine all expressions with OR or AND but not mixed.
* On the left side of a filter term is the dimension or metric and on the right side the value to compare.
* Unlike in UA you can now add spaces in your filter expressions to make the better readable.

Filter comparison operators:

|  |  |
| --- | --- |
| **Operator** | **Meaning** |
| == | Exact match to include (both filters) |
| != | Exact match to exclude (both filters) |
| =~ | Regex match to include (only for dimension filters) |
| !~ | Regex match to exclude (only for dimension filters) |
| =@ | Contains string (only for dimension filters) |
| !@ | Does not contain string (only for dimension filters) |
| > | Greater than (only for metric filters) |
| >= | Greater or equals than (only for metric filters) |
| <= | Lower or equals than (only for metric filters) |
| < | Lower than (only for metric filters) |

**Example for dimension filter in OR combination**

"city == Berlin **,** city == Paris"

**Example for dimension filter in AND combination:**

"city == (not set) **;** country =@ Fran"

**Building Flat Schema for Component**

In the schema you need same number columns equals to the sum of the number of dimensions and metrics.

Columns in the schema must start at first with dimensions (if provided) and ends with metrics.

Schema column types must match to the data types of the dimensions and metrics. The schema column names can differ from the names of dimensions and metrics. Only the order and their types are important.

In Talend schema columns must follow the Java naming rules therefore avoid writing customEvent:xxx instead e.g. use the name without the namespace prefix.

Important: For date dimensions (e.g.: date) you must specify the date pattern as “yyyyMMdd” if you want it as Date typed value.

**Advanced Option Parameters**

|  |  |
| --- | --- |
| **Property** | **Content** |
| Timeout in s | How long should the component wait for getting the first result and fetching all result with one internal iteration |
| Fetch Size | This is the amount of data the component fetches at once. The value is used to set the max\_rows attribute. max\_rows mean not the absolute amount of data! The component manages setting the start index to get all data. To achieve this, the component iterates if the last result set are completely fetched. |
| Local Number Format | You can get numbers in various formats. Here you can define the locale in which format double or float values are should textually format by the API. |
| Reuse Client for Iterations | If you use this component in iterations, it is strongly recommended to set this option. It saves time to authenticate unnecessary often and avoids problems because of max amount of connects per time range. |

**Return values**

|  |  |
| --- | --- |
| **Return value** | **Content** |
| ERROR\_MESSAGE | Last error message |
| NB\_LINE | Number of delivered lines (only set if normalization is not used) |
| TOTAL\_AFFECTED\_ROWS | Number of rows, which are collected by Google to calculate the result. |
| NB\_LINE\_DIMENSIONS | Number of normalized dimension records (only set if normalization is used) |
| NB\_LINE\_METRICS | Number of normalized metric records (only set if normalization is used) |
| CURRENT\_DATE | The value of the date dimension (if present in the file) for every row. This value is only available in the “Normalized Flow” mode. Type: java.util.Date |
| ERROR\_CODE | The last error code as Integer. Default value is 0 at start. |

**Scenario 1: Plain flat output**

Graphical user interface

Description automatically generated

The schema must be setup with the dimensions first and then the metrics. If dimensions are not used, start with the metrics.

This is the output:

Table

Description automatically generated

**Scenario 2: Normalized output**

This mode helps you to store the dimension and metric values for different reports in the same table design.

Graphical user interface

Description automatically generated with medium confidence

Output for normalized mode in in job (left) compared to the same output in plain flat mode (right)

Table

Description automatically generatedTable

Description automatically generated