Salesforce1 Reporting REST API Developer Guide

Version 32.0, Winter '15

@salesforcedocs

Last updated: September 26, 2014

CONTENTS

CHAPTER 1 Introducing the Salesforce1 Reporting REST API

In this chapter ...

The REST-based Salesforce1 Reporting API gives you programmatic access to your report and dashboard data as defined in the report builder and dashboard builder. The API lets you integrate the data into any web or mobile application, inside or outside the Salesforce platform. For example, you might use the API to trigger a Chatter post with a snapshot of top-performing reps each quarter.

The Salesforce1 Reporting API will revolutionize the way you access and visualize your data. You can:

- Integrate report data into custom objects.
- Define rich visualizations on top of the API to animate the data.
- Build custom dashboards.
- Automate reporting tasks.

At a high level, the API resources let you query and filter report data. You can:

- Run tabular, summary, or matrix reports synchronously or asynchronously.
- Filter for specific data on the fly.
- Query report metadata.

You can also work with dashboard resources to:

- Get a list of recently used dashboards.
- Get dashboard metadata and data.
- Query dashboard status.
- Refresh dashboards.

Requirements and Limitations

The Salesforce1 Reporting API is available for any organization that has API enabled. You must establish an authenticated session using OAuth in order to access the Salesforce1 Reporting API. When working with the Salesforce1 Reporting API, consider these restrictions in addition to general API limits.

Note: Responses and requests are in JSON. While using the Salesforce1 Reporting API with a POST request body, you must use content-type: application/json. You might get unexpected results if you don't use this content type.

Reports Limits

- Cross filters, standard report filters, and filtering by row limit are unavailable when filtering data.
- Historical trend reports are only supported for matrix reports.
- The API can process only reports that contain up to 100 fields selected as columns.
- A list of up to 200 recently viewed reports can be returned.
- Your organization can request up to 500 synchronous report runs per hour.
- The API supports up to 20 synchronous report run requests at a time.
- A list of up to 2,000 instances of a report that was run asynchronously can be returned.
- The API supports up to 200 requests at a time to get results of asynchronous report runs.
- Your organization can request up to 1,200 asynchronous requests per hour.
- Asynchronous report run results are available within a 24-hour rolling period.
- The API returns up to the first 2,000 report rows. You can narrow results using filters.
- You can add up to 20 custom field filters when you run a report.

Dashboards Limits

- Your organization can request up to 200 dashboard refreshes per hour.
- Your organization can request results for up to 5,000 dashboards per hour.

Note: All limits that apply to reports created in the report builder also apply to the API, as do limits for dashboards created in the dashboard builder. For more information, see "Salesforce1 Reporting Limits" in the Salesforce online help.

CHAPTER 2 Understanding Reports REST API Resources

In this chapter ...

The Reports API is designed to let you query report data easily. Use the Reports API to:

- Run Reports Synchronously or Asynchronously.

  Run a report immediately or asynchronously to get summary data with or without details. We recommend that you run reports asynchronously to avoid report timeouts and other API limits.

- Get Report Metadata.

  Get information about fields in the report and report type. This includes information about fields used for report groupings, summaries, detailed data, and filters.

- List Asynchronous Runs of a Report.

  Get a list of all instances of a report run asynchronously.

- Filter Reports on Demand.

  Get specific data back by running a report with filter changes in the metadata.

- List Recently Viewed Reports.

  Get most recently viewed reports that you have permission to access.

- Decode the Fact Map.

  Get a visualized view of your report data.

Run Reports Synchronously or Asynchronously

Get summary data with or without details by running a report synchronously or asynchronously through the API. When you run a report, the API returns data for the same number of records that are available when the report is run in the Salesforce user interface.

Run a report synchronously if you expect it to finish running quickly. Otherwise, we recommend that you run reports through the API asynchronously for these reasons:

- Long running reports have a lower risk of reaching the timeout limit when run asynchronously.
- The 2-minute overall Salesforce API timeout limit doesn't apply to asynchronous runs.
- The Salesforce1 Reporting API can handle a higher number of asynchronous run requests at a time.
- Since the results of an asynchronously run report are stored for a 24-hr rolling period, they're available for recurring access.

To run a report synchronously:

- Send a GET or POST request to the Execute Sync resource to get data.
- Use a POST request to get specific results on the fly by passing filters in the report metadata.

To fetch report data asynchronously:

1. Send a POST request to the Execute Async resource. If you're passing filters, include them in the POST request metadata. The request returns the instance ID where results of the run are stored.
2. Send a GET request to the Instance Results resource to fetch data using the instance ID.

Example of a synchronous report run

This GET request to the Execute Sync resource, `/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK?includeDetails=true`, for a synchronous run returns summary data with details.

```
{
    "hasDetailRows": true,
    "attributes": {
        "instancesUrl":
"/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK/instances",
       "describeUrl": "/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK/describe",

        "type": "Report",
        "reportId": "00OD0000001ZbP7MAK",
        "reportName": "Stuck Opportunities"
    },
    "groupingsDown": {
        "groupings": [
...
        ]
    },
    "groupingsAcross": {
        "groupings": [
...
        ]
    },
    "reportExtendedMetadata": {
...
```

```
    },
    "allData": true,
    "reportMetadata": {
        "name": "Stuck Opportunities",
        "id": "00OD0000001ZbP7MAK",
        "currency": null,
        "developerName": "StuckOpportunities",
        "groupingsDown": [
            {
                "name": "ACCOUNT_NAME",
                "sortOrder": "Asc",
                "dateGranularity": "None"
            },
            {
                "name": "CLOSE_DATE",
                "sortOrder": "Desc",
                "dateGranularity": "FiscalQuarter"
            }
        ],
        "groupingsAcross": [
            {
                "name": "STAGE_NAME",
                "sortOrder": "Desc",
                "dateGranularity": "None"
            },
            {
                "name": "ACCOUNT_LAST_ACTIVITY",
                "sortOrder": "Asc",
                "dateGranularity": "Week"
            }
        ],
        "reportType": {
            "type": "Opportunity",
            "label": "Opportunities"
        },
        "aggregates": [
            "s!EXP_AMOUNT",
            "FORMULA1",
            "RowCount"
        ],
        "reportFormat": "MATRIX",
        "reportBooleanFilter": null,
        "reportFilters": [
            {
                "value": "Closed Won,Closed Lost",
                "column": "STAGE_NAME",
                "operator": "notEqual"
            },
            {
                "value": "50",
                "column": "PROBABILITY",
                "operator": "greaterThan"
            }
        ],
```

```
        "detailColumns": [
            "OPPORTUNITY_NAME",
            "PROBABILITY",
            "EXP_AMOUNT",
            "NEXT_STEP",
            "BucketField_34840671"
        ]
    },
    "factMap": {
...
        "4_0!0_2": {
            "rows": [
                {
                    "dataCells": [
                        {
                            "value": "006D000000CrRLxIAN",
                            "label": "DFC Inc. - 10K"
                        },
                        {
                            "value": 90,
                            "label": "90%"
                        },
                        {
                            "value": {
                                "amount": 9450,
                                "currency": null
                            },
                            "label": "$9,450.00"
                        },
                        {
                            "value": "Signed SOW",
                            "label": "Signed SOW"
                        },
                        {
                            "value": "Chemicals",
                            "label": "Chemicals"
                        }
                    ]
                }
            ],
            "aggregates": [
                {
                    "value": 9450,
                    "label": "$9,450.00"
                },
                {
                    "value": 9450,
                    "label": "9,450.00"
                },
                {
                    "value": 1,
                    "label": "1"
                }
            ]
```

```
        },
...
        "4_1!0_2": {
            "rows": [
                {
                    "dataCells": [
                        {
                            "value": "006D000000CrRLyIAN",
                            "label": "DFC Inc. - 12K"
                        },
                        {
                            "value": 90,
                            "label": "90%"
                        },
                        {
                            "value": {
                                "amount": 10800,
                                "currency": null
                            },
                            "label": "$10,800.00"
                        },
                        {
                            "value": "Signed SOW",
                            "label": "Signed SOW"
                        },
                        {
                            "value": "Chemicals",
                            "label": "Chemicals"
                        }
                    ]
                }
            ],
            "aggregates": [
                {
                    "value": 10800,
                    "label": "$10,800.00"
                },
                {
                    "value": 10800,
                    "label": "10,800.00"
                },
                {
                    "value": 1,
                    "label": "1"
                }
            ]
        },
...
    }
}
```

Example of an asynchronous report run

1. This is a POST request, `/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK/instances`, to the Execute Async resource for an asynchronous run requesting summary results.

```
{
    "reportMetadata": {
        "name": "StuckOpportunities",
        "id": "00OD0000001ZbP7MAK",
        "currency": null,
        "developerName": "StuckOpportunities",
        "reportType": {
            "type": "Opportunity",
            "label": "Opportunities"
        },
        "aggregates": [
            "s!EXP_AMOUNT",
            "FORMULA1",
            "RowCount"
        ],
        "groupingsDown": [
            {
                "name": "ACCOUNT_NAME",
                "sortOrder": "Asc",
                "dateGranularity": "None"
            },
            {
                "name": "CLOSE_DATE",
                "sortOrder": "Desc",
                "dateGranularity": "FiscalQuarter"
            }
        ],
        "groupingsAcross": [
            {
                "name": "STAGE_NAME",
                "sortOrder": "Desc",
                "dateGranularity": "None"
            },
            {
                "name": "ACCOUNT_LAST_ACTIVITY",
                "sortOrder": "Asc",
                "dateGranularity": "Week"
            }
        ],
        "reportFormat": "MATRIX",
        "reportFilters": [
            {
                "value": "Closed Won,Closed Lost",
                "column": "STAGE_NAME",
                "operator": "notEqual"
            },
            {
                "value": "50",
                "column": "PROBABILITY",
                "operator": "greaterThan"
```

```
            }
        ],
        "detailColumns": [
            "OPPORTUNITY_NAME",
            "PROBABILITY",
            "EXP_AMOUNT",
            "NEXT_STEP",
            "BucketField_34840671"
        ],
        "reportBooleanFilter": null
    }
}
```

The response to the POST request returns the instance handle that stores the summary results of the run.

```
{
    "id": "0LGD000000000IjOAI",
    "requestDate": "2013-08-12T18:39:06Z",
    "status": "New",
    "ownerId": "005D0000001KvxRIAS",
    "url":
"/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK/instances/0LGD000000000IjOAI",

    "hasDetailRows": false,
    "completionDate": null
}
```

2. A GET request, `/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK/instances/0LGD000000000IjOAI`, to the Instance Results resource for the instance handle fetches the report results.

```
{
    "hasDetailRows": false,
    "attributes": {
        "id": "0LGD000000000IjOAI",
        "requestDate": "2013-08-12T18:39:06Z",
        "status": "Success",
        "ownerId": "005D0000001KvxRIAS",
        "completionDate": "2013-08-12T18:39:07Z",
        "type": "ReportInstance",
        "reportId": "00OD0000001ZbP7MAK",
        "reportName": "Stuck Opportunities"
    },
    "groupingsDown": {
        "groupings": [
...
        ]
    },
    "groupingsAcross": {
        "groupings": [
            {
                "value": "Negotiation/Review",
                "key": "0",
                "label": "Negotiation/Review",
                "groupings": [
```

```
                    {
                        "value": null,
                        "key": "0_0",
                        "label": "-",
                        "groupings": []
                    },
                    {
                        "value": "2012-07-08",
                        "key": "0_1",
                        "label": "7/8/2012 - 7/14/2012",
                        "groupings": []
                    },
                    {
                        "value": "2013-07-14",
                        "key": "0_2",
                        "label": "7/14/2013 - 7/20/2013",
                        "groupings": []
                    }
                ]
            },
...
        ]
    },
    "reportExtendedMetadata": {
        "detailColumnInfo": {
            "OPPORTUNITY_NAME": {
                "label": "Opportunity Name",
                "dataType": "string"
            },
            "PROBABILITY": {
                "label": "Probability (%)",
                "dataType": "percent"
            },
            "EXP_AMOUNT": {
                "label": "Expected Revenue",
                "dataType": "currency"
            },
            "NEXT_STEP": {
                "label": "Next Step",
                "dataType": "string"
            },
            "BucketField_34840671": {
                "label": "Industry",
                "dataType": "string"
            }
        },
        "aggregateColumnInfo": {
            "RowCount": {
                "label": "Record Count",
                "dataType": "int",
                "downGroupingContext": null,
                "acrossGroupingContext": null
            },
            "FORMULA1": {
```

```
                "label": "formula1",
                "dataType": "double",
                "downGroupingContext": "ALL_SUMMARY_LEVELS",
                "acrossGroupingContext": "ALL_SUMMARY_LEVELS"
            },
            "s!EXP_AMOUNT": {
                "label": "Sum of Expected Revenue",
                "dataType": "currency",
                "downGroupingContext": null,
                "acrossGroupingContext": null
            }
        },
        "groupingColumnInfo": {
            "CLOSE_DATE": {
                "label": "Close Date",
                "dataType": "date",
                "groupingLevel": 1
            },
            "STAGE_NAME": {
                "label": "Stage",
                "dataType": "picklist",
                "groupingLevel": 0
            },
            "ACCOUNT_NAME": {
                "label": "Account Name",
                "dataType": "string",
                "groupingLevel": 0
            },
            "ACCOUNT_LAST_ACTIVITY": {
                "label": "Account: Last Activity",
                "dataType": "date",
                "groupingLevel": 1
            }
        }
    },
    "allData": true,
    "reportMetadata": {
        "name": "Stuck Opportunities",
        "id": "00OD0000001ZbP7MAK",
        "currency": null,
        "developerName": "StuckOpportunities",
        "groupingsDown": [
            {
                "name": "ACCOUNT_NAME",
                "sortOrder": "Asc",
                "dateGranularity": "None"
            },
            {
                "name": "CLOSE_DATE",
                "sortOrder": "Desc",
                "dateGranularity": "FiscalQuarter"
            }
        ],
        "groupingsAcross": [
```

```
            {
                "name": "STAGE_NAME",
                "sortOrder": "Desc",
                "dateGranularity": "None"
            },
            {
                "name": "ACCOUNT_LAST_ACTIVITY",
                "sortOrder": "Asc",
                "dateGranularity": "Week"
            }
        ],
        "reportType": {
            "type": "Opportunity",
            "label": "Opportunities"
        },
        "aggregates": [
            "s!EXP_AMOUNT",
            "FORMULA1",
            "RowCount"
        ],
        "reportFormat": "MATRIX",
        "reportBooleanFilter": null,
        "reportFilters": [
            {
                "value": "Closed Won,Closed Lost",
                "column": "STAGE_NAME",
                "operator": "notEqual"
            },
            {
                "value": "50",
                "column": "PROBABILITY",
                "operator": "greaterThan"
            }
        ],
        "detailColumns": [
            "OPPORTUNITY_NAME",
            "PROBABILITY",
            "EXP_AMOUNT",
            "NEXT_STEP",
            "BucketField_34840671"
        ]
    },
    "factMap": {
        "4_0!2_0": {
            "aggregates": [
                {
                    "value": 0,
                    "label": "$0.00"
                },
                {
                    "value": null,
                    "label": "-"
                },
                {
```

```
                    "value": 0,
                    "label": "0"
                }
            ]
        },
        "12_0!4": {
            "aggregates": [
                {
                    "value": 0,
                    "label": "$0.00"
                },
                {
                    "value": null,
                    "label": "-"
                },
                {
                    "value": 0,
                    "label": "0"
                }
            ]
        },
...
    }
}
```

SEE ALSO:

- Execute Sync
- Instances List
- Instance Results

Get Report Metadata

Report metadata gives information about a report and its report type. It includes information on fields used in the report for filters, groupings, detailed data, and summaries. You can use the metadata to do several things.

- Find out what fields in the report type you can filter on and by what values.
- Build custom chart visualizations using the metadata information on fields, groupings, detailed data, and summaries.
- Change filters in the report metadata during a report run.

To get report metadata, send a GET request to the Describe resource.

Example

This GET request, `/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK/describe`, to the Describe resource returns metadata for a matrix report. This includes a bucket field, groupings, summaries, and a custom summary formula.

```
{
    "reportTypeMetadata": {
        "categories": [
            {
```

```
                "label": "Opportunity Information",
                "columns": {
                    "CREATED": {
                        "filterValues": [],
                        "label": "Created By",
                        "dataType": "string",
                        "filterable": true
                    },
...
                    "TYPE": {
                        "filterValues": [
                            {
                                "name": "Add-On Business",
                                "label": "Add-On Business"
                            },
                            {
                                "name": "New Business",
                                "label": "New Business"
                            },
                            {
                                "name": "Services",
                                "label": "Services"
                            }
                        ],
                        "label": "Type",
                        "dataType": "picklist",
                        "filterable": true
                    },
        }
...
    },
    "reportExtendedMetadata": {
        "detailColumnInfo": {
            "OPPORTUNITY_NAME": {
                "label": "Opportunity Name",
                "dataType": "string"
            },
            "PROBABILITY": {
                "label": "Probability (%)",
                "dataType": "percent"
            },
            "EXP_AMOUNT": {
                "label": "Expected Revenue",
                "dataType": "currency"
            },
            "NEXT_STEP": {
                "label": "Next Step",
                "dataType": "string"
            },
            "BucketField_34840671": {
                "label": "Industry",
                "dataType": "string"
            }
        },
```

```
        "aggregateColumnInfo": {
            "RowCount": {
                "label": "Record Count",
                "dataType": "int",
                "downGroupingContext": null,
                "acrossGroupingContext": null
            },
            "FORMULA1": {
                "label": "formula1",
                "dataType": "double",
                "downGroupingContext": "ALL_SUMMARY_LEVELS",
                "acrossGroupingContext": "ALL_SUMMARY_LEVELS"
            },
            "s!EXP_AMOUNT": {
                "label": "Sum of Expected Revenue",
                "dataType": "currency",
                "downGroupingContext": null,
                "acrossGroupingContext": null
            }
        },
        "groupingColumnInfo": {
            "CLOSE_DATE": {
                "label": "Close Date",
                "dataType": "date",
                "groupingLevel": 1
            },
            "STAGE_NAME": {
                "label": "Stage",
                "dataType": "picklist",
                "groupingLevel": 0
            },
            "ACCOUNT_NAME": {
                "label": "Account Name",
                "dataType": "string",
                "groupingLevel": 0
            },
            "ACCOUNT_LAST_ACTIVITY": {
                "label": "Account: Last Activity",
                "dataType": "date",
                "groupingLevel": 1
            }
        }
    },
    "reportMetadata": {
        "name": "Stuck Opportunities",
        "id": "00OD0000001ZbP7MAK",
        "currency": null,
        "developerName": "StuckOpportunities",
        "groupingsDown": [
            {
                "name": "ACCOUNT_NAME",
                "sortOrder": "Asc",
                "dateGranularity": "None"
            },
```

```
            {
                "name": "CLOSE_DATE",
                "sortOrder": "Desc",
                "dateGranularity": "FiscalQuarter"
            }
        ],
        "groupingsAcross": [
            {
                "name": "STAGE_NAME",
                "sortOrder": "Desc",
                "dateGranularity": "None"
            },
            {
                "name": "ACCOUNT_LAST_ACTIVITY",
                "sortOrder": "Asc",
                "dateGranularity": "Week"
            }
        ],
        "reportType": {
            "type": "Opportunity",
            "label": "Opportunities"
        },
        "aggregates": [
            "s!EXP_AMOUNT",
            "FORMULA1",
            "RowCount"
        ],
        "reportFormat": "MATRIX",
        "reportBooleanFilter": null,
        "reportFilters": [
            {
                "value": "Closed Won,Closed Lost",
                "column": "STAGE_NAME",
                "operator": "notEqual"
            },
            {
                "value": "50",
                "column": "PROBABILITY",
                "operator": "greaterThan"
            }
        ],
        "detailColumns": [
            "OPPORTUNITY_NAME",
            "PROBABILITY",
            "EXP_AMOUNT",
            "NEXT_STEP",
            "BucketField_34840671"
        ]
    }
}
```


SEE ALSO:

Describe

List Asynchronous Runs of a Report

You can get as many as 2000 instances of a report for which you requested asynchronous runs by sending a GET request to the Instances List resource. The instance list is sorted by the date when the run was requested. Report results are stored for a rolling 24-hour period. During this time, based on your user access level, you can access results for each instance of the report that was run.

Example

A GET request, `/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK/instances`, to the Instances List resource returns two instances of the report that was run asynchronously. Each URL handle stores report results for that instance.

```
[
    {
        "id": "0LGD000000000IyOAI",
        "requestDate": "2013-08-12T19:06:47Z",
        "status": "Success",
        "url":
"/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK/instances/0LGD000000000IyOAI",

        "ownerId": "005D0000001KvxRIAS",
        "hasDetailRows": false,
        "completionDate": "2013-08-12T19:06:48Z"
    },
    {
        "id": "0LGD000000000IjOAI",
        "requestDate": "2013-08-12T18:39:06Z",
        "status": "Success",
        "url":
"/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK/instances/0LGD000000000IjOAI",

        "ownerId": "005D0000001KvxRIAS",
        "hasDetailRows": false,
        "completionDate": "2013-08-12T18:39:07Z"
    }
]
```

SEE ALSO:

Instances List

Filter Reports on Demand

To get specific results on the fly, filter reports through the API. Filter changes made through the API does not affect the source report definition. Using the API, you can filter with up to 20 custom field filters and add filter logic (such as AND, OR). But standard filters (such as range), filtering by row limit, and cross filters are unavailable.

Before you filter a report, it's helpful to check these properties in the metadata that tell you if a field can be filtered, the values and criteria you can filter by, and filters that already exist in the report.

- `filterable`
- `filterValues`

- `dataTypeFilterOperatorMap`
- `reportFilters`

You can filter reports during synchronous or asynchronous report runs by making a POST request to the Execute Sync or Execute Async resource.

Example

In a POST request, an accounts report is filtered synchronously by these passing filters with filter logic in the metadata to the Execute Sync resource.

1. `Account Name` not equal to Data Mart
2. `Account Owner` not equal to Admin User
3. `Annual Revenue` greater than "100,000"
4. `Industry` not equal to Manufacturing,Recreation

Filter logic: (1 OR 4) AND 2 AND 3.

```
{
    "reportMetadata": {
        "name": "FilterAcctsReport",
        "id": "00OD0000001cw27MAA",
        "reportFormat": "SUMMARY",
        "reportBooleanFilter": "(1OR4)AND2AND3",
        "reportFilters": [
            {
                "value": "DataMart",
                "operator": "notEqual",
                "column": "ACCOUNT.NAME"
            },
            {
                "value": "AdminUser",
                "operator": "notEqual",
                "column": "USERS.NAME"
            },
            {
                "value": "\"100,000\"",
                "operator": "greaterThan",
                "column": "SALES"
            },
            {
                "value": "Manufacturing,Recreation",
                "operator": "notEqual",
                "column": "INDUSTRY"
            }
        ],
        "detailColumns": [
            "RATING",
            "LAST_UPDATE",
            "SALES"
        ],
        "developerName": "Filter_Accts_Report",
        "reportType": {
```

```
            "type": "AccountList",
            "label": "Accounts"
        },
        "currency": null,
        "aggregates": [
            "s!SALES",
            "RowCount"
        ],
        "groupingsDown": [
            {
                "name": "USERS.NAME",
                "sortAggregate": "s!SALES",
                "sortOrder": "Desc",
                "dateGranularity": "None"
            },
            {
                "name": "ACCOUNT.NAME",
                "sortAggregate": null,
                "sortOrder": "Asc",
                "dateGranularity": "None"
            },
            {
                "name": "DUE_DATE",
                "sortAggregate": null,
                "sortOrder": "Asc",
                "dateGranularity": "Month"
            }
        ],
        "groupingsAcross": []
    }
}
```

In response to the POST request, the report returns data that meets the given criteria.

```
{
    "hasDetailRows": false,
    "attributes": {
       "describeUrl": "/services/data/v29.0/analytics/reports/00OD0000001cw27MAA/describe",

        "instancesUrl":
"/services/data/v29.0/analytics/reports/00OD0000001cw27MAA/instances",
        "type": "Report",
        "reportName": "Filter Accts Report",
        "reportId": "00OD0000001cw27MAA"
    },
    "factMap": {
        "1_0!T": {
            "aggregates": [
                {
                    "value": 56000000,
                    "label": "$56,000,000"
                },
                {
                    "value": 1,
                    "label": "1"
```

```
                }
            ]
        },
        "7_1!T": {
            "aggregates": [
                {
                    "value": 24000000,
                    "label": "$24,000,000"
                },
                {
                    "value": 1,
                    "label": "1"
                }
            ]
        },
...
    "allData": true,
    "reportMetadata": {
        "name": "Filter Accts Report",
        "id": "00OD0000001cw27MAA",
        "reportFormat": "SUMMARY",
        "reportBooleanFilter": "(1 OR 4) AND 2 AND 3",
        "reportFilters": [
            {
                "value": "Data Mart",
                "operator": "notEqual",
                "column": "ACCOUNT.NAME"
            },
            {
                "value": "Admin User",
                "operator": "notEqual",
                "column": "USERS.NAME"
            },
            {
                "value": "\"100,000\"",
                "operator": "greaterThan",
                "column": "SALES"
            },
            {
                "value": "Manufacturing,Recreation",
                "operator": "notEqual",
                "column": "INDUSTRY"
            }
        ],
        "detailColumns": [
            "RATING",
            "LAST_UPDATE",
            "SALES"
        ],
...
```

```
    }
}
```

SEE ALSO:

Execute Sync

List Recently Viewed Reports

Get up to 200 of the reports you most recently viewed in Salesforce by sending a GET request to the List resource.

Each report listing in the response has resource URLs to get metadata and run a report asynchronously or synchronously.

For a more extensive reports list, query the Report object using a SOQL query in a Salesforce API such as SOAP API or REST API. This SOQL query, for example, returns all reports that are in matrix format: `SELECT Description,Format,LastRunDate FROM Report WHERE Format = 'MATRIX' ORDER BY Id ASC NULLS FIRST`

Example

This GET request `/services/data/v29.0/analytics/reports` to the List resource returns a list of 5 recently viewed reports.

```
[
    {
        "name": "Total # Accounts",
        "id": "00OD0000001ZbJqMAK",
        "url": "/services/data/v29.0/analytics/reports/00OD0000001ZbJqMAK",
       "describeUrl": "/services/data/v29.0/analytics/reports/00OD0000001ZbJqMAK/describe",

       "instancesUrl": "/services/data/v29.0/analytics/reports/00OD0000001ZbJqMAK/instances"

    },
    {
        "name": "Open Cases by Account",
        "id": "00OD0000001ZbLzMAK",
        "url": "/services/data/v29.0/analytics/reports/00OD0000001ZbLzMAK",
       "describeUrl": "/services/data/v29.0/analytics/reports/00OD0000001ZbLzMAK/describe",

       "instancesUrl": "/services/data/v29.0/analytics/reports/00OD0000001ZbLzMAK/instances"

    },
    {
        "name": "Usage - Neglected Opportunities",
        "id": "00OD0000001ZbJbMAK",
        "url": "/services/data/v29.0/analytics/reports/00OD0000001ZbJbMAK",
       "describeUrl": "/services/data/v29.0/analytics/reports/00OD0000001ZbJbMAK/describe",

       "instancesUrl": "/services/data/v29.0/analytics/reports/00OD0000001ZbJbMAK/instances"

    },
    {
        "name": "Won opptys with partners",
        "id": "00OD0000001bvOeMAI",
```

```
        "url": "/services/data/v29.0/analytics/reports/00OD0000001bvOeMAI",
       "describeUrl": "/services/data/v29.0/analytics/reports/00OD0000001bvOeMAI/describe",

       "instancesUrl": "/services/data/v29.0/analytics/reports/00OD0000001bvOeMAI/instances"

    },
    {
        "name": "Usage - Accounts Last Modified",
        "id": "00OD0000001ZbJdMAK",
        "url": "/services/data/v29.0/analytics/reports/00OD0000001ZbJdMAK",
       "describeUrl": "/services/data/v29.0/analytics/reports/00OD0000001ZbJdMAK/describe",

       "instancesUrl": "/services/data/v29.0/analytics/reports/00OD0000001ZbJdMAK/instances"

    },
]
```


SEE ALSO:

List

Decode the Fact Map

Depending on how you run a report, the fact map in the report results can contain values for only summary or both summary and detailed data. The fact map values are expressed as keys, which you can programmatically use to visualize the report data. Fact map keys provide an index into each section of a fact map, from which you can access summary and detailed data.

The pattern for the fact map keys varies by report format as shown in this table.

Report format	Fact map key pattern
Tabular	`T!T`: The grand total of a report. Both record data values and the grand total are represented by this key.
Summary	***<First level row grouping_second level row grouping_third level row grouping>***`!T`: T refers to the row grand total.
Matrix	***<First level row grouping_second level row grouping>!<First level column grouping_second level column grouping>***.

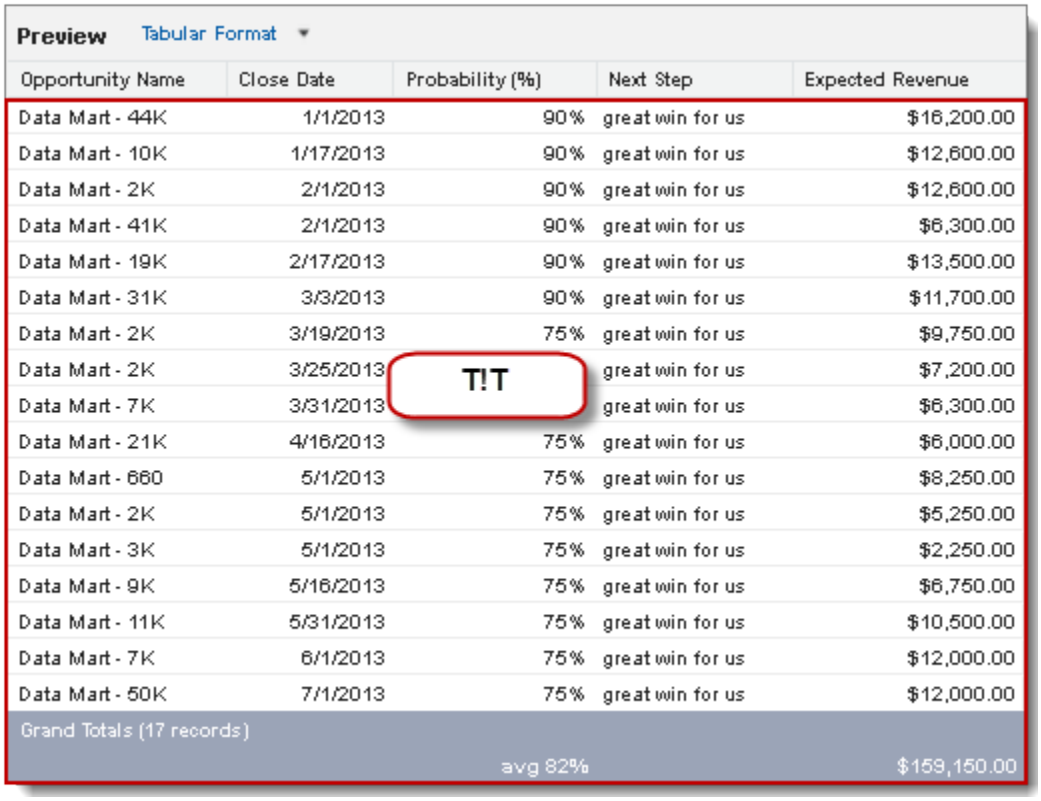
Each item in a row or column grouping is numbered starting with `0`. Here are some examples of fact map keys:

Fact Map Key	Description
`0!T`	The first item in the first-level grouping.
`1!T`	The second item in the first-level grouping.
`0_0!T`	The first item in the first-level grouping and the first item in the second-level grouping.
`0_1!T`	The first item in the first-level grouping and the second item in the second-level grouping.

Let's look at examples of how fact map keys represent data as it appears in a Salesforce tabular, summary, or matrix report.

Tabular Report Fact Map

Here's an example of an opportunities report in tabular format. Since tabular reports don't have groupings, all of the record level data and summaries are expressed by the `T!T` key, which refers to the grand total.

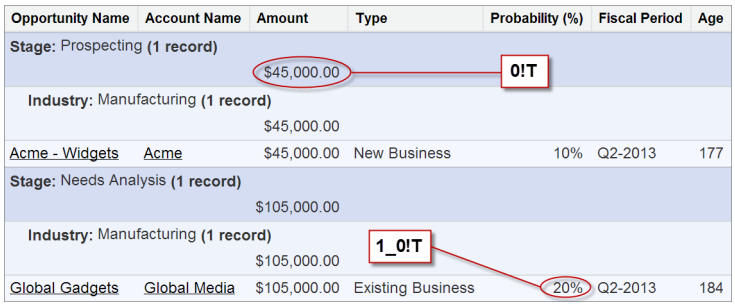


Preview Tabular Format

Opportunity Name	Close Date	Probability (%)	Next Step	Expected Revenue
Data Mart - 44K	1/1/2013	90%	great win for us	$16,200.00
Data Mart - 10K	1/17/2013	90%	great win for us	$12,600.00
Data Mart - 2K	2/1/2013	90%	great win for us	$12,600.00
Data Mart - 41K	2/1/2013	90%	great win for us	$6,300.00
Data Mart - 19K	2/17/2013	90%	great win for us	$13,500.00
Data Mart - 31K	3/3/2013	90%	great win for us	$11,700.00
Data Mart - 2K	3/19/2013	75%	great win for us	$9,750.00
Data Mart - 2K	3/25/2013	T!T	great win for us	$7,200.00
Data Mart - 7K	3/31/2013		great win for us	$6,300.00
Data Mart - 21K	4/16/2013	75%	great win for us	$6,000.00
Data Mart - 660	5/1/2013	75%	great win for us	$8,250.00
Data Mart - 2K	5/1/2013	75%	great win for us	$5,250.00
Data Mart - 3K	5/1/2013	75%	great win for us	$2,250.00
Data Mart - 9K	5/16/2013	75%	great win for us	$6,750.00
Data Mart - 11K	5/31/2013	75%	great win for us	$10,500.00
Data Mart - 7K	6/1/2013	75%	great win for us	$12,000.00
Data Mart - 50K	7/1/2013	75%	great win for us	$12,000.00
Grand Totals (17 records)		avg 82%		$159,150.00

Summary Report Fact Map

This example shows how the values in a summary report are represented in the fact map.

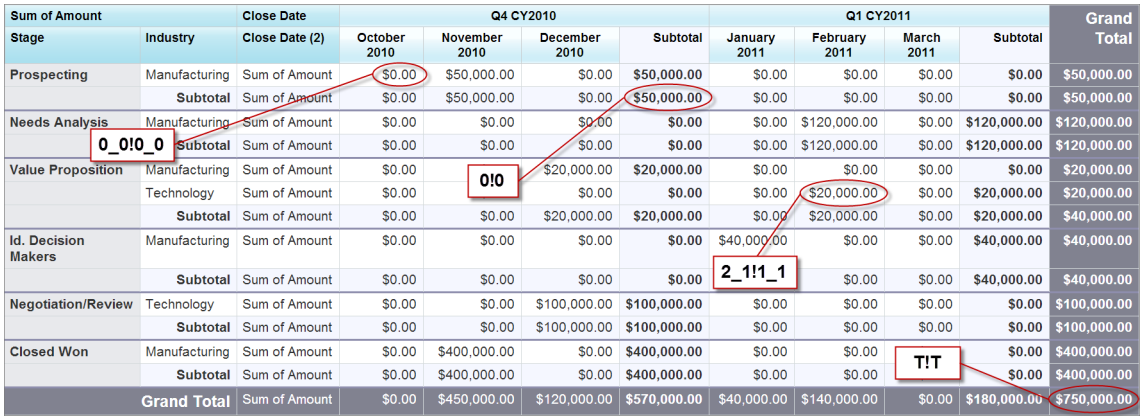


Opportunity Name	Account Name	Amount	Type	Probability (%)	Fiscal Period	Age
Stage: Prospecting **(1 record)**		$45,000.00		0!T		
Industry: Manufacturing **(1 record)**		$45,000.00				
Acme - Widgets	Acme	$45,000.00	New Business	10%	Q2-2013	177
Stage: Needs Analysis **(1 record)**		$105,000.00				
Industry: Manufacturing **(1 record)**		$105,000.00	1_0!T			
Global Gadgets	Global Media	$105,000.00	Existing Business	20%	Q2-2013	184

Fact Map Key	Description
`0!T`	Summary for the value of opportunities in the Prospecting stage.
`1_0!T`	Summary of the probabilities for the Manufacturing opportunities in the Needs Analysis stage.

Matrix Report Fact Map

Here's an example of some fact map keys for data in a matrix opportunities report with a couple of row and column groupings.



Sum of Amount		Close Date	Q4 CY2010				Q1 CY2011				Grand Total
Stage	Industry	Close Date (2)	October 2010	November 2010	December 2010	Subtotal	January 2011	February 2011	March 2011	Subtotal	
Prospecting	Manufacturing	Sum of Amount	$0.00	$50,000.00	$0.00	$50,000.00	$0.00	$0.00	$0.00	$0.00	$50,000.00
	Subtotal	Sum of Amount	$0.00	$50,000.00	$0.00	$50,000.00	$0.00	$0.00	$0.00	$0.00	$50,000.00
Needs Analysis	Manufacturing	Sum of Amount	$0.00	$0.00	$0.00	$0.00	$0.00	$120,000.00	$0.00	$120,000.00	$120,000.00
	Subtotal	Sum of Amount	$0.00	$0.00	$0.00	$0.00	$0.00	$120,000.00	$0.00	$120,000.00	$120,000.00
Value Proposition	Manufacturing	Sum of Amount	$0.00		$20,000.00	$20,000.00	$0.00	$0.00	$0.00	$0.00	$20,000.00
	Technology	Sum of Amount	$0.00		$0.00	$0.00	$0.00	$20,000.00	$0.00	$20,000.00	$20,000.00
	Subtotal	Sum of Amount	$0.00	$0.00	$20,000.00	$20,000.00	$0.00	$20,000.00	$0.00	$20,000.00	$40,000.00
Id. Decision Makers	Manufacturing	Sum of Amount	$0.00	$0.00	$0.00	$0.00	$40,000.00	$0.00	$0.00	$40,000.00	$40,000.00
	Subtotal	Sum of Amount	$0.00	$0.00	$0.00	$0.00	$40,000.00	$0.00	$0.00	$40,000.00	$40,000.00
Negotiation/Review	Technology	Sum of Amount	$0.00	$0.00	$100,000.00	$100,000.00	$0.00	$0.00	$0.00	$0.00	$100,000.00
	Subtotal	Sum of Amount	$0.00	$0.00	$100,000.00	$100,000.00	$0.00	$0.00	$0.00	$0.00	$100,000.00
Closed Won	Manufacturing	Sum of Amount	$0.00	$400,000.00	$0.00	$400,000.00	$0.00	$0.00		$0.00	$400,000.00
	Subtotal	Sum of Amount	$0.00	$400,000.00	$0.00	$400,000.00	$0.00	$0.00		$0.00	$400,000.00
	Grand Total	Sum of Amount	$0.00	$450,000.00	$120,000.00	$570,000.00	$40,000.00	$140,000.00	$0.00	$180,000.00	$750,000.00

Fact Map Key	Description
`0!0`	Total opportunity amount in the Prospecting stage in Q4 2010.
`0_0!0_0`	Total opportunity amount in the Prospecting stage in the Manufacturing sector in October 2010.
`2_1!1_1`	Total value of opportunities in the Value Proposition stage in the Technology sector in February 2011.
`T!T`	Grand total summary for the report.

SEE ALSO:

- Execute Sync
- Execute Async

CHAPTER 3 Understanding Dashboards REST API Resources

In this chapter ...

The Dashboards API is designed to let you access and refresh dashboards easily. Use the API to:

- Get List of Recently Used Dashboards

  Get a list of dashboards with URLs to access status and results.

- Get Dashboard Results

  Get dashboard metadata, data, and status.

- Filter Dashboard Results

  Filter dashboard results, status, or refresh requests.

- Get Dashboard Status

  Get dashboard refresh status.

- Refresh a Dashboard

  Trigger a dashboard refresh.

Get List of Recently Used Dashboards

You can get a list of recently used dashboards by using the Dashboard List resource.

Use a GET request on the Dashboard List resource to retrieve a list of recently used dashboards. The list is sorted by the date when the dashboard was last refreshed.

Example Usage

```
/services/data/v31.0/analytics/dashboards
```

Example Response Body

In this case, the Dashboard List resource returns information for two dashboards. Each URL handle stores the status or results for the dashboard.

```
[ {
  "id" : "01ZD00000007QeuMAE",
  "name" : "Adoption Dashboard",
  "statusUrl" : "/services/data/v31.0/analytics/dashboards/01ZD00000007QeuMAE/status",
  "url" : "/services/data/v31.0/analytics/dashboards/01ZD00000007QeuMAE"
}, {
  "id" : "01ZD00000007QevMAE",
  "name" : "Global Sales Dashboard",
  "statusUrl" : "/services/data/v31.0/analytics/dashboards/01ZD00000007QevMAE/status",
  "url" : "/services/data/v31.0/analytics/dashboards/01ZD00000007QevMAE"
} ]
```

SEE ALSO:

Dashboard List

Get Dashboard Results

You can get dashboard metadata, data, and status by sending a GET request to the Dashboard Results resource.

Use a GET request to the Dashboard Results resource to retrieve metadata, data, and status for a dashboard and its components. The results response contains:

- Metadata: information about the dashboard as a whole, including the dashboard ID, name, component metadata, and any dashboard filters.
- Data: underlying report data for each component, filtered by the optional filter parameters. For more information about filtering, see Filter Dashboard Results on page 30.
- Status: data and refresh status for each component of the dashboard. The data status can be `NODATA`, `DATA`, or `ERROR`. If an error occurs, the component status will contain additional properties with the error code, message, and severity. The refresh status can be `IDLE`, if components are finished running, or `RUNNING`, if components are still being refreshed.

Example Usage

```
/services/data/v31.0/analytics/dashboards/01ZD00000007S89MAE
```

Example Response Body

```
{
  "componentData" : [ {
```

```
"componentId" : "01aD0000000a36LIAQ",
"reportResult" : {
  "attributes" : null,
  "allData" : true,
  "factMap" : {
    "T!T" : {
      "aggregates" : [ {
        "label" : "USD 67,043,365.50",
        "value" : 67043365.5016691833734512329101562 5
      } ]
    },
    "0!T" : {
      "aggregates" : [ {
        "label" : "USD 10,083.33",
        "value" : 10083.33333333333393966313451528549194335937 5
      } ]
    },
    "1!T" : {
      "aggregates" : [ {
        "label" : "USD 25,016,768.67",
        "value" : 25016768.67006600648164749145507812 5
      } ]
    },
    "2!T" : {
      "aggregates" : [ {
        "label" : "USD 42,016,513.50",
        "value" : 42016513.49826984107494354248046875
      } ]
    }
  },
  "groupingsAcross" : null,
  "groupingsDown" : {
    "groupings" : [ {
      "groupings" : [ ],
      "key" : "0",
      "label" : "-",
      "value" : null
    }, {
      "groupings" : [ ],
      "key" : "1",
      "label" : "Existing Business",
      "value" : "Existing Business"
    }, {
      "groupings" : [ ],
      "key" : "2",
      "label" : "New Business",
      "value" : "New Business"
    } ]
  },
  "hasDetailRows" : false,
  "reportExtendedMetadata" : {
    "aggregateColumnInfo" : {
      "s!AMOUNT" : {
        "acrossGroupingContext" : null,
```

```
            "dataType" : "currency",
            "downGroupingContext" : null,
            "label" : "Sum of Amount"
          }
        },
        "detailColumnInfo" : { },
        "groupingColumnInfo" : {
          "TYPE" : {
            "dataType" : "picklist",
            "groupingLevel" : 0,
            "label" : "Type"
          }
        }
      },
      "reportMetadata" : {
        "aggregates" : [ "s!AMOUNT" ],
        "currency" : "USD",
        "detailColumns" : [ ],
        "developerName" : "Simple_Test",
        "groupingsAcross" : [ ],
        "groupingsDown" : [ {
          "dateGranularity" : "None",
          "name" : "TYPE",
          "sortAggregate" : null,
          "sortOrder" : "Asc"
        } ],
        "historicalSnapshotDates" : [ ],
        "id" : "00OD0000001g2nWMAQ",
        "name" : "Simple Test",
        "reportBooleanFilter" : null,
        "reportFilters" : [ ],
        "reportFormat" : "SUMMARY",
        "reportType" : {
          "label" : "Opportunities",
          "type" : "Opportunity"
        }
      }
    },
    "status" : {
      "dataStatus" : "DATA",
      "errorCode" : null,
      "errorMessage" : null,
      "errorSeverity" : null,
      "refreshDate" : "2014-04-09T00:28:16.000+0000",
      "refreshStatus" : "IDLE"
    }
  } ],
  "dashboardMetadata" : {
    "attributes" : {
      "dashboardId" : "01ZD00000007S89MAE",
      "dashboardName" : "Simple Dashboard",
     "statusUrl" : "/services/data/v31.0/analytics/dashboards/01ZD00000007S89MAE/status",

      "type" : "Dashboard"
```

```
    },
    "canChangeRunningUser" : false,
    "components" : [ {
      "componentData" : 0,
      "footer" : null,
      "header" : null,
      "id" : "01aD0000000a36LIAQ",
      "properties" : {
        "aggregateName" : "s!AMOUNT",
        "maxRows" : null,
        "sort" : {
          "column" : "TYPE",
          "sortOrder" : "asc"
        },
        "visualizationProperties" : { },
        "visualizationType" : "Bar"
      },
      "reportId" : "00OD0000001g2nWMAQ",
      "title" : null,
      "type" : "Report"
    } ],
    "description" : null,
    "developerName" : "Simple_Dashboard",
    "filters" : [ {
      "name" : "Amount",
      "options" : [ {
        "alias" : null,
        "endValue" : null,
        "id" : "0ICD00000004CBiOAM",
        "operation" : "greaterThan",
        "startValue" : null,
        "value" : "USD 2000000"
      } ],
      "selectedOption" : null
    } ],
    "id" : "01ZD00000007S89MAE",
    "layout" : {
      "columns" : [ {
        "components" : [ 0 ]
      } ]
    },
    "name" : "Simple Dashboard",
    "runningUser" : {
      "displayName" : "Allison Wheeler",
      "id" : "005D00000016V2qIAE"
    }
  }
}
```


SEE ALSO:

Dashboard Results

Filter Dashboard Results

You can filter dashboard results, status, or refresh requests, by using filter parameters.

Dashboard results are always unfiltered, unless you have specified filter parameters in your request. When requesting a dashboard result, status, or refresh, you can specify up to three optional filter parameters: `filter1`, `filter2` and `filter3`. These parameters allow you to apply filter options, which can be selected from the filters that are currently defined for the dashboard. Filters can be applied to the following requests:

- A GET request on the Dashboard Results resource: returns data filtered by the specified parameters.
- A PUT request on the Dashboard Results resource: refreshes the data that has been filtered by the specified parameters.
- A GET request on the Dashboard Status resource: returns status for the data that has been filtered by the specified parameters.

Example Usage

A dashboard with one filter ("Country") and two options ("United States" and "Canada") appears like this in the dashboard metadata:

```
{
  "dashboardMetadata" : {
    ...

    "filters" : [ {
        "name" : "Country",
        "options" : [ {
            "id" : "0ICxx0000000001GAA",
            "alias" : "United States",
            "operation" : "equals",
            "value" : "US",
            "startValue" : null,
            "endValue" : null
         } ], [ {
            "id" : "0ICxx0000000002GAA",
            "alias" : "Canada",
            "operation" : "equals",
            "value" : "CA",
            "startValue" : null,
            "endValue" : null
         } ],

    ...
}
```

To retrieve dashboard results with a filter of "Country equals Canada" you could make the following GET request:

```
/services/data/v31.0/analytics/dashboards/01Zxx0000000000000?filter1=0ICxx0000000002GAA
```

SEE ALSO:

Dashboard Results

Dashboard Status

Get Dashboard Status

You can get the dashboard status by sending a GET request to the Dashboard Status resource.

Use the Dashboard Status resource to retrieve a status for each component of the dashboard. The components are listed in the order in which they were refreshed. The request returns `IDLE` if a component is not currently being refreshed, and `RUNNING` if a component is currently being refreshed.

Example Usage

To retrieve the status for a dashboard with an ID of `01ZD00000007QevMAE`, you could make the following request:

```
/services/data/v31.0/analytics/dashboards/01ZD00000007QevMAE/status
```

Example Response Body

The response contains the status for each component, along with the refresh date and time:

```
{
  "componentStatus" : [ {
    "componentId" : "01aD0000000J7M7",
    "refreshDate" : "2014-03-10T17:26:07.000+0000",
    "refreshStatus" : "IDLE"
  }, {
    "componentId" : "01aD0000000J7M9",
    "refreshDate" : "2014-03-10T17:26:08.000+0000",
    "refreshStatus" : "IDLE"
  }, {
    "componentId" : "01aD0000000J7MB",
    "refreshDate" : "2014-03-10T17:26:09.000+0000",
    "refreshStatus" : "IDLE"
  } ]
}
```

SEE ALSO:

Dashboard Status

Refresh a Dashboard

You can refresh a dashboard by using a PUT Dashboard Results request.

Use a PUT request on the Dashboard Results resource to trigger a refresh of a dashboard. The refresh response returns the URL of the status resource after the refresh is triggered. If filter parameters are included in the PUT request, only the filtered data will be refreshed. For more information on filtering, see Filter Dashboard Results on page 30.

Example Usage

The following PUT request refreshes the dashboard with the ID of `01ZD00000007S89MAE`.

```
/services/data/v31.0/analytics/dashboards/01ZD00000007S89MAE
```

Example Request Body

None required.

Example Response Body

The response contains the status URL for the refreshed dashboard:

```
{
  "statusUrl" : "/services/data/v31.0/analytics/dashboards/01ZD00000007S89MAE/status"
}
```

SEE ALSO:

Dashboard Results

CHAPTER 4 Reports API Resource Reference

In this chapter ...

Resources for the Reports API are available at `/services/data/<latest API version>/analytics/reports`. You can query each resource with a HTTP method (such as GET). Use these resources to integrate report data directly into your applications.

Resource	Supported HTTP Method	Description
Describe	GET	Gives report metadata. This includes information about fields that are defined in the report as detail columns, summaries, custom summary formulas, filters, and groupings.
Execute Sync	GET, POST	Gives report summary level data with or without details. Returns specific results if you define filters in the metadata of a POST request.
Execute Async	POST	Returns an instance that stores summary level data with or without details for a report run asynchronously. To get specific results, define filters in the metadata of the request.
Instances List	GET	List of instances of a report that were requested for an asynchronous run.
Instance Results	GET	Depending on the type of your request, gives summary level data with or without details for an instance of a report run asynchronously.
List	GET	List of reports that were recently viewed by the API user.

Describe

Retrieves report, report type, and related metadata for a tabular, summary, or matrix report.

- Report type metadata tells you about all the fields available in the report type, those you can filter, and by what filter criteria.
- Report extended metadata tells you about the fields that are summaries, groupings, and contain record details in the report. A property that displays null indicates that its value is not available.
- Report metadata gives information about the report as a whole. Tells you such things as, the report type, format, the fields that are summaries, row or column groupings, filters saved to the report, and so on.

Resource URL

```
/services/data/<latest API version>/analytics/reports/<report ID>/describe
```

Formats

JSON

HTTP methods

GET

Response body

Property	Type	Description
`reportTypeMetadata`	Report type metadata	Fields in each section of a report type plus filter information for those fields.
`reportExtendedMetadata`	Report extended metadata	Additional information about summaries and groupings.
`reportMetadata`	Report metadata	Unique identifiers for groupings and summaries.

Report type metadata

Property	Type	Description
`categories`	Categories[]	All fields in the report type organized by section.
`dataTypeFilterOperatorMap`	Filter operator reference	Lists all the possible field data types that can be used to filter the report. Each data type, such as phone, percent, currency, or picklist has two properties: `name`: Of type string, this is a unique API name for each field type's filter criteria. Use this API name in the metadata to define filter criteria for a report.

Property	Type	Description
		`label`: Of type string, this is the display name for each filter criteria available to fields of a particular data type. For example, `multipicklist` fields can have for filter criteria, "equals," "not equal to," "includes," and "excludes." Bucket fields are considered to be of `string` data type.

Categories

Property	Type	Description
`label`	String	Display name of a section in the report type under which fields are organized. For example, in an Accounts with Contacts custom report type, `Account General` is the display name of the section that has fields on general account information.
`columns`	Column map	Information for all fields in the report type organized under a particular section's unique API name.

Column map

Property	Type	Description
`label`	String	Display name of a field.
`filterValues`	String array	All filter values for a field, if the field data type is of picklist, multi-select picklist, boolean, or checkbox. For example, checkbox fields always have a value of `True` or `False`. For fields of other data types, the filter value is an empty array because their values can't be determined. Filter values have two properties: `name`: Unique API name for a filter value. Of type string. `label`: Display name of a filter value. Of type string.
`dataType`	String	Data type of the field.
`filterable`	Boolean	`False` means that the field is of a type that can't be filtered. For example, fields of the type `Encrypted Text` can't be filtered.

Report extended metadata

Property	Type	Description
`groupingColumnInfo`	Grouping column information	Map of each row or column grouping to its metadata. Contains values for each grouping identified in the groupingsDown and groupingsAcross list.

Property	Type	Description
`detailColumnInfo`	Detail column information	Two properties for each field that has detailed data identified by its unique API name. The detailed data fields are also listed in the report metadata.
`aggregateColumnInfo`	Aggregate column information	Includes all report summaries such as, `Record Count`, `Sum`, `Average`, `Max`, `Min`, and custom summary formulas. Contains values for each summary listed in the report metadata aggregates.

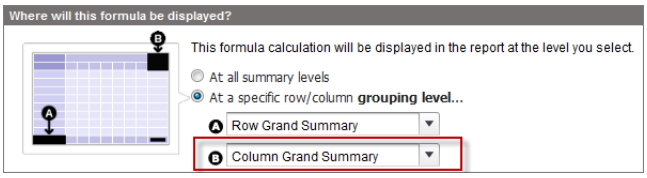
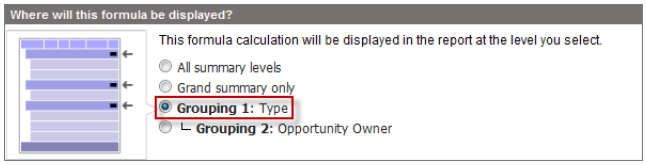
Detail column information

Property	Type	Description
`label`	String	The localized display name of a standard field, the ID of a custom field, or the API name of a bucket field that has detailed data.
`dataType`	String	The data type of the field that has detailed data.

Aggregate column information

Property	Type	Description
`label`	String	Display name for record count, or the summarized or custom summary formula field.
`dataType`	String	Data type of the summarized or custom summary formula field.
`acrossGroupingContext`	String	Column grouping in the report where the custom summary formula is displayed. As this example shows in the JSON response and in the custom summary formula editor of the matrix report, the custom summary formula is set at the grand summary level for the columns.

```
{
   "reportExtendedMetadata" : {
      "aggregateColumnInfo" : {
         "FORMULA1" : {
            "label" : "Stalled Oppty Avg",
            "dataType" : "Percent",
            "acrossGroupingContext" :
"GRAND_SUMMARY",
            "downGroupingContext" :
"GRAND_SUMMARY"
         },
      }
   }
}
```

Property	Type	Description
		
`downGroupingContext`	String	Row grouping in the report where the custom summary formula is displayed. In this example, the custom summary formula for a summary report is displayed at the first grouping level This example is shown in both the JSON response and in the custom summary formula editor of the summary report. `{ "reportExtendedMetadata" : { "aggregateColumnInfo" : { ...}, "FORMULA1" : { "label" : "Average Won", "dataType" : "Number", "acrossGroupingContext" : null, "downGroupingContext" : "TYPE" }, } }` 

Grouping column information

Property	Type	Description
`label`	String	Display name of the field or bucket field used for grouping.
`dataType`	String	Data type of the field used for grouping.
`groupingLevel`	Integer	Level of the grouping. Value can be: • `0`, `1`, or `2`. Indicates first, second, or third row level grouping in summary reports. • `0` or `1`. Indicates first or second row or column level grouping in a matrix report.

Report metadata

Property	Type	Description
`name`	String	Report name.
`id`	String	Unique report ID.
`currency`	String	Report currency, such as USD, EUR, GBP, for an organization that has Multi-Currency enabled. Value is `null` if the organization does not have Multi-Currency enabled.
`reportFormat`	String	Format of the report. Value can be: • `TABULAR` • `SUMMARY` • `MATRIX`
`developerName`	String	Report API name.
`reportType`	Report type	Unique API name and display name for the report type. `type`: Of type string, this is the unique identifier of the report type. `label`: Of type string, this is the display name of the report type.
`aggregates`	Array of strings	Unique identities for summary or custom summary formula fields in the report. For example: • `a!Amount` represents the average for the `Amount` column. • `s!Amount` represents the sum of the `Amount` column. • `m!Amount` represents the minimum value of the `Amount` column. • `x!Amount` represents the maximum value of the `Amount` column. • `s!<customfieldID>` represents the sum of a custom field column. For custom fields and custom report types, the identity is a combination of the summary type and the field ID.
`groupingsDown`	Groupings down[]	Unique identities for each row grouping in a report. The identity is: • `BucketField_(ID)` for bucket fields. • ID of a custom field when the custom field is used for grouping.
`groupingsAcross`	Groupings across[]	Unique identities for each column grouping in a report. The identity is: • An empty array for reports in summary format as it can't have column groupings. • `BucketField_(ID)` for bucket fields. • ID of a custom field when the custom field is used for a column grouping.

Property	Type	Description
`reportBooleanFilter`	String	Logic to parse custom field filters. Value is `null` when filter logic is not specified. This is an example of a report filtered to show opportunities for accounts that are either of customer or partner type OR their annual revenue exceeds 100K AND they are medium or large sized businesses. The filters are processed by the logic, "(1 OR 2) AND 3." `{ ... "reportBooleanFilter": "(1 OR 2) AND 3", "reportFilters": [ { "value": "Analyst,Integrator,Press,Other", "column": "TYPE", "operator": "notEqual" }, { "value": "100,000", "column": "SALES", "operator": "greaterThan" }, { "value": "Small", "column": "Size", "operator": "notEqual" } ] } }`
`reportFilters`	Filter details[]	List of each custom filter in the report along with the field name, filter operator, and filter value.
`detailColumns`	Array of strings	Unique API names for the fields that have detailed data.
`historicalSnapshotDates`	Array of strings	List of historical snapshot dates.

Groupings down

Property	Type	Description
`name`	String	API name of the field used as a row grouping.
`sortOrder`	String	Order in which data is sorted within a row grouping. Value can be: • `Asc` for ascending order. • `Desc` for descending order.

Property	Type	Description
`dateGranularity`	String	Interval set on a date field that's used as a row grouping. Value can be: • Day • Calendar Week • Calendar Month • Calendar Quarter • Calendar Year • Fiscal Quarter • Fiscal Year • Calendar Month in Year • Calendar Day in Month
`sortAggregate`	String	Summary field that's used to sort data within a grouping in a report that's in summary format. Applies if you have the Aggregate Sort feature enabled as part of its pilot program. The value is null when data within a grouping is not sorted by a summary field. In this example, data grouped by Account Owner is sorted by the sum of Annual Revenue. `{ "aggregates": ["s!SALES","RowCount"], "groupingsDown": [ { "name": "USERS.NAME", "sortOrder": "Desc", "dateGranularity": "None", "sortAggregate": "s!SALES" } ] }`

Groupings across

Property	Type	Description
`name`	String	API name of the field used as a column grouping.
`sortOrder`	String	Order in which data is sorted within a column grouping. Value can be: • `Asc` for ascending order. • `Desc` for descending order.
`dateGranularity`	String	Interval set on a date field used as a column grouping. Value can be: • Day • Calendar Week

Property	Type	Description
		• Calendar Month • Calendar Quarter • Calendar Year • Fiscal Quarter • Fiscal Year • Calendar Month in Year • Calendar Day in Month

Filter details

Property	Type	Description
`value`	String	Value by which a field is filtered. For example, the field `Age` can be filtered by a numeric value.
`column`	String	Unique API name for the field that's being filtered.
`operator`	String	Unique API name for the condition used to filter a field such as "greater than" or "not equal to." Filter conditions depend on the data type of the field.

Example

See this sample of a GET request for report metadata.

SEE ALSO:

Execute Sync

Execute Async

Execute Sync

Runs a report immediately with or without changing filters and returns the latest summary data with or without details for your level of access.

Resource URL

Data	URL
Summary	`/services/data/<latest API version>/analytics/reports/<report ID>`
Summary with detail	`/services/data/<latest API version>/analytics/reports/<report ID>?includeDetails=true`

Formats

JSON

HTTP Methods

Method	Description
GET	Get report results.
POST	Get specific results by passing filters in the report metadata.

POST Request Body

Property	Type	Description
`name`	String	Report name.
`id`	String	Unique report ID.
`currency`	String	Report currency, such as USD, EUR, GBP, for an organization that has Multi-Currency enabled. Value is `null` if the organization does not have Multi-Currency enabled.
`reportFormat`	String	Format of the report. Value can be: • `TABULAR` • `SUMMARY` • `MATRIX`
`developerName`	String	Report API name.
`reportType`	Report type	Unique API name and display name for the report type. `type`: Of type string, this is the unique identifier of the report type. `label`: Of type string, this is the display name of the report type.

Property	Type	Description
`aggregates`	Array of strings	Unique identities for summary or custom summary formula fields in the report. For example: • `a!Amount` represents the average for the `Amount` column. • `s!Amount` represents the sum of the `Amount` column. • `m!Amount` represents the minimum value of the `Amount` column. • `x!Amount` represents the maximum value of the `Amount` column. • `s!<customfieldID>` represents the sum of a custom field column. For custom fields and custom report types, the identity is a combination of the summary type and the field ID.
`groupingsDown`	Groupings down[]	Unique identities for each row grouping in a report. The identity is: • `BucketField_(ID)` for bucket fields. • ID of a custom field when the custom field is used for grouping.
`groupingsAcross`	Groupings across[]	Unique identities for each column grouping in a report. The identity is: • An empty array for reports in summary format as it can't have column groupings. • `BucketField_(ID)` for bucket fields. • ID of a custom field when the custom field is used for a column grouping.

`reportBooleanFilter`	String	Logic to parse custom field filters. Value is `null` when filter logic is not specified.
		This is an example of a report filtered to show opportunities for accounts that are either of customer or partner type OR their annual revenue exceeds 100K AND they are medium or large sized businesses. The filters are processed by the logic, "(1 OR 2) AND 3."
		`{`
		`...`
		`"reportBooleanFilter": "(1 OR 2) AND 3",`
		`"reportFilters": [`
		`{`
		`"value": "Analyst,Integrator,Press,Other",`
		`"column": "TYPE",`
		`"operator": "notEqual"`
		`},`
		`{`
		`"value": "100,000",`
		`"column": "SALES",`
		`"operator": "greaterThan"`
		`},`

Property	Type	Description

		<pre>{
		"value": "Small",
		"column": "Size",
		"operator": "notEqual"
		}
		]
		}
		}</pre>

`reportFilters`	Filter details[]	List of each custom filter in the report along with the field name, filter operator, and filter value.
`detailColumns`	Array of strings	Unique API names for the fields that have detailed data.
`historicalSnapshotDates`	Array of strings	List of historical snapshot dates.

Response Body

Property	Type	Description
`hasDetailRows`	Boolean	When `true`,the fact map returns values for both summary level and record level data. When `false`, the fact map returns summary values.
`attributes`	Attributes	Key report attributes and child resource URLs.
`reportMetadata`	Report metadata	Unique identifiers for groupings and summaries.
`reportExtendedMetadata`	Report extended metadata	Additional information about columns, summaries, and groupings.
`groupingsDown`	Groupings down	Collection of row groupings, keys, and their values.
`groupingsAcross`	Groupings across	Collection of column groupings, keys, and their values.
`allData`	Boolean	When `True`, all report results are returned. When `False`, results are returned for the same number of rows as a report run in Salesforce. Note: For reports that have too many records, use filters to refine results.
`factMap`	Fact map	Summary level data or both summary and detailed data for each row or column grouping. Detailed data is available if `hasDetailRows` is `true`. Each row or column grouping is represented by combination of row and column grouping keys defined in Groupings down and Groupings across.

Attributes

Property	Type	Description
`describeUrl`	String	Resource URL to get report metadata.
`instancesUrl`	String	Resource URL to run a report asynchronously. The report can be run with or without filters to get summary or both summary and detailed data. Results of each instance of the report run are stored under this URL.
`type`	String	API resource format.
`reportName`	String	Display name of the report.
`reportId`	String	Unique report ID.

Groupings down

Property	Type	Description
`groupings`	Groupings[]	Information for each row grouping as a list.

Groupings across

Property	Type	Description
`groupings`	Groupings[]	Information for each column grouping as a list.

Groupings

Property	Type	Description
`value`	String	Value of the field used as a row or column grouping. The value depends on the field's data type. • Currency fields: – `amount`: Of type currency. Value of a data cell. – `currency`: Of type picklist. The ISO 4217 currency code, if available; for example, USD for US dollars or CNY for Chinese yuan. (If the grouping is on the converted currency, this is the currency code for the report and not for the record.) • Picklist fields: API name. For example, a custom picklist field, `Type of Business` with values 1, 2, 3 for Consulting, Services, and Add-On Business, has `1`, `2`, or `3` as the grouping value. • ID fields: API name.

Property	Type	Description
		• Record type fields: API name. • Date and time fields: Date or time in ISO-8601 format. • Lookup fields: Unique API name. For example, for the `Opportunity Owner` lookup field, the ID of each opportunity owner's Chatter profile page can be a grouping value.
`key`	String	Unique identity for a row or column grouping. The identity is used by the fact map to specify data values within each grouping.
`label`	String	Display name of a row or column grouping. For date and time fields, the label is the localized date or time.
`groupings`	Array	Second or third level row or column groupings. If there are none, the value is an empty array.
`dategroupings`	Array	Start date and end date of the interval defined by `date granularity`.

Fact map

Property	Type	Description
`rows`	Data cells[]	Array of detailed report data listed in the order of the detail columns provided by the report metadata.
`aggregates`	Aggregates[]	Summary level data including record count for a report.

Data cells

Property	Type	Description
`value`	Detail column info data type	The value of a specified cell.
`label`	String	Display name of the value as it appears for a specified cell in the report.

Aggregates

Property	Type	Description
`value`	Number	Numeric value of the summary data for a specified cell.
`label`	String	Formatted summary data for a specified cell.

Examples

- See this sample of a GET request to fetch report data.
- See this sample of a POST request with filters to get specific report results on the fly.
- See these examples of fact map keys that represent report data from the user interface.

SEE ALSO:

Describe

Execute Async

Execute Async

Runs an instance of a report asynchronously with or without filters and returns a handle that stores the results of the run. The results can contain summary data with or without details.

Resource URL

Data	URL
Summary	`/services/data/<latest API version>/analytics/reports/<report ID>/instances`
Summary with detail	`/services/data/<latest API version>/analytics/reports/<report ID>/instances?includeDetails=true`

Formats

JSON

HTTP methods

POST

POST request body

Property	Type	Description
`name`	String	Report name.
`id`	String	Unique report ID.
`currency`	String	Report currency, such as USD, EUR, GBP, for an organization that has Multi-Currency enabled. Value is `null` if the organization does not have Multi-Currency enabled.

Property	Type	Description
`reportFormat`	String	Format of the report. Value can be: • `TABULAR` • `SUMMARY` • `MATRIX`
`developerName`	String	Report API name.
`reportType`	Report type	Unique API name and display name for the report type. `type`: Of type string, this is the unique identifier of the report type. `label`: Of type string, this is the display name of the report type.
`aggregates`	Array of strings	Unique identities for summary or custom summary formula fields in the report. For example: • `a!Amount` represents the average for the `Amount` column. • `s!Amount` represents the sum of the `Amount` column. • `m!Amount` represents the minimum value of the `Amount` column. • `x!Amount` represents the maximum value of the `Amount` column. • `s!<customfieldID>` represents the sum of a custom field column. For custom fields and custom report types, the identity is a combination of the summary type and the field ID.
`groupingsDown`	Groupings down[]	Unique identities for each row grouping in a report. The identity is: • `BucketField_(ID)` for bucket fields. • ID of a custom field when the custom field is used for grouping.
`groupingsAcross`	Groupings across[]	Unique identities for each column grouping in a report. The identity is: • An empty array for reports in summary format as it can't have column groupings. • `BucketField_(ID)` for bucket fields. • ID of a custom field when the custom field is used for a column grouping.

`reportBooleanFilter`	String	Logic to parse custom field filters. Value is `null` when filter logic is not specified.
		This is an example of a report filtered to show opportunities for accounts that are either of customer or partner type OR their annual revenue exceeds 100K AND they are medium or large sized businesses. The filters are processed by the logic, "(1 OR 2) AND 3."
		`{`
		`...`
		`"reportBooleanFilter": "(1 OR 2) AND`

Property	Type	Description

		<pre>3",
		"reportFilters": [
		{
		"value":
		"Analyst,Integrator,Press,Other",
		"column": "TYPE",
		"operator": "notEqual"
		},
		{
		"value": "100,000",
		"column": "SALES",
		"operator": "greaterThan"
		},
		{
		"value": "Small",
		"column": "Size",
		"operator": "notEqual"
		}
		]
		}
		}</pre>

`reportFilters`	Filter details[]	List of each custom filter in the report along with the field name, filter operator, and filter value.
`detailColumns`	Array of strings	Unique API names for the fields that have detailed data.
`historicalSnapshotDates`	Array of strings	List of historical snapshot dates.

Response body

Property	Type	Description
`id`	String	Unique ID for an instance of a report that was run asynchronously.
`status`	String	• `New` if the report run has just been triggered through a request. • `Success` if the report ran. • `Running` if the report is being run. • `Error` if the report run failed. The instance of a report run can return an error if, for example, your permission to access the report has been removed since you requested the run.
`url`	String	URL where results of the report run for that instance are stored. The value is `null` if the report couldn't be run because of an error.
`ownerId`	String	API name for the user ID.

Property	Type	Description
`completionDate`	Date, time string	Date, time when the instance of the report run finished. Only available if the report instance ran successfully or couldn't be run because of an error. Date-time information is in ISO-8601 format.
`hasDetailRows`	Boolean	• When `false`, indicates that summary level data was requested for the report instance. • When `true`, indicates that detailed data, which includes summary level data, was requested for the report instance.
`requestDate`	Date, time string	Date and time when an instance of the report run was requested. Date-time information is in ISO-8601 format.

Examples

- See a sample of a GET request to return a list of asynchronous runs of a report.
- See a sample of a POST request to request an asynchronous report run.

SEE ALSO:

Describe

Execute Sync

Instances List

Returns a list of instances for a report that you requested to be run asynchronously. Each item in the list is treated as a separate instance of the report run with metadata in that snapshot of time.

Resource URL

```
/services/data/<latest API version>/analytics/reports/<report ID>/instances
```

Formats

JSON

HTTP methods

GET

Response body

Property	Type	Description
`id`	String	Unique ID for a report instance that was requested for a run. The ID is used to obtain results of the report run for that instance.
`status`	String	• `New` if the report run has just been triggered through a POST request. • `Success` if the report ran. • `Running` if the report is being run. • `Error` if the report run failed. The instance of a report run can return an error if, for example, your permission to access the report has been removed since you requested the run.
`url`	String	URL where results of the report run for that instance are stored. The value is `null` if the report couldn't be run because of an error.
`ownerId`	String	API name for the user ID.
`hasDetailRows`	Boolean	• When `false`, indicates that summary level data was requested for the report run. • When `true`, indicates that detailed data, which includes summary level data, was requested for the report run.
`completionDate`	Date, time string	Date, time when the instance of the report run finished. Only available if the report instance ran successfully or couldn't be run because of an error. Date-time information is in ISO-8601 format.
`requestDate`	Date, time string	Date and time when an instance of the report run was requested. Date-time information is in ISO-8601 format.

Examples

See a sample of a GET request to return a list of asynchronous runs of a report.

SEE ALSO:

- Execute Async
- Instance Results

Instance Results

Retrieves results for an instance of a report run asynchronously with or without filters. Depending on your asynchronous report run request, data can be at the summary level or include details.

Resource URL

```
/services/data/<latest API version>/analytics/reports/<report ID>/instances/<instance ID>
```

Formats

JSON

HTTP methods

GET

Response body

Property	Type	Description
`hasDetailRows`	Boolean	• When `false`, report results are at summary level. • When `true`, report results are at the record detail level.
`allData`	Boolean	When `True`, all report results are returned. When `False`, detailed data for the first 2000 report rows are returned.
`reportMetadata`	Report metadata	Information about the fields used to build the report.
`factMap`	Fact map	Collection of summary level data or both detailed and summary level data.
`attributes`	Attributes	Attributes for the instance of the report run.
`reportExtendedMetadata`	Report extended metadata	Information on report groupings, summary fields, and detailed data columns, which is available if detailed data is requested.
`groupingsDown`	Groupings down	Collection of row groupings.
`groupingsAcross`	Groupings across	Collection of column groupings.

Attributes

Property	Type	Description
`id`	String	Unique ID for an instance of a report that was run.
`status`	String	• `New` if the report run has just been triggered through a request. • `Success` if the report ran. • `Running` if the report is being run.

Property	Type	Description
		• `Error` if the report run failed. The instance of a report run can return an error if, for example, your permission to access the report has been removed since you requested the run.
`ownerId`	String	API name for the user ID.
`completionDate`	Date, time string	Date, time when the instance of the report run finished. Only available if the report instance ran successfully or couldn't be run because of an error. Date-time information is in ISO-8601 format.
`requestDate`	Date, time string	Date and time when an instance of the report run was requested. Date-time information is in ISO-8601 format.
`type`	String	Format of the resource.
`reportId`	String	Unique report ID.
`reportName`	String	Display name of the report.

Example

See a sample of a GET request to fetch results of an asynchronous report run.

SEE ALSO:

- Execute Async
- Instances List

List

Displays a list of up to 200 tabular, matrix, or summary reports that you recently viewed. To get a full list of reports by format, name, and other fields, use a SOQL query on the Report object.

Resource

```
/services/data/<latest API version>/analytics/reports
```

Formats

JSON

HTTP methods

GET

Response body

Property	Type	Description
`name`	String	Report display name.
`id`	String	Unique report ID.
`url`	String	URL that returns report data.
`describeUrl`	String	URL that retrieves report metadata.
`instancesUrl`	String	Information for each instance of the report that was run asynchronously.

Example

See a sample GET request.

CHAPTER 5 Dashboards API Resource Reference

In this chapter ...

The Dashboards API provides several resources for accessing and refreshing dashboards.

Resources for the Dashboards API are available at `/services/data/<latest API version>/analytics/dashboards`. You can query each resource with an HTTP method (such as GET). Use these resources to integrate dashboard data directly into your applications.

Resource	Supported HTTP Method	Description
Dashboard List	GET	Returns a list of recently used dashboards.
Dashboard Results	GET	Returns the metadata, data, and status for the specified dashboard.
	PUT	Triggers a dashboard refresh.
Dashboard Status	GET	Returns the status for the specified dashboard.

Dashboard List

Returns a list of recently used dashboards.

Syntax

URI

`/vXX.X/analytics/dashboards`

Available since release

31.0

Formats

JSON

HTTP methods

GET

Authentication

`Authorization: Bearer token`

Response body

An array of recent dashboard objects. Each object contains the following fields:

Property	Type	Description
`id`	String	Unique identifier of the dashboard.
`name`	String	Localized display name of the dashboard.
`statusUrl`	String	Dashboard status URL.
`url`	String	Dashboard result URL.

Dashboard Results

Returns metadata, data, and status for the specified dashboard using the GET method. Can also trigger a dashboard refresh using the PUT method.

Syntax

URI

`/vXX.X/analytics/dashboards/dashboardID`

Or, with optional parameters:

`/vXX.X/analytics/dashboards/dashboardID`
`?runningUser=runningUserID&filter1=filter1ID&filter2=filter2ID&filter3=filter3ID`

Available since release

31.0

Formats

JSON

HTTP methods

GET, PUT

Authentication

```
Authorization: Bearer token
```

Parameters

The following optional parameters can be used with the GET and PUT methods:

Parameter Name	Description
`runningUser`	Identifier of the running user. Gives an error if the user is not allowed to change the running user, or if the selected running user is invalid.
`filter1`	Identifier of the selected filter option for the first filter. Gives an error if the filter option is invalid.
`filter2`	Identifier of the selected filter option for the second filter. Gives an error if the filter option is invalid.
`filter3`	Identifier of the selected filter option for the third filter. Gives an error if the filter option is invalid.

Response body

Property	Type	Description
`componentData`	Component data[]	Ordered array containing data and status for each component of the dashboard.
`dashboardMetadata`	Dashboard metadata	Metadata for the entire dashboard.

Component data

Property	Type	Description
`componentId`	String	Unique identifier of the component.
`reportResult`	Report results	Report metadata and summary data for the dashboard component. Uses the same data format as the Report API.
`status`	Component status	Queue and data status of the component.

Component status

Property	Type	Description
`dataStatus`	String	Status of the data set of the component. Value can be: • `NODATA`: The data set was never generated or is invalid due to a change in the report.

Property	Type	Description
		• `DATA`: The data set is available and was last refreshed at the `refreshDate`. • `ERROR`: A component error has occurred. Details can be found in `errorCode`, `errorMessage`, and `errorSeverity`.
`errorCode`	String	Unique identifier of error message. This property is only populated in case of error.
`errorMessage`	String	Localized error message. This property is only populated in case of error.
`errorSeverity`	String	Severity of error code and message. Value can be: • `Error` • `Warning` This property is only populated in case of error.
`refreshDate`	Date and time string	Date and time of last refresh in ISO-8601 format.
`refreshStatus`	String	Refresh status of the component. Value can be: • `IDLE`: The component is not currently being refreshed. • `RUNNING`: The component is currently being refreshed.

Dashboard metadata

Property	Type	Description
`attributes`	Attributes	Attributes for the dashboard resource, such as name, identifier, and references to other related resources.
`canChangeRunningUser`	Boolean	Indicates whether the user is allowed to select a specific running user. Always `true` for team dashboards.
`components`	Component[]	Ordered array of components in this dashboard.
`description`	String	Dashboard description.
`developerName`	String	Unique API name of the dashboard.
`filters`	Filter[]	Ordered array of filters for this dashboard. The dashboard can have 0-3 filters.
`id`	String	Unique identifier of dashboard.
`layout`	Layout	Component layout for this dashboard.
`name`	String	Dashboard name.
`runningUser`	Running user	The running user, which is either specified at dashboard design time, or is overridden by the `runningUser` parameter specified in the GET request. For dynamic dashboards, this is always the current user.

Attributes

Property	Type	Description
`dashboardId`	String	Unique identifier of dashboard.
`dashboardName`	String	Dashboard name.
`statusUrl`	Url	The URL of the status resource for the dashboard.
`type`	String	This property is always set to `Dashboard`.

Component

Property	Type	Description
`componentData`	Integer	Index into the component data array in the response body.
`footer`	String	Footer of the component.
`header`	String	Header of the component.
`id`	String	Unique identifier of the component.
`properties`	Properties (for Report component type) Properties (for Visualforce page component type)	Component properties, including type-specific visualization properties.
`reportId`	String	Unique identifier of the underlying report.
`title`	String	Title of the component
`type`	String	Type of the component. Value can be: • `Report` • `VisualforcePage` If the component is an SControl, the value is not set.

Filter

Property	Type	Description
`name`	String	Localized display name of filter.
`options`	Filter option	Ordered array of possible filter options.
`selectedOption`	Integer	Index of the selected option from the `options` array. This matches the selection that was made based on the `filter1`, `filter2`, or `filter3` parameter. Value is `null` if no option is selected.

Filter option

Property	Type	Description
`alias`	String	Optional alias of the filter option.
`id`	String	Unique identifier of the filter option. Used as a value for the `filter1`, `filter2`, and `filter3` parameters.
`operation`	String	Unique API name for the filter operation. Valid filter operations depend on the data type of the filter field. Value can be: • `equals` • `notEqual` • `lessThan` • `greaterThan` • `lessOrEqual` • `greaterOrEqual` • `contains` • `notContain` • `startsWith` • `includes` • `excludes` • `within` • `between`
`value`	String	Value to filter on. Used for all operations except `between`.
`startValue`	String	Start value when using a `between` operation. Not set for all other operations.
`endValue`	String	End value when using a `between` operation. Not set for all other operations.

Layout

Property	Type	Description
`columns`	Column[]	Dashboard layout columns. Can have 2 or 3 columns, including empty columns.

Column

Property	Type	Description
`components`	Integer[]	Ordered list of components in a column (top to bottom). Components are represented by indices into the array of components in the dashboard metadata object.

Running user

Property	Type	Description
`displayName`	String	Display name of running user.
`id`	String	Unique identifier of running user.

Properties (for Report component type)

Property	Type	Description
`aggregateName`	String	Primary aggregate developer name for the component. Value is `null` for scatter charts.
`maxRows`	Number	Maximum number of rows to be rendered, based on the sort value.
`sort`	Sort	Sorting information for the component.
`visualizationProperties`	Visualization properties (Table) Visualization properties (Metric) Visualization properties (Gauge)	Type-specific visualization properties.
`visualizationType`	String	Type of the component. Value can be: • `Bar` • `Column` • `Donut` • `Funnel` • `Gauge` • `Line` • `Metric` • `Pie` • `Scatter` • `Table`

Visualization properties (Table)

Property	Type	Description
`tableColumns`	Table column[]	Columns of the table component.
`breakPoints`	Break point[]	Break points for the table component.

Visualization properties (Metric)

Property	Type	Description
`metricLabel`	String	Label for the metric component.
`breakPoints`	Break point[]	Break points for the metric component.

Visualization properties (Gauge)

Property	Type	Description
`breakPoints`	Break point[]	Break points for the gauge component.

Sort

Property	Type	Description
`column`	String	Developer name for a sorted column.
`sortOrder`	String	Sort order. Value can be: • `asc` • `desc`

Break point

Property	Type	Description
`aggregateName`	String	Aggregate column developer name that the break points have been applied to.
`breaks`	Break[]	Break values for a break point.

Break

Property	Type	Description
`color`	String	A hex value representing the color for the break point. Note: A color value of black displays only 1 character (0) instead of 6 characters (000000).
`lowerBound`	Number	Lower bound for the break point.
`upperBound`	Number	Upper bound for the break point.

Table column

Property	Type	Description
`column`	String	Developer name for the aggregate or grouping column.
`isPercent`	Boolean	Indicates whether the column value is shown as a percent.
`scale`	Number	The number of decimal places for the column value.
`showTotal`	Boolean	Indicates whether the column shows the total.
`type`	String	Type of the column. Value can be: • `aggregate` • `grouping`

Properties (for Visualforce page component type)

Property	Type	Description
`pageName`	String	Developer name of the Visualforce page.
`height`	String	Height of the Visualforce page, in pixels.

Refresh Response body

Property	Type	Description
`statusUrl`	String	URL of the status resource for the dashboard.

Dashboard Status

Returns the status for the specified dashboard.

Syntax

URI

`/vXX.X/analytics/dashboards/`***`dashboardID`***`/status`

Or, with optional parameters:

`/vXX.X/analytics/dashboards/`***`dashboardID`***`/status`
`?runningUser=`***`runningUserID`***`&filter1=`***`filter1ID`***`&filter2=`***`filter2ID`***`&filter3=`***`filter3ID`***

Available since release

31.0

Formats

JSON

HTTP methods

GET

Authentication

```
Authorization: Bearer token
```

Parameters

The following optional parameters can be used with the GET method:

Parameter Name	Description
`runningUser`	ID of the running user. Gives an error if the user is not allowed to change the running user, or if the selected running user is invalid.
`filter1`	ID of the selected filter option for the first filter. Gives an error if the filter option is invalid.
`filter2`	ID of the selected filter option for the second filter. Gives an error if the filter option is invalid.
`filter3`	ID of the selected filter option for the third filter. Gives an error if the filter option is invalid.

Response body

Property	Type	Description
`componentStatus`	Component status with id[]	Status for each component of the dashboard. The order of the array is the same as in previous calls, unless the dashboard has changed in the meantime.

Component status with id

Property	Type	Description
`componentId`	String	Unique ID of the dashboard component.
`refreshDate`	Date and time string	Date and time of last refresh in ISO-8601 format.
`refreshStatus`	String	Refresh status of the component. Value can be: • `IDLE`: The component is not currently being refreshed. • `RUNNING`: The component is currently being refreshed.

Dashboard and Component Error Codes

Errors can occur at the dashboard level and at the component level.

Dashboard-level error messages are returned in the response header, and component-level error messages are returned as part of the component status object.

Dashboard-level errors

When a dashboard-level error occurs, the response header contains an HTTP response code and one of the following error messages:

HTTP Response Code	Error Message
400	The running user for this dashboard doesn't have permission to run reports. Your system administrator should select a different running user for this dashboard.
400	The running user for this dashboard is inactive. Your system administrator should select an active user for this dashboard.
400	You don't have permission to view data as this user.
400	Your organization has reached the limit for dynamic dashboards, or doesn't have access. Ask your administrator to enable dynamic dashboards or convert them to dashboards with a specific running user.
400	The selected filter item isn't valid.
400	You can't refresh this dashboard. A refresh is already in progress.

Component-level errors

If an error occurs at the component level, the `errorCode`, `errorMessage`, and `errorSeverity` properties of the component status field are populated. The `errorSeverity` property distinguishes between errors and warnings. Errors are blocking issues that prevent the query from returning any data. Warnings are non-blocking issues; queries will finish, but they might return incomplete data. The following table shows the possible values for the error fields.

errorCode	errorMessage	errorSeverity
201	This component must have a type and a data source.	Error
202	The source report isn't available; it's been deleted or isn't in a folder accessible to the dashboard's running user.	Error
203	This report can no longer be edited or run. Your administrator has disabled all reports for the custom object, or its relationships have changed.	Error
205	The source report is based on a report type that is inaccessible to the dashboard's running user.	Error
208	Unable to run source report because its definition is invalid.	Error
209	This report cannot be used as the source for this component. If it is a summary or matrix report, add one or more groupings in the report. If it is a tabular report with a row limit, specify the Dashboard Settings in the report.	Error
210	This row-limited tabular report cannot be used as the source for this component. Use the dashboard component editor to specify the data you want to display, or specify the Dashboard Settings in the report.	Error

errorCode	errorMessage	errorSeverity
211	To use this row-limited tabular report as the source, edit the report and specify the Name and Value under Dashboard Settings. When updating the report, make sure you are the running user of the dashboard.	Error
212	Groupings and combination charts are not available for a row-limited tabular report. Set “Group By” to None and deselect “Plot Additional Values.”	Error
300	The results below may be incomplete because the underlying report produced too many summary rows, and the sort order of the component is different from the sort order in the underlying report. Try adding filters to the report to reduce the number of rows returned.	Error
301	Results may be incomplete because the source report had too many summary rows. Try filtering the report to reduce the number of rows returned.	Warning
302	The component can't be displayed because the source report exceeded the time limit.	Warning
303	The component can't be displayed because the source report failed to run.	Error
304	The component can't be displayed because the dashboard filter raises the number of source report filters above the limit. Reduce the number of report filters and try again.	Error
305	The component can't be displayed because the field(s) you chose for the filter are unavailable.	Error
308	You can’t filter this component because data is in the joined report format. To filter the component, change its report format.	Error
309	The underlying report uses a snapshot date that is out of range.	Error

INDEX