

Integration and Configuration of the Realtime Data Feed Service



Contents

1	General	3
2	Webtrekk Q3 – creating a feed.....	4
2.1	Analyses as data feeds	4
2.2	Manually create a data feed	5
3	Integration and use	6
3.1	Using the plugin	7
4	Contact.....	10

1 General

With the Realtime Data Feed Service you can retrieve data from Webtrekk Q3 in real time. Based on tracking data, you can now customise your website depending on specific users. Based on the data, ads can be dynamically selected or search results and recommendations adapted to the each user's surfing behaviour.

For example, you can integrate applications into your ad offering to show, in real time,

- the day's top 10 products,
- the most-read articles,
- the most-bought articles for men

as dynamic lists.

Use the service to retrieve content from your website that perfectly matches your user groups and makes their experience of using it that much better.

This document describes the configuration and technical integration needed to use the Realtime Data Feed Service.

Prerequisites for the successful integration are:

- an existing Webtrekk account
- an existing Webtrekk integration with pixel on the website
- the released Realtime Data Feed Service
- data feeds saved in the tool

The integration process includes the following points:

- possible release of features (by getting in touch with your contact)
- feed set-up (analyses) in Webtrekk
- integration of feed queries in your website.

2 Webtrekk Q3 – creating a feed

Realtime data queries are only possible if the relevant feed has been set up in Webtrekk. Once the service has been released, there are two ways of creating a feed in the Webtrekk interface.

- save an analysis as a data feed or
- manually create a data feed.

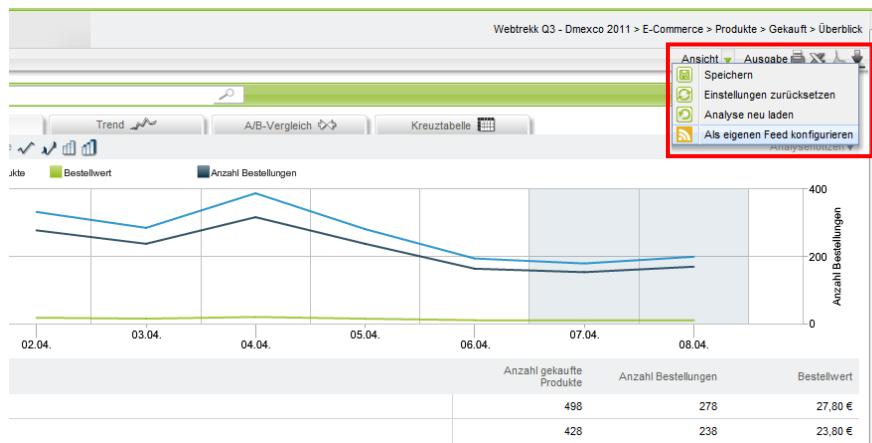
See under “Configuration → Feeds” for an overview of data feeds already created. You can always view and edit their statuses and configurations there.

Please contact your key account contact or technical support at support@webtrekk.com to have the Realtime Data Feed Service released.

2.1 Analyses as data feeds

Any analysis can be saved as a data feed.

To do this, call up an analysis in the tool, e.g. “E-commerce → Products → Bought → Bought products”. Under “View” click the green arrow and select “Configure as custom feed”.



A screen now opens for entering a name and description. You can also specify a time period (yesterday or last seven days) and a line limit. The feed is then saved.

See the online help for further configuration information (question mark in the top left). Training documentation is available on data feeds, which you are welcome to order from us.

2.2 Manually creating a data feed.

Alternatively, data feeds can be created manually.

This is done in the Webtrekk tool interface under “Configuration → Feeds → New feed”.

Because data feeds can be used in different ways, those for the realtime service must be labelled accordingly:

- **Type of feed:** Export
- **Export destination:** Realtime Data Feed Service

You can customise your feed under “Type of data”, “Exported data”, “Time period” and “Line limit”. The feed is then saved.

When configuring a feed please note the following.

- If you choose one of the periods "Yesterday", "Last 7 days", "Last week", "Last 14 days", or "Last 28 days" it is not sensible to set "Actualization" (in the case of "Daily") to "00:00" o clock because in that case the feed will in general not contain all data of the chosen period. "Actualization" determines when the feed is populated and at 0:00 may not include the most recent data. It is recommended to set the "Actualization" to "06:00" or later to ensure that all data of the chosen period is included in the feed.
- If you choose one of the periods "Last 10 minutes", "Last 20 minutes", "Last 30 minutes", "Last 60 minutes", "Last 60 minutes, to the minute", "Last full hour", "Last full 6 hours", or "Current hour" then "Actualization" should be carefully set. For example, it is not sensible to set "Actualization" to "Daily" if the period is set to "Last 10 minutes".

? Configure Feed	
Title	<input type="text"/>
Description	<input type="text"/>
Active	<input checked="" type="radio"/> Yes <input type="radio"/> No
Actualization	Daily <input type="text" value="00:00"/> o'clock
Feed Type	Export
Data Type	Segment
Exported Data	URM - Visitor Id
Period	Last 10 minutes
Row Limit	10000 <small>Maximum: 10000</small>
Export Target	FTP
Host	<input type="text"/>
Remote Path	<input type="text"/>
Login	<input type="text"/>
Password	<input type="text"/>
Port	21
	<input type="button" value="Test Connection"/>
File Name	Filename_<YYYY-MM-dd-HHmm> .CSV
File Format	CSV
Field Separator	:
Field Delimiter	'
Header Exists	<input checked="" type="radio"/> Yes <input type="radio"/> No
Time Format	YYYY-MM-dd
Preview	A preview is not possible, because the configuration is not complete

See the online help for configuration information on each of the fields (question mark in the top left).

3 Integration and use

To query the tracking data with the Realtime Data Feed Service, you must make a HTTP-GET request to the Webtrekk server.

The URL is made up as follows:

[Feed Domain + /api/] + [TrackId] + [/feeds/] + [Feed ID] + [Segment 1] + [Segment 2] + [...] + [Segment n]

[http://feeds.webtrekk.com/api/] + [111111111111111] + [/feeds/] +
[06e8915562dc615e6420e1bce4c5f6b8] + [/page1/]

The Realtime Data Feed Service returns a JSON object in reply.

Example of a feed URL:

http://feeds.webtrekk.com/api/385255285199574/feeds/24862ae69416de10d0de9cce1f5a50f9/page1

Return value:

["http://webtrekk.de/example1", "value1"]

3.1 Using the plugin

To simplify use, Webtrekk offers a ready plugin to retrieve data directly from the Realtime Data Feed Service.

The “WebtrekkRDFSClient” function initialises the plugin and returns the “getFeedData” function that you need to send queries to the Realtime Data Feed Service. To use the “WebtrekkRDFSClient” function, integrate the “webtrekk_rdfs.js” file into your page.

All queries are started automatically and asynchronously in the background and, as a result, do not slow up the page display.

```
/**  
 * @param {String} trackId Webtrekk Track-ID  
 * @param {Integer} max. timeout in milliseconds [optional]  
 * @param {String} feedURL Base URL for querying the Realtime Data Feed Service [optional]  
 * @ a function to query the data is returned  
 */  
WebtrekkRDFSClient(trackId, timeout, feedURL)
```

```
/**
 * @param {Function} callback function subsequently executed
 * @param {String} feedId Feed ID
 * @param {Array} segments segments to filter by [optional]
 */
getFeedData(callback, feedId, segments)
```

Standard query – segment unspecified

Example of a complete query without segments:

```
<script type="text/javascript">
<!--
  var wt_rdsfConfig = {
    trackId : "385255285199574",
    feedId : "83142aac0858927fd70757282dab0003",
    timeout : 3000
  };

  function alertFeed(json) {
    if(json.length > 0) {
      alert(json.join(", "));
    }
  };

  var wt_rdfsTest = WebtrekkRDFSClient(wt_rdsfConfig.userId, wt_rdsfConfig.timeout);
  wt_rdfsTest.getFeedData(alertFeed, wt_rdsfConfig.feedId);
  //-->
</script>
```

In the above example, the RDFS client with User ID “385255285199574” and a timeout of 3000 milliseconds is initialised. The data is then polled for the feed with the ID “83142aac0858927fd70757282dab0003”. Once the data is reloaded the callback function “alertFeed” is called and the data output as an array.

If no reload is possible within the timeout, the callback function returns an empty array as the result.

Nested queries – segment specified

Example of a complete query with segments:

```
<script type="text/javascript">
<!--
  var wt_rdsfConfig = {
    trackId : "385255285199574",
    contentFeedId : "83142aac0858927fd70757282dab0003",
    linkFeedId : "24862ae69416de10d0de9cce1f5a50f9",
    timeout : 3000
  };

  function alertContent(json) {
    if(json.length > 0) {
      alert(json.join("; "));
      for(var i = 0; i < json.length; i++) {
        wt_rdfsTest.getFeedData(alertLink, wt_rdsfConfig.linkFeedId, [json[i]]);
      }
    }
  };

  function alertLink(json) {
    if(json.length > 0) {
      alert(json.join("; "));
    }
  };

  var wt_rdfsTest = WebtrekkRDFSClient(wt_rdsfConfig.trackId, wt_rdsfConfig.timeout);
  wt_rdfsTest.getFeedData(alertContent, wt_rdsfConfig.contentFeedId);
  //-->
</script>
```

In the above example, the RDFS client with User ID “385255285199574” and a timeout of 3000 milliseconds is initialised. The data is then polled for the feed with the ID “83142aac0858927fd70757282dab0003”. Once the data is reloaded the callback function “alertContent” is called and the data output as an array.

If the passed array contains elements, each Content ID triggers a query to another feed with ID “24862ae69416de10d0de9cce1f5a50f9”. Here, the Content ID acts as a segment passed as an array. Once the data is reloaded the callback function “alertLink” is called and the data output as an array.

If no reload is possible within the timeout, the callback function returns an empty array as the result.

4 Contact

Please don't hesitate to contact us if you have any questions about using the product. Webtrekk offers various support packages for priority support or comprehensive advice. Please feel free to contact us to obtain your own personalised offer.

Webtrekk GmbH
Hannoversche Strasse 19
D-10115 Berlin

Phone +49 (0)30 755 415 0
Fax +49 (0)30 755 415 100

support@webtrekk.com

<http://www.webtrekk.com>