# Site JavaScript API

version 1.7

selligent

Product Manual

Selligent

# 1  Foreword

Copyright

User is aware that this manual may contain errors or inaccuracies and that it may be revised without advance notice. This manual is updated frequently.

Selligent welcomes any recommendations or suggestions regarding the manual, as it helps to continuously improve the quality of our products and manuals.

# 2  Table of Contents

# 3  Standard include script

The following script should be included on the website and will be provided with the Site web tool, all variables will be pre-filled and are here replaced with placeholders.

```
var wa   = document.createElement('script'),
   wa_s = document.getElementsByTagName('script')[0];
wa.src  = '//path_to_script_will_be_provided_in_tool/xxxxxx.js ';
wa.type = 'text/javascript';
wa_s.parentNode.insertBefore(wa, wa_s);
wa.bt_queue = [];
wa.afterInit = function()
{
 wa.bt_queue.push(JSON_OBJECT);
};
```

The **property wa.src** refers to the tracking js file in the Site module. The installation path may differ from one installation to another.


This is the standard way the script can be included. The following section will provide an example of how a tracking call can be performed.

# 4  API Overview

## 4.1  Tracking

In order to perform a tracking call we need to push a json-object (tracking definition) onto a queue. This is done with the instruction wa.bt_queue.push(JSON-OBJECT).

Site constantly evaluates the queue and performs a tracking call based on the provided configuration per item, logging the user activity and returning the requested data. All parameters belonging to this call have to be passed as a JSON object.

The following table will provide an overview of all the different properties which can be passed in the configuration-object. A tracking-call example is shown below:

| Property | Default value | Description |
|---|---|---|
| customIdentifier | empty | Sets a possible custom identifier by which the user is uniquely defined. Typically, this will be a value by which you will be able to recognize users between different sessions (for instance login id or guid). **It is of great importance that the value provided here is valid and unique per user. If no customIdentifier is known at the time of the call, this property should not be included in the tracking call** |
| tagValues | empty | A JSON array of tags with their corresponding tag-values. These are the tags that will actually be measured:<br><br>`[`<br>`    {`<br>`        "tag":"CATEGORY",`<br>`        "value":"CYCLING"`<br>`    }, {`<br>`        "tag":"ZIPCODE",`<br>`        "value":"3800"`<br>`    }`<br>`]`<br><br>Note: If TRAFFICSOURCE is included in the tracking call, the corresponding system tag is overruled. Possible values for TARFFICSOURCE are :<br>• Search,<br>• PaidSearch<br>• Social<br>• Affiliate<br>• Direct<br>• Email |
| finishedCallback | empty | It is possible to supply a function which is executed once the tracking call is successfully finished.<br><br>For instance this can be useful when we would like to make sure that the segment evaluation has occurred on the last tracking data and hence the tracking call is finished, a callback can be supplied.<br><br>Example Function:<br>`window.bt_trackingFinishedCallback = function (data) {`<br>`    var profileInfo = data;`<br>`}` |
| errorCallback | empty | Similar to the finishedCallback, it is possible to supply a function which is executed once the tracking call has finished with an error.<br><br>The parameter for this function is the trackingCall for which the error has occurred. |

| Property | Default value | Description |
|---|---|---|
| | | Example function:<br>```window.bt_trackingErrorCallback = function (data) {```<br>```    var trackingCall = data;```<br>```}``` |
| async | true | Whether the tracking call should be performed asynchronous or not. |
| isEvent | false | Whether the tracking call represents an event on the website instead of a page visit |
| isTrack | true | Whether to perform tracking or not. Setting this to false will allow you for example to perform targeting without actually tracking a new hit. Typically used when targeting should only be done after tracking and some extra custom javascript has been executed. |
| isTargeting | false | Whether the tracking call is used for targeting. If so, the call will only be executed when the DOM is ready. The call will make sure the found placeholders are properly filled with offer/action-content.<br><br>Important remark: Site only adds content to the website, it doesn't remove it. So, if isTargeting is set to true and a profile is currently in an audience for which specific content must displayed on the website, this content will be shown. If at a certain moment in time the profile is no longer in this audience, the content will still be displayed on the website (eg. An offer exists where some content is shown when the profile has at least one item in his cart. When a profile adds an item in the cart and the property isTargeting is set to true, the offer is evaluated and the content is shown. However, if the profile removes the item from the cart and the property isTargeting is set to true, the offer is re-evaluated but the content will NOT be removed although the profile is no longer in the audience. |
| exposedFields | empty | Used to retrieve exposed fields that are configured in the universe. See section 4.4 Using exposedFields for more information. |
| profileData | empty | [DEPRECATED] Use exposedFields<br><br>The property is still available for compatibility reasons, but should no longer be used, it may be removed in the future. |
| tagValueData | empty | [DEPRECATED] Use exposedFields<br><br>The property is still available for compatibility reasons, but should no longer be used, it may be removed in the future. |
| CRMData | empty | [DEPRECATED] Use exposedFields<br><br>The property is still available for compatibility reasons, but should no longer be used, it may be removed in the future. |
| events | empty | A JSON array of events with their type, itemName and itemId. Currently, only view and read are available as event types.<br>The itemName refers to the item entity that can be configured within the universe.<br>The itemId should be the customer's unique identifier for the event resource (e.g. the article being read).<br><br>```[```<br>```    {```<br>```        "type":"view",```<br>```        "itemName":"ARTICLE" ,``` |

| Property | Default value | Description |
|---|---|---|
|  |  | ```
      “itemId”:”1000”
    }, {
      “type”:”read”,
      “itemName”:”ARTICLE” ,
      “itemId”:”1002”
    }
  ]
``` |
| items | empty | A JSON array of items with the item's name, id, referenceDate and fields.<br>The name of the item should correspond to the public name that is defined within the universe.<br>The id should be the customer's unique identifier of the item resource.<br>The referenceDate specifies when the content has last been updated.<br>The fields property is a JSON object that contains the fields containing the content for the current item. The field names should correspond to the item definition in the universe.<br><br>```
[
    {
      “name”:”ARTICLE”,
      “id”:”1000”,
      “referenceDate”:”2017-06-06T12:00:00” ,
      “fields”: {
        “title”: “Title of article 1000”,
        “description”: “Description of article 1000”,
        “imgUrl”: “http://articles.com/1000”
      }
    }, {
      “name”:”ARTICLE”,
      “id”:”1002”,
      “referenceDate”:”2017-06-06T14:00:00” ,
      “fields”: {
        “title”: “Title of article 1002”,
        “description”: “Description of article 1002”,
        “imgUrl”: “http://articles.com/1002”
      }
    }
]
``` |

Following is an example of a tracking call:

```
wa.bt_queue.push({
  "customIdentifier": "12345",
  "tagValues":[
    { "tag": "CATEGORY", "value": "CYCLING" },
    { "tag": "ZIPCODE", "value": "3800" },
    { "tag": "INTEREST", "value": "SPORT" }
  ],
  "finishedCallback": "bt_trackingFinishedCallback",
  "errorCallback": "bt_errorCallback",
  "async": false,
  "isEvent": false,
  "isTargeting": false,
  "exposedFields" : [
    { "field": "Reidentified"},
    { "field": "Identified"},
    { "field": "CustomId"},
    { "field": "FirstHitDateTime"},
    { "field": "AvgVisitDuration"},
    { "field": "HitsVisit"},
    { "field": "DAYOFWEEK", "type": "Count", "parameter": ""},
    { "field": "BOUGHTITEMS", "type": "Count", "parameter": ""},
    { "field": "MAIL"},
    { "field": "NAME"},
  ]
});
```

## 4.2  Targeting

The following methods are available in the API:

| Function | Description |
|---|---|
| BT.getProfileInfo() | Returns an object with profile-information. This is the same object that is returned to the callback-function when performing the tracking-call.<br><br>Note that the names returned for offers, tags and profile-fields are the internal public names.<br><br><pre>{<br>    "profileId": "aaa-aaa-aaa-aaa-aaa",<br>    "profileInfo": {<br>        "offers": [<br>            { "name": "OfferName", "inOffer": false,<br>                "date": "2014-04-15T17:15:08.375+02:00" }<br>        ],<br>        "exposedFields": [<br>            { "field": "tagName", value:["value1", "value2"]},<br>            { "field": "aaa", value:["xxx"]},<br>            { "field": "fieldName", value:["xxx"]},<br>        ],<br>        "tags": [<br>            { "name": "tagName", "values: [ "value1", "value2" ]}<br>        ],<br>        "CRMFields": [<br>            { "name": "aaa", "value": "xxx" }<br>        ],<br>        "profileFields": [<br>            { "name": "fieldname", "values": "xxx" }<br>        ]<br>    }<br>}</pre>Note: the boolean indicates if the profile is in the offer or not. It is still possible that even if the boolean is 1 (so the profile is in the offer) that the offer is not shown. (ex. The offer domain does not correspond with the current domain; or |

| Function | Description |
|---|---|
| | the planning of the offer prevents it from being shown; or there is another offer with higher priority;…) |
| BT.saveProfileInfo() | Stores the returned profile-info in local storage or a cookie with name "sbt_pi". |
| BT.clearProfileInfo() | Clears the sbt_pi-stored value (if it exists). |
| BT.isCustomIdentified() | Returns a Boolean indicating if the web site visitor is custom-identified or not. **This function will only return a Boolean if isCustomIdentified was selected in the exposable fields in the universe configuration.** |
| BT.isThirdPartyIdentified() | Returns a Boolean indicating if the web site visitor is third-party-identified or not. **This function will only return a Boolean if isThirdPartyIdentified was selected in the exposable fields in the universe configuration.** |
| BT.isInOffer(offer) | Returns a Boolean (true or false) indicating if the web site visitor is inside the offer-target-audience or not. Note: the boolean indicates if the profile is in the offer or not. It is still possible that even if the boolean is 1 (so the profile is in the offer) that the offer is not shown. (ex. The offer domain does not correspond with the current domain; or the planning of the offer prevents it from being shown; or there is another offer with higher priority;…) offer = The public offer name for which inclusion should be checked. |
| BT.getOffers() | Returns an array of objects for all the offers the user has been or is in the Site-audience. Each returned offer-object contains: <br> - name: the public name of the offer. <br> - isInOffer: Boolean indicating if the user is still in the Site-audience. <br> - date: <br>   o if isInOffer is true, this is the date and time the user joined the offer-Site. <br> Note: the boolean indicates if the profile is in the offer or not. It is still possible that even if the boolean is 1 (so the profile is in the offer) that the offer is not shown. (ex. The offer domain does not correspond with the current domain; or the planning of the offer prevents it from being shown; or there is another offer with higher priority;…) <br><br>   o If isInOffer is false, this is the date and time the user dropped out of the offer-Site. |
| BT.getOffer(offer) | Returns, if found, the offer-object for the given offer-name. offer = The public offer name to search for. |
| BT.getProfileId() | Returns the unique identifier (hash) for the current profile. |
| BT.getExposedFields() | Returns an array of of objects for all the exposedFields that were returned in the tracking call. See section 4.4 Exposed Fields for more information. |
| BT.getExposedField(fieldname) | Returns, if found, a single exposed field object for the requested field name. See section 4.4 Exposed Fields for more information. |
| BT.getCrmFields() | [DEPRECATED] Use getExposedFields |

| Function | Description |
|---|---|
|  | The function is still available for compatibility reasons, but should no longer be used, it may be removed in the future.<br><br>- |
| BT.getCrmField(*field*) | [DEPRECATED] Use getExposedField(fieldname)<br><br>The function is still available for compatibility reasons, but should no longer be used, it may be removed in the future. |
| BT.getProfileField(*field*) | [DEPRECATED] Use getExposedField<br><br>The function is still available for compatibility reasons, but should no longer be used, it may be removed in the future. |
| BT.getTags() | [DEPRECATED] Use getExposedFields<br><br>The function is still available for compatibility reasons, but should no longer be used, it may be removed in the future.<br><br>- |
| BT.getTag(*tagName*) | [DEPRECATED] Use getExposedFields<br><br>The function is still available for compatibility reasons, but should no longer be used, it may be removed in the future. |
| BT.addTag(*tag*, *value*) | Adds a tag-value-pair onto the internal processing-queue.<br>This tag will be recorded in the next tracking-call. |
| BT.optout(period, offer) | When no arguments are specified for the optout function, a cookie with name "behavioral-disable-track" is stored to disable all behavioral tracking. To undo the opt-out, the user should manually remove this cookie or the method BT.optin() should be called.<br><br>To optout of all offers for a specified time, specify a period in minutes, in this case the cookie will not be created and Site will handle the optout.<br><br>To optout of a specific offer for a specific time also specify the public name of the offer. |
| BT.undoOptout() | [DEPRECATED] Use BT.optin()<br><br>The function is still available for compatibility reasons, but should no longer be used, it may be removed in the future. |
| BT.optin(offer) | Clears the behavioral-optout-cookie if set.<br>If an offer is specified and the user has an active optout for that offer, it will be cleared and the user will start receiving that offer again. |
| BT.trackActivity(*offer*, *activity*) | Tracks an activity defined on an Offer as "via API". As parameters, the public name for the offer and the unique name for the activity should be passed.<br><br>*offer* = The public offer name to track activity for.<br>*activity* = The activity name to track activity for. |
| BT.trackClick(*object*) | The object contains the content id and one or more callback functions.<br>The function measures a click-event for a popup or popin opened for an offer, since these can't be tracked automatically in a consistent way. This allows the user to only sense the click when (a) specific element(s) in the popup/popin is/are clicked. (e.g. an OK-button). |

| Function | Description |
|---|---|
| | E.g.<br><br>```<br>BT.trackClick ({<br>    "id": "~@CONTENTID~",<br>    "finishedCallback": function(data) {<br>        console.log(data);<br>    },<br>    "errorCallback": function(trackingCall) {<br>        console.log(trackingCall);<br>    }<br>});<br>``` |
| BT.addCartItems(*cartName*, *items*, *isTargeting*) | Adds the specified items to the cart with the specified name. As parameters, the public name for the cart, the list of items and the isTargeting flag should be passed.<br><br>*cartName* = The public name of the cart to which the items should be added<br>*items* = A json array containing the items that should be added to the cart. Each item should contain an identifier, a value and a quantity.<br>*isTargeting* = This flag indicates whether or not this call should be used for targeting.<br><br>E.g.<br><br>```<br>BT.addCartItems(<br>    "CARTNAME",<br>    [<br>        {<br>            "id": 7,<br>            "value": 649,<br>            "count": 1<br>        }<br>    ],<br>    true);<br>``` |
| BT.removeCartItems(*cartName*, *items*, *isTargeting*) | Removes the specified items from the cart with the specified name. As parameters, the public name for the cart, the list of items and the isTargeting flag should be passed.<br><br>*cartName* = The public name of the cart of which the items should be removed.<br>*items* = A json array containing the items that should be removed from the cart. Each item should contain an identifier and a quantity.<br>*isTargeting* = This flag indicates whether or not this call should be used for targeting.<br><br>E.g.<br><br>```<br>BT.addCartItems(<br>    "CARTNAME",<br>    [<br>        {<br>            "id": 7,<br>            "count": 1<br>        }<br>    ],<br>    true);<br>``` |
| BT.clearCart(*cartName*, *isTargeting*) | Clears all items from the cart with the specified name. As parameters, the public name for the cart and the isTargeting flag should be passed.<br><br>*cartName* = The public name of the cart that should be cleared.<br>*isTargeting* = This flag indicates whether or not this call should be used for targeting.<br><br>E.g.<br><br>```<br>BT.clearCart(<br>    "CARTNAME",<br>    true);<br>``` |

| Function | Description |
|---|---|
| BT.checkoutCart(cartName, isTargeting) | Performs a checkout operation on the cart with the specified name. As parameters, the public name for the cart and the isTargeting flag should be passed.<br><br>*cartName* = The public name of the cart that should be checked out.<br>*isTargeting* = This flag indicates whether or not this call should be used for targeting.<br><br>E.g.<br><br>```<br>BT.checkoutCart(<br>    "CARTNAME",<br>    true);<br>``` |
| BT.getDoNotTrack() | Return whether or not the tracking call will be executed for the current user. In case the user has the DNT header/cookie set and the universe is configured as 'do not track behavior', this method will return false. |

## 4.3  Recommendations

Currently, no API methods have been added to handle recommendations since we do not want to make it visible to all customers yet. The result of the tracking call will contain the recommendations in the following format:

```
"recommendations": [{
    "name": "RecommendationName",
    "items": [{
      "id": "1000",
      "fields": {
         "description": "Description of article 1000",
         "title": "Title of article 1000",
         "imgURL": "http://articles.com/1000"
      }
    }, {
      "id": "1002",
      "fields": {
         "description": "Description of article 1002",
         "title": "Title of article 1002",
         "imgURL": "http://articles.com/1002"
      }
    }]
  }]
```

## 4.4  Disable behavioral tracking

It is possible for a user to disable behavioral tracking, in the same way that cookies can be disabled. A message appears on the screen of the user allowing him to allow or disable behavioral tracking.

Following is an example of how this can be achieved:

1. Add an html hyperlink to execute the opt-out logic:

```
<a href="javascript:BT.optout()">Click here to opt-out of Selligent Site</a>
```

2. Make sure the Site-javascript code is loaded in the page with this hyperlink.

In the above example, when a user clicks the opt-out link the function "optout" is launched. This function sets a cookie for a long time and disables behavioral.js data collection. When a user returns later on to this

site, a check is made if the opt-out cookie has been set. If it has, the analytics.js data collection will also be disabled.

> Note: This example code assumes that you are using a single web property on your site and are only using a single domain. It only provides an opt-out function which is based on a long-term cookie. If you require opt-in functionality or if your site uses multiple web properties or domains, you will need to modify this example code, write your own opt-out code, or use other opt-out tools.

## 4.5  Using exposedFields

ExposedFields is used to retrieve exportable data for a profile from Site.

The exposedFields property replaces the CRMData, profileData and tagValueData properties.
Currently both, the old and new, are still accepted in the request as well as in the result. The data is returned in exposedFields as well as in the old properties, which should facilitate a migration from the old properties to the new exposedField format.  Existing scripts should be updated to use only exposedFields as the older properties are currently marked as deprecated and may be removed in the future.

The fields that are available are configured in the universe configuration:



The exposedField property for a request is an array of objects with the following layout:

{
   "field": "required, name of the field, configured in the universe",
   "type": "optional function type, used with tag data",
   "parameter": "optional parameter, for use with tag data"

}


The result format for exposedField is an array of objects with the following format:

{
   "field": "name of the requested field",
   "value": [ "array of values", "array of values" ]
}

Example Request:

```
wa.bt_queue.push({
  "customIdentifier": "",
  "async": false,
  "isEvent": false,
  "exposedFields": [
    { "field": "Number_of_hits_in_the_last_day" },
    { "field": "DAYOFWEEK", "type": "Count", "parameter": "" },
    { "field": "Is_quality_profile" },
    { "field": "Do_Not_Track_is_set" },
    { "field": "The_MASTER_MAIL_value" }
  ]
});
```

Example Response:

```
{
  "doNotTrack": false,
  "profileId": "…",
  "profile": null,
  "targetingInfo": null,
  "profileInfo": {
    "exposedFields": [
      {"field": "DAYOFWEEK","value": ["6"]},
      {"field": "Number_of_hits_in_the_last_day","value": ["2"]},
      {"field": "Is_quality_profile","value": ["True"]},
      {"field": "Do_Not_Track_is_set","value": ["False"]}
    ],
    "profileFields": [
      {"field": "Number_of_hits_in_the_last_day","value": "2"},
      {"field": "Is_quality_profile","value": "True"},
      {"field": "Do_Not_Track_is_set","value": "False"}
    ],
    "crmFields": [],
    "tags": [
      {"tag": "DAYOFWEEK", "values": ["6"]}
    ],
    "offers": []
  }
}
```

The type and parameter property are optional and can only be used with tags. If the tag's configuration is set to 'defined in tracking script', a function and optional parameter needs to be specified in the tracking script.



The possible options for type and parameter are :

**Type**: a choice among following options is possible:

- **Count**: counts the number of values measured for the tag. (tag value and its sub-values) at the given level (parameter)
- **Last**: the last value measured for this tag
- **Last3**: the last 3 values measured for this tag
- **TopCategory**: returns the highest scoring value for the tag at the given level
- **Top3Category**: returns the 3 highest scoring values for the tag at the given level (parameter)
- TopCategoryScore: returns the score of the highest scoring value for the tag, for the given level (see parameter).
- Top3CategoryScore: returns the scores of the 3 highest scoring values, for the given level (see parameter). Returned pipe (|) separated. E.g. 173,221|114,793|57,410
- **TopTagValue**: returns the highest scoring value at any level in the tag, starting from the given level (parameter)
- **Top3TagValue**: returns the 3 highest scoring values at any level in the tag, starting from the given level (parameter)
- **TopTagValueScore**: returns the score for the highest scoring value at any level in the tag, starting from the given level (parameter)
- **Top3TagValueScore**: returns the scores of the 3 highest scoring values at any level in the tag, starting from the given level (parameter)

**Parameter**: *optional*; indicates the hierarchical level to perform the "type"-function on. If empty, the top level is used.