Webtrekk
Drive Your Profit.

Webtrekk JSON/RPC API

Last update: May 24, 2018

1 Content

2 General

Webtrekk JSON/RPC API gives you the opportunity to automatically recall data from Webtrekk. The connection with the Webtrekk JSON/RPC API takes place via HTTP or HTTPS protocols.

URL: http(s)://report2.webtrekk.de/cgi-bin/wt/JSONRPC.cgi

Methods

Following methods are available by the JSON/RPC API:

Method	Description
getConnectionTest	Tests the connection.
login	Logs into an account and creates a token.
logout	Logs out of an account deletes a token.
getAccountList	Lists all accounts available for a login.
getAnalysisObjectsAndMetricsList	Lists all available analysis objects and metrics for an account.
getCustomReportsList	Lists all available reports for an account.
getDynamicTimeIntervalList	Lists all available dynamic time intervals for an account.
getAnalysisData	Gets output data of analyses.
getReportData	Gets output data of reports.
exportData	Exports category / parameter data.
importData	Imports category / parameter data.
deleteTvData	Initiates the deletion of TV data.

Webtrekk
Drive Your Profit.

Errors

Errors are received in the parameter “error” in the result.

Example:

```
{
	"version": "1.1",
	"error": {
			"name": "JSONRPCError",
			"message": "Authorization failed",
			"code": 501
	}
}
```

3 Connection Test

In order to test the connection, you simply use the method „getConnectionTest“. The method works without any login data or other parameters. The string „Connection successful“ obtains the test result.

Request

Parameter	Description
none	

Example:

```
{
  "params": {},
  "version": "1.1",
  "method": "getConnectionTest"
}
```

Response

Parameter	Type	Description
	string	Submits string „Connection successful“

Example:

```
{
  "version": "1.1",
  "result": "Connection successful"
}
```

4 Login

By using the method „login“ you retrieve a token to use in following requests. The token is valid until you logout if you use it at least once per hour. If you don't, the token becomes invalid.

Request

Parameter	Optional	Description
customerId	no	ID of wanted customer (Track-ID)
login	no	Login Webtrekk Frontend
pass	no	Password Webtrekk Frontend
language	yes	Language (‚de‘, ‚en‘, ‚fr‘, ‚es‘, ‚cn‘)

Example:

```
{
    "params": {
        "login": "login",
        "pass": "pass",
        "customerId": "11111111111",
        "language": "en"
    },
    "version": "1.1",
    "method": "login"
}
```

Response

The response of the message is the token.

Example:

```
{
    "version": "1.1",
    "result": "98991ec3459031c4bd9ddfeb4e142045_52b6b54cbb4e9654d1ff032521a86270"
}
```

5 Logout

By using the method „login“ you delete a token.

Request

Parameter	Optional	Description
token	no	Token

Example:

```
{
    "params": {
        "token": "98991ec3459031c4bd9ddfeb4e142045_52b6b54cbb4e9654d1ff032521a86270"
    },
    "version": "1.1",
    "method": "logout"
}
```

Response

The result is “1” if the logout was successful, otherwise “0”.

Example:

```
{
    "version": "1.1",
    "result": "1"
}
```

6 Account List

By using the method „getAccount“ you retrieve information about all available Accounts for a login.

Request

Parameter	Optional	Description
login	no	Login Webtrekk Frontend
pass	no	Password Webtrekk Frontend

Example:

```
{
    "params": {
        "login": "login",
        "pass": "pass"
    },
    "version": "1.1",
    "method": "getAccountList"
}
```

Response

The result is an array of account information.

Example:

```
{
    "version": "1.1",
    "result": [
        {
            "customerId": "11111111111",
            "title": "Account 1"
        },
        {
            "customerId": "11111111112",
            "title": "Account 2"
        }
    ]
}
```

7 Analysis Objects and Metrics List

By using the method „getAnalysisObjectsAndMetricsList“ you retrieve information about all available analysis objects and metrics in an account.

Request

Parameter	Optional	Description
token	no	Token (Parameters login, pass, customerId are also possible)

Example:

```
{
    "params": {
        "token": "98991ec3459031c4bd9ddfeb4e142045_52b6b54cbb4e9654d1ff032521a86270"
    },
    "version": "1.1",
    "method": "getAnalysisObjectsAndMetricsList"
}
```

Response

The results are three arrays of Analysis Objects, Metrics and Customer Scores

Example:

```
{
    "version": "1.1",
    "result": {
        "analysisObjects": [
            "Plugins",
            "Provider",
            "Region",
            "Pages",
        ]
        "metrics": [
            "Page Impressions",
            "Visitors",
            "Visits",
        ],
```

```
            "customMetrics": [
                "Engagement",
                "Page Impressions on index pages",
                "Visits Google",
            ],
        }
    }
```

8 Custom Reports List

By using the method „getCustomReportsList“ you retrieve information about all available reports in an account.

Request

Parameter	Optional	Description
token	no	Token (Parameters login, pass, customerId are also possible)

Example:

```
{
    "params": {
        "token": "98991ec3459031c4bd9ddfeb4e142045_52b6b54cbb4e9654d1ff032521a86270"
    },
    "version": "1.1",
    "method": " getCustomReportsList"
}
```

Response

The result is an array of all reports with their report groups.

Example:

```
{
    "version": "1.1",
    "result": [
        {
            "reportGroup": "Dashboards",
            "title": "E-Commerce Dashboard"
        },
        {
            "reportGroup": "",
            "title": "Clickstream"
        },
    ]
}
```

9 Dynamic Time Interval List

By using the method „getDynamicTimeIntervalList“ you retrieve information about all available dynamic time intervals in an account.

Request

Parameter	Optional	Description
token	no	Token (Parameters login, pass, customerId are also possible)

Example:

```
{
    "params": {
        "token": "98991ec3459031c4bd9ddfeb4e142045_52b6b54cbb4e9654d1ff032521a86270"
    },
    "version": "1.1",
    "method": "getDynamicTimeIntervalList"
}
```

Response

The result is an array of all dynamic time intervals and the codes to use in getAnalysisData function.

Example:

```
{
    "version": "1.1",
    "result": [
        {
            "name": "Today",
            "code": "today"
        },
        {
            "name": "Yesterday",
            "code": "yesterday"
        },
    ]
}
```

10 Analysis Requests

By using the method „getAnalysisData“ you retrieve data of various analyses.

Request

Parameter	Optional	Description
token	no	Token (Parameters login, pass, customerId are also possible)
analysisConfig	no	See chapter “analysisConfig”

analysisConfig

Parameter	Optional	Description
analysisObjects	no	Array of analysisObjects. See chapter “analysisObject”
metrics	yes	Array of metrics. See chapter “metric”
analysisFilter	yes	Filter. See chapter “filter”
startTime	yes	Start time. Format YYYY-MM-TT or YYYY-MM-TT hh:mm:ss.
stopTime	yes	Stop time. Format YYYY-MM-TT or YYYY-MM-TT hh:mm:ss.
dynamicTime	yes	See getDynamicTimeIntervalList. Overwrites startTime and stopTime. Default is the account setting.
startRow	yes	Start row. Default is 1.
rowLimit	yes	Row limit per level. Default is the account setting. If hideFooters is active, the row limit is the total number of rows.
hideFooters	yes	Hides footer rows. Boolean (‘0’, ‘1’). Default is ‘0’.

footerIdentifier	yes	The string that identifies a footer row. Default is '-'.
forceRawData	yes	Force calculation on raw data. Boolean ('0', '1'). Default is '0'.
scope	yes	Scope of a metric. Values are 'auto', 'visit', and 'strong' (as in the frontend). Default value is 'auto'.

analysisObject

Parameter	Optional	Description
title	no	Title of the analysis object. See getAnalysisObjectsAndMetricsList.
sortOrder	yes	Sort order. Values are 'asc' (ascending) or 'desc' (descending) or '' (none).
alias	yes	Alias (e.g. 'Page URL' for 'Pages'). Default is ''.
rowLimit	yes	Row limit per level. Overwrites rowLimit of analysisConfig. Will be ignored if not set in all analysis objects. Ignores startRow in analysisConfig, startRow will be set to 1.

metric

Parameter	Optional	Description
title	no	Title of metric. See getAnalysisObjectsAndMetricsList.
sortOrder	yes	Sort order. Values are 'asc' (ascending) or 'desc' (descending) or '' (none).
metricFilter	yes	Filter. See chapter "filter"
objectScope	yes	Object scope. Boolean ('0', '1'). Default is '1'.

filter

Parameter	Optional	Description
filterRules	no	Array of filter rules. See chapter “filterRule”

filterRule

Parameter	Optional	Description
objectTitle	no	Title of the analysis object or metric. See getAnalysisObjectsAndMetricsList.
filter	no	Search phrase.
filter2	yes	Additional search phrase. Needed for “between”-filters.
link	yes	Link. Values are ‘and’, ‘or’, ‘’. Default is ‘and’.
comparator	no	Comparison predicate. Values for analysis objects are ‘=’, ‘!=’. Values for metrics are ‘=’, ‘<’, ‘>’, ‘between’
caseSens	yes	Case sensitivity. Boolean (‘0’, ‘1’). Default is ‘0’.
scope	yes	Context. Values are ‘visitor’, ‘visit’, ‘page’, ‘action’, ‘’. Default is ‘’ (automatic).

Example:


```
{
    "params": {
        "token": "98991ec3459031c4bd9ddfeb4e142045_52b6b54cbb4e9654d1ff032521a86270",
        "analysisConfig": {
            "hideFooters": 1,
            "analysisFilter": {
                "filterRules": [{
                    "objectTitle": "Pages",
                    "comparator": "=",
                    "filter": "*manieren*",
                    "scope": "visit"
                },
                {
                    "link": "and",
                    "objectTitle": "Browser",
                    "comparator": "!=",
                    "filter": "*Chrome*"
                }]
            },
            "metrics": [{
                "sortOrder": "desc",
                "title": "Page Impressions"
            },
            {
                "title": "Visits",
                "metricFilter": {
                    "filterRules": [{
                            "objectTitle": "Pages",
                            "comparator": "=",
                            "filter": "*index*"
                        }]
                }
            }],
            "rowLimit": 10000,
            "analysisObjects": [{
                "title": "Pages"
            },
            {
                "sortOrder": "asc",
                "title": "Days"
            }]
        }
    },
    "version": "1.1",
     "method": "getAnalysisData"
}
```

Response

Parameter	Description
analysisData	Analysis data as 2 dimensional array
analysisTabHead	Table head
analysisTabFoot	Table foot
totalRowCount	Number of possible rows whereas not all rows have to be read out.
analysisWarnings	Array of warnings given by the system (strings) (e.g. missing rights)
dailyUnique	Contains an advice if the analysis includes only metrics that are calculated uniquely per day
calculationTime	Time of calculation (format YYYY-MM-DD HH:mm:ss)
timeStart	Start time (format YYYY-MM-DD HH:mm:ss)
timeStop	Stop time (format YYYY-MM-DD HH:mm:ss)
analysisTitle	Analysis title
analysisType	“analysis”, “scorecard”, “status”, “pivot”
scorecardData	Analyses of type “scorecard” may have this additional parameter (see Example for analysisType “scorecard”)

Please note: If the analysis object is a time object the “total” values in responses are unique values (the analysis type “analysis” switches to “pivot” in the JSON response).

Example for analysisType „analysis“:


```
{
    "version": "1.1",
    "result": {
        "analysisWarnings": [],
        "dailyUnique": 0,
        "totalRowCount": 1504,
        "analysisTabHead": [
            {
                "dataType": "string",
                "title": "Seiten",
                "contentType": "main"
            },
            {
                "dataType": "number",
                "title": "Page Impressions",
                "contentType": "normal"
            },
            {
                "dataType": "number",
                "title": "% Page Impressions",
                "contentType": "percent"
            }
        ],
        "timeStop": "2011-01-20 23:59:59",
        "calculationTime": "2011-01-20 12:12:22",
        "analysisTabFoot": [
            [
                "",
                "Gesamt",
                "211165",
                "100.00"
            ]
        ],
        "analysisData": [
            [
                "1",
                "manieren_per_mausklick",
                "17213",
                "8.15"
            ],
            [
                "2",
                "index",
                "10000",
                "4.74"
            ],
        ],
        "timeStart": "2010-12-21 00:00:00",
        "analysisTitle": "Seiten",
        "chartType": "bars",
        "analysisType": "analysis"
    }
}
```

Example for analysisType „scorecard“:

There is an additional hashtable “scorecardData” which has data for the visualization of the scorecard and has the parameters “analysisTabHead” and “analysisData” and also the scorecardValue, which is the value presented next to the visualization.


```
{
    "version": "1.1",
    "result": {
        "analysisWarnings": [

        ],
        "scorecardData": {
            "analysisTabHead": [
                {
                    "dataType": "number",
                    "title": "Date",
                    "contentType": "normal"
                },
                {
                    "dataType": "number",
                    "title": "Score",
                    "contentType": "normal"
                }
            ],
            "scorecardValue": "8",
            "analysisData": [
                [
                    "29.12.2010",
                    "5"
                ],
                [
                    "30.12.2010",
                    "5"
                ],
                [
                    "31.12.2010",
                    "5"
                ],
                [
                    "01.01.2011",
                    "5"
                ],
                [
                    "02.01.2011",
                    "5"
                ],
                [
                    "03.01.2011",
                    "4"
                ],
                [
                    "04.01.2011",
                    "5"
                ],
```

```
[
    "05.01.2011",
    "5"
],
[
    "06.01.2011",
    "5"
],
[
    "07.01.2011",
    "4"
],
[
    "08.01.2011",
    "4"
],
[
    "09.01.2011",
    "3"
],
[
    "10.01.2011",
    "4"
],
[
    "11.01.2011",
    "5"
],
[
    "12.01.2011",
    "5"
],
[
    "13.01.2011",
    "5"
],
[
    "14.01.2011",
    "5"
],
[
    "15.01.2011",
    "5"
],
[
    "16.01.2011",
    "5"
],
[
    "17.01.2011",
    "5"
],
[
    "18.01.2011",
    "6"
],
```

```
            [
                "19.01.2011",
                "7"
            ],
            [
                "20.01.2011",
                "7"
            ],
            [
                "21.01.2011",
                "7"
            ],
            [
                "22.01.2011",
                "7"
            ],
            [
                "23.01.2011",
                "7"
            ],
            [
                "24.01.2011",
                "8"
            ],
            [
                "25.01.2011",
                "8"
            ],
            [
                "26.01.2011",
                "8"
            ],
            [
                "27.01.2011",
                "9"
            ],
            [
                "28.01.2011",
                "8"
            ]
        ]
    },
    "analysisType": "scorecard",
    "dailyUnique": 0,
    "totalRowCount": 0,
    "analysisTabHead": [
        {
            "dataType": "string",
            "title": "Key figure",
            "contentType": "normal"
        },
        {
            "dataType": "number",
            "title": "Today and last 30 days",
            "contentType": "normal"
        },
```

```
        {
            "dataType": "number",
            "title": "Today and last 30 days (%)",
            "contentType": "normal"
        },
        {
            "dataType": "number",
            "title": "Last 7 days",
            "contentType": "normal"
        },
        {
            "dataType": "number",
            "title": "Last 7 days (%)",
            "contentType": "normal"
        },
        {
            "dataType": "number",
            "title": "Last 14 days",
            "contentType": "normal"
        },
        {
            "dataType": "number",
            "title": "Last 14 days (%)",
            "contentType": "normal"
        },
        {
            "dataType": "number",
            "title": "Last 90 days",
            "contentType": "normal"
        },
        {
            "dataType": "number",
            "title": "Score",
            "contentType": "normal"
        }
    ],
    "timeStop": "2011-01-28 23:59:59",
    "calculationTime": "2011-02-24 14:21:07",
    "chartType": "lines",
    "analysisTabFoot": [

    ],
    "analysisData": [
        [
            "Page Impressions",
            "7562",
            "-2.2",
            "7733",
            "-0.9",
            "7805",
            "33.8",
            "5835",
            "10"
        ],
```

```
            [
                "Visitors",
                "1503",
                "-3.3",
                "1554",
                "-3.7",
                "1614",
                "29.4",
                "1247",
                "7"
            ]
        ],
        "timeStart": "2010-12-29 00:00:00",
        "analysisTitle": "Scorecard"
    }
}
```


Example for analysisType „status“:

There is an additional array “kpiData” which has data for the visualization of the status kpi (=speedometer). This is a simple table of two or three rows.


```
{
    "version": "1.1",
    "result": {
        "analysisWarnings": [

        ],
        "kpiData": [
            [
                "Durchschnitt",
                "7733"
            ],
            [
                "Gesamt",
                "54134"
            ]
        ],
        "analysisType": "status",
        "dailyUnique": 0,
        "totalRowCount": 7,
        "analysisTabHead": [
            {
                "dataType": "string",
                "title": "Tage",
                "contentType": "main"
            },
            {
                "dataType": "number",
                "title": "Page Impressions",
                "contentType": "normal"
            }
        ],
```

```
            "timeStop": "2011-01-27 23:59:59",
            "calculationTime": "2011-02-24 14:30:11",
            "chartType": "lines",
            "analysisTabFoot": [
                [
                    "Gesamt",
                    "54134"
                ]
            ],
            "analysisData": [
                [
                    "21.01.2011",
                    "8228"
                ],
                [
                    "22.01.2011",
                    "4918"
                ],
                [
                    "23.01.2011",
                    "7975"
                ],
                [
                    "24.01.2011",
                    "9261"
                ],
                [
                    "25.01.2011",
                    "8294"
                ],
                [
                    "26.01.2011",
                    "7403"
                ],
                [
                    "27.01.2011",
                    "8055"
                ]
            ],
            "timeStart": "2011-01-21 00:00:00",
            "analysisTitle": "KPI"
        }
    }
```

Example for analysisType „pivot“:

This is the same as the analysis for analysisType “analysis”, except there can be more than one elements in the analysisTabHead parameter with contentType “main”


```
{
    "version": "1.1",
    "result": {
        "analysisWarnings": [

        ],
        "analysisType": "pivot",
        "dailyUnique": 0,
        "totalRowCount": 10078,
        "analysisTabHead": [
            {
                "dataType": "string",
                "title": "Browser",
                "contentType": "main"
            },
            {
                "dataType": "string",
                "title": "Pages",
                "contentType": "main"
            },
            {
                "dataType": "number",
                "title": "Page Impressions",
                "contentType": "normal"
            },
            {
                "dataType": "number",
                "title": "Visitors",
                "contentType": "normal"
            }
        ],
        "timeStop": "2011-01-28 23:59:59",
        "calculationTime": "2011-02-24 14:35:44",
        "chartType": "bars",
        "analysisTabFoot": [

        ],
        "analysisData": [
            [
                "Firefox 3.6",
                "manieren_per_mausklick",
                "6551",
                "4616"
            ],
            [
                "Firefox 3.6",
                "dresscode_einmaleins",
                "2188",
                "1648"
            ],
```

```
                [
                    "Firefox 3.6",
                    "-",
                    "83160",
                    "15008"
                ],
                [
                    "Internet Explorer 8.x",
                    "manieren_per_mausklick",
                    "4530",
                    "2872"
                ],
                [
                    "Internet Explorer 8.x",
                    "-",
                    "0",
                    "1503"
                ],
                [
                    "Internet Explorer 8.x",
                    "-",
                    "59680",
                    "11502"
                ],
                [
                    "-",
                    "-",
                    "226857",
                    "42673"
                ]
            ],
            "timeStart": "2010-12-29 00:00:00",
            "analysisTitle": "Pivot"
        }
    }
```

11 Request of Reports

By using the method „getReportData“ you can retrieve data of reports.

Request

Parameter	Optional	Description
token	no	Token (Parameters login, pass, customerId are also possible)
language	yes	Language
time_start	yes	Start time, format: YYYY-MM-DD or YYYY-MM-DD HH:mm:ss
time_stop	yes	Stop time, format: YYYY-MM-DD or YYYY-MM-DD HH:mm:ss
report_name	no	Id of wanted report
footerIdentifier	yes	The string that identifies a footer row in a pivot analysis. Default is ‘-’.

Example:

```
{
    "params": {
        "token": "98991ec3459031c4bd9ddfeb4e142045_52b6b54cbb4e9654d1ff032521a86270",
        "report_name": "Overview Report"
    },
    "version": "1.1",
    "method": "getReportData"
}
```

Response

Parameter	Description
author	Name of the author of the report
title	Title of the report
description	Description of the report
analyses	Array of analysis data (see Analysis Requests)

Example:


```
{
    "version": "1.1",
    "result": {
        "author": "John Doe",
        "title": "Overview Report",
        "description": "",
        "analyses": […]
    }
}
```

12 Data Export

By using the method „exportData“ you can export category / parameter data. The Limit for exportable rows is 10.000 by default. If more rows are to be exported then it is possible to split the data using the startRow and endRow parameters to export, for example, rows 1 to 10.000, then rows 10.001 to 20.000 and so on.

Request

Parameter	Optional	Description
token	no	Token (Parameters login, pass, customerId are also possible)
startRow	no	Initial line
endRow	no	End line
type	no	Possible values: content_categories, content_categories_time, customer_categories (URM Categories), page_urls, media_categories, media_playtime, basket_categories, basket_categories_time, product_urls, campaigns, campaigns_time, ecommerce_parameters, time_categories
startDate	yes	Format YYYY-MM-DD Mandatory for parameter export and timedependent categories.
endDate	yes	Format YYYY-MM-DD Mandatory for parameter export and timedependent categories.
title	yes	Title of own parameter Mandatory for parameter export
rowsWithoutData	yes	Possible values: 0,1 Export only rows without data Standard: 0

filter	yes	Filters the results. Only supported for Content, Product and Media Categories. The filtered column is Pages, Products or Media.
activeCampaigns	yes	Export only active campaigns. This entry is only analyzed for campaigns. Yes: 1, No: 0 (Default)
dataSourceTypeFilter	yes	Only for campaign categories. Filter of the Data Source Type. Allowed values: “all”. “interfaces”, “regular”, “secondary”. Default is “regular”.
recordIntervalBegin	yes	Format YYYY-MM-DD. “recordIntervalEnd” must be set.
recordIntervalEnd	yes	Format YYYY-MM-DD. “recordIntervalBegin” must be set.
campaignsStartTimeIntervalBegin	yes	Only for campaign categories. Format YYYY-MM-DD. “campaignsStartTimeIntervalEnd” must be set.
campaignsStartTimeIntervalEnd	yes	Only for campaign categories. Format YYYY-MM-DD. “campaignsStartTimeIntervalBegin” must be set.
campaignsStopTimeIntervalBegin	yes	Only for campaign categories. Format YYYY-MM-DD. “campaignsStopTimeIntervalEnd” must be set.
campaignsStopTimeIntervalEnd	yes	Only for campaign categories. Format YYYY-MM-DD. “campaignsStopTimeIntervalBegin” must be set.

Example:

```
{
    "params": {
        "token": "98991ec3459031c4bd9ddfeb4e142045_52b6b54cbb4e9654d1ff032521a86270",
        "startRow": 1,
        "endRow": 1000,
        "type": "content_categories"
    },
    "version": "1.1",
    "method": "exportData"
}
```

Response

Parameter	Type	Description
	array	

Example:

```
{
    "version": "1.1",
    "result": [
        ["Pages", "Category (Text) – Main category", "Category (Figure) – ownC. "],
        ["index", "archive", "1"],
        ["home", "archive", "2"]
    ]
}
```

13 Data Import

By using the method „importData“ you can import category / parameter data. The Limit for importable rows is 10.000 by default.

Request

Parameter	Optional	Description
token	no	Token (Parameters login, pass, customerId are also possible)
uploadData	no	Two-dimensional Array – Format should be the same as given by the export method (see “Data Export”). The first row of the array contains the headlines of data, all other rows the data itself. The decision which data will be imported is based on the headlines.
uploadType	no	content_categories (including Page URLs), content_categories_time, customer_categories (URM Categories – upload via customer_id), customer_categories_deid (URM Categories – upload via End-Device-Visior-ID), media_categories (including Media Playtime) basket_categories (including Product URLs) basket_categories_time, product_categories, campaigns, campaigns_time, ecommerce_parameters, time_categories, session_parameter_tv

Example:

```
{
    "params": {
        "token": "98991ec3459031c4bd9ddfeb4e142045_52b6b54cbb4e9654d1ff032521a86270",
        "uploadType": "content_categories",
        "uploadData": [
            ["Pages", "Category (Text) – Main category", "Category (Figure) – ownC. "],
            ["index", "archive", "1"],
            ["home", "archive", "2"]
        ]
    },
    "version": "1.1",
    "method": "importData"
}
```

Example for changing a campaign:

```
{
    "params": {
        "token": "98991ec3459031c4bd9ddfeb4e142045_52b6b54cbb4e9654d1ff032521a86270",
        "uploadType": "campaigns",
        "uploadData": [
            [
                "Action (delete, suggest)",
                "Advertising Media Title*",
                "DataSourceType (page, mediacode, action, seo, referrer)",
                "DataSourceValue*"
            ],
            [
                "",
                <newCampaignName1>,
                "mediacode",
                <Mediacode1>
            ],
            [
                "",
                <newCampaignName2>,
                "mediacode",
                <Mediacode2>
            ],
            ...
            [
                "",
                <newCampaignNameN>,
                "mediacode",
                <MediacodeN>
            ]
        ]
    },
    "version": "1.1",
    "method": "importData"
}
```

It will rename the campaign <Mediacode1> to <newCampaignName1>, <Mediacode2> to <newCampaignName2> and so on.

Example for deleting a campaign:

```
{
    "params": {
        "token": "98991ec3459031c4bd9ddfeb4e142045_52b6b54cbb4e9654d1ff032521a86270",
        "uploadType": "campaigns",
        "uploadData": [
            [
                "Action (delete, suggest)",
                "Advertising Media Title*",
                "DataSourceType (page, mediacode, action, seo, referrer)",
                "DataSourceValue*"
            ],
            [
                "delete",
                <CampaignName1>,
                "mediacode",
                <Mediacode1>
            ],
            [
                “delete",
                <CampaignName2>,
                "mediacode",
                <Mediacode2>
            ],
            ...
            [
                "delete",
                <CampaignNameN>,
                "mediacode",
                <MediacodeN>
            ]
        ]
    },
    "version": "1.1",
    "method": "importData"
}
```

It will delete the campaigns with <Mediacode1>, <Mediacode2>, and so on.
It is also possible to merge different actions in one upload for changing and deleting campaigns.

Response

Parameter	Type	Description
	string	Submits empty array

Example:

```
{
    "version": "1.1",
    "result": []
}
```

Example for URM Category Import via URM - Custom Visitor Id:

```
{
  "params": {
    "token": "c7a0768a865b9a20a6ba2278e6ebf686_6f9d875ca7abdf76969f07a2fefcdaba",
    "uploadType": "customer_categories",
    "uploadData": [
      [
        "Customer",
        "URM – Category Name",
        "URM - Gender"
      ],
      [
        "508371",
        "Caras 1116",
        "Woman"
      ]
},
  "version": "1.1",
  "method": "importData"
}
```

Example for URM Category Import via End Device Visitor ID:

URM Categories are related to one URM – Custom Visitor ID. If you're Account is configured to run cross-device, several End Device Visitor Ids can belong to one URM – Custom Visitor ID.
Please be aware, that if you upload different values for a category to different devices of one Customer, only one value will be taken.
We recommend to use the Upload via URM – Custom Visitor ID instead.


```
{
  "params": {
    "token": "c7a0768a865b9a20a6ba2278e6ebf686_b431121901c3e97ed8da17fc331da9c4",
    "uploadType": "customer_categories_deid",
    "uploadData": [
      [
        "deid",
        "URM – Category Name",
        "URM - Gender"
      ],
      [
        "2148028690700883469",
        "Caras 1116",
        "Woman"
      ]
    ]
  },
  "version": "1.1",
  "method": "importData"
}
```

14 TV data deletion

By using the method „deleteTvData“ you initiate the deletion of TV data.

Request

Parameter	Optional	Description
customerId	no	ID of wanted customer (Track ID)
login	no	Login Webtrekk Frontend
pass	no	Password Webtrekk Frontend
time_start	no	Format YYYY-MM-DD HH:mm:ss
time_stop	no	Format YYYY-MM-DD HH:mm:ss

Example:

```
{
    "params": {
        "login": "login",
        "pass": "pass",
        "customerId": "11111111111",
        "time_start": "2013-12-01 00:00:00",
        "time_stop": "2013-12-31 23:59:59"
    },
    "version": "1.1",
    "method": "deleteTvData"
}
```

Response

Parameter	Type	Description
	array	Submits array [“Deleting Tv Data initiated“]

Example:


```
{
  "version": "1.1",
  "result": ["Deleting Tv Data initiated"]
}
```

15 Request Limitations

To prevent blocking the JSON API and the server by a high frequency of API requests of one customer the number of requests per time period is limited. If one single Q3 user sends requests to the JSON API with a frequency higher than the defined limits, the JSON API will refuse subsequent requests for the defined period.

The limitations are defined by the following values.

Name	Value	Description
JSON_REQUESTS_ALLOWED_PER_PERIOD	120	Default long-term number of requests
JSON_REQUESTS_PERIOD	600	Default long-term period time
JSON_REQUESTS_ALLOWED_PER_PERIOD_SEC	5	Default short-term number of requests per seconds time
JSON_REQUESTS_PERIOD_SEC	10	Default short-term period seconds time

Short-term limit

The short-term limit applies if the customer sends a lot of requests in a relatively short period of time, e.g. 1 request per second. In the default configuration, the limitation applies after 10 seconds when there are more than 1 requests per 2 seconds. After the limit is reached the following request will be accepted 1 per 10 seconds until the frequency goes back to the allowed level.

Long-term limit

The long-term limit applies if the customer sends a lot of requests in a relatively long period of time, e. g. 1 request every 4 seconds. In the default configuration, the limitation applies after 10 minutes (600 seconds) when there are more than 120 requests within this time period. After the limit is reached the following request will be accepted 1 per 10 minutes until the frequency goes back to the allowed level. The long-term limit will be also applied to the requests which are limited by the short-term limit and still coming with the same frequency within the long-term period.

In case a limit was reached, the API will respond with the following message and header:

```
{
    "version":"1.1",
    "error": {
        "name": "JSONRPCError",
        "message": "To many requests ([JSON_REQUESTS_ALLOWED_PER_PERIOD |
            JSON_REQUESTS_ALLOWED_PER_PERIOD_SEC]) within
            [JSON_REQUESTS_PERIOD | JSON_REQUESTS_PERIOD_SEC] seconds."
    },
    "code" : 501
}

HTTP-Header:
Status: 200 OK
Content-Type: application/x-www-form-urlencoded
```

Example:

```
{
    "version":"1.1",
    "error": {
        "name": "JSONRPCError",
        "message": "To many requests (120) within 600 seconds."
    },
    "code" : 501
}

HTTP-Header:
Status: 200 OK
Content-Type: application/x-www-form-urlencoded
```

Please note that the described limitations can not be changed individually per account. We reserve the right to change the limitation values in case there is still a risk of blocking the API with the current values.

16 Contact

If you have any questions please feel free to contact us:

Webtrekk GmbH
Robert-Koch-Platz 4
10115 Berlin, Germany

fon +49(0)30 - 755 415 - 0
fax +49(0)30 - 755 415 - 100
info@webtrekk.com

http://www.webtrekk.com