

WebBehavior: Consumer Guide



Index

Index	2
What is WebBehavior?.....	3
GET Method:.....	4
POST Method:	4
Creating and updating cookies	5
What of Web Behavior must be validated on the website?.....	7
Launch.....	8

What is WebBehavior?

- WebBehavior is a component that installs and obtains information from web pages generating feeds based on information contained in said pages. These feeds are stored in Customer-defined Containers in your MasterBase Automata® account.
- WebBehavior is a web application that contains Web-API Services that interact with the browser to obtain additional information from the client that is separate from its web environment and cookies.

To do this, the driver named "Script" is established and it contains the following methods to be referenced by the client browser:

- GET: Web method that receives client data, generates a script and sends the information to the POST method of the same driver script.
- POST: Web method that receives the POST information configured in the script sent to the client with the GET web method. The system associates the information with the configured container.



Receive 2 variables per URL:

- CustomerID: Customer's identifying value (assigned to each MasterBase[®] client account)
- SiteID: Identifying value of the client's Web site (assigned to one or more web pages where the client wants to configure the WebBehavior.)
- The CustomerID and SiteID are obtained and configured internally at MasterBase[®] and must be provided to the customer.
- The script proceeds to create or update the cookies in the client with: Device and Contact.
- With the values stored in the cookies (if there are any), plus the data of the client page variables and/or URL information, the feed to be sent by the script to the container configured by the client.

POST Method:

- Receive JSON data per POST
- The feed is inserted into the configured container.
- The customer is answered with an OK.

Creating and updating cookies

A cookie is a small information sent by a website and stored in the user's browser so that the website can check the user's previous activity.

There are two types of Behavior cookies: Device and Contact:

- Device ("de" is its internal name): contains information related to the device and the connection characteristics of the contact.

The data they contain are:

- id = device ID or unique string that is generated when the cookie is created.
- ty = If M or B (Mobile or Browser)
- os = Platform
- br = UserAgent of the request.
- Expires in +3 years.
- It is assigned the masterbase.com domain or the one that appears in the configuration file.

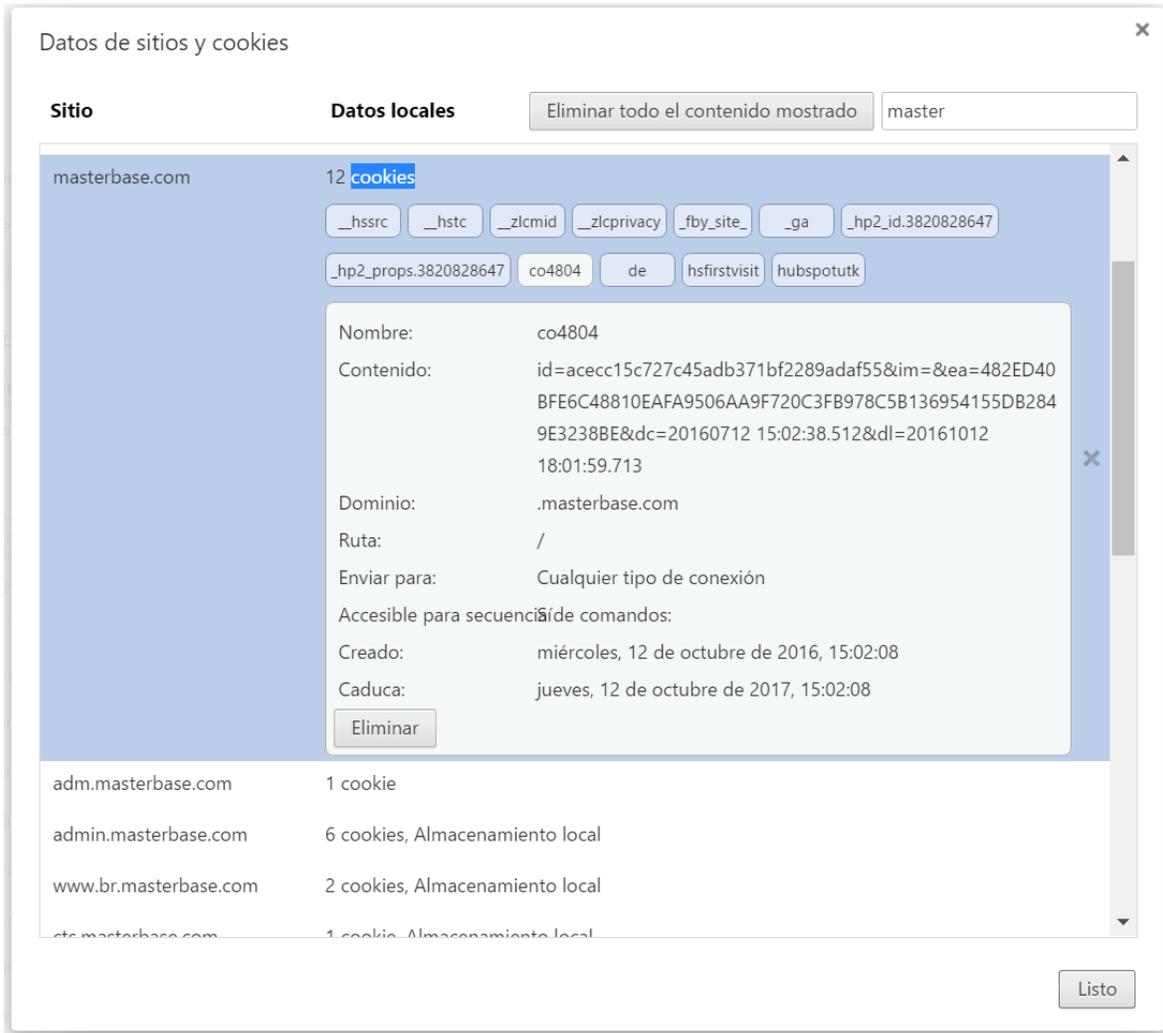
The reason for creating or updating this cookie are the following:

- It indicates if a cookie exists within the client's request.
 - If no cookie exists, the corresponding values are created and assigned.
 - If one exists, the expiration data is updated and the corresponding domain assigned
- .
- Contact (whose internal name is "co" + the identifier of the client or *unknown* if it does not exist): it contains particular information about the registered contact in each cookie.

The data they contain are:

- id = ID of the newly created or previously created device or string.
- im = e-mail identifier
- ea = email
- dc = creation date of the cookie
- dl = date of last access or reading of the cookie
- Expires in +1 year.

- It is assigned the domain masterbase.com or the one that appears in the configuration file.



View of a Contact cookie captured by WebBehavior for the site www.masterbase.com (seen from Chrome).

What of Web Behavior must be validated on the website?

1. Container: The existence of the data container (CFC) in the MasterBase® account, which will receive the information coming from WebBehavior.
2. Customer ID: The account ID (YYYY) that must be requested from Customer Service.
3. Site ID: The SiteID XX has been configured internally by MasterBase®. It must be requested from Customer Service.
4. Script: The WebBehavior code to be installed with the client (his/her web pages,) as previously mentioned, has the following standard structure and only varies depending on Client ID and Site ID:

```
<scripttype="text/javascript">
  setTimeout(function() {
    var a = document.createElement("script");
    var b = document.getElementsByTagName("script")[0];
    a.src
    =
    "https://trkweb.masterbase.com/v1/Script/YYYY/XX/automatabehavior.js";
    a.async = true;
    a.type = "text/javascript";
    b.parentNode.insertBefore(a, b
  }, 1);
</script>
```

1. Configuration of client container that must be enabled (we are delivering preconfigured CFC11).
2. Place the WebBehavior code on the selected pages, along with the variables that have been determined to identify what needs to be incorporated into the feed.
3. Customizing variables:
 - a. Variables in metadata HTML (HTML header): It is in this section where you usually find valuable information, such as title, description, keywords, etc.

Example of metadata variables in HTML on our site www.masterbase.com:

```
<!DOCTYPE html>
<html class=" js no-touch" lang="es">
  <head>
    <title>MasterBase®</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="title" content="Soluciones destinadas a la bandeja de entrada | Masterbase" />
    <meta name="Author" content="Masterbase" />
    <meta name="Subject" content=" Comunicaciones Digitales destinadas a la bandeja de entrada " />
    <meta name="keywords" content="Email Marketing, emailing, comunicaciones por email, campañas de email, envio de email, email con archivos adjuntos, facturas online, envio de cartolas, Email Transaccional, mensajes transaccionales" />
    <meta name="description" content="Expertos en soluciones de Comunicación Digital: Email Marketing, Archivos Adjuntos, Email Transaccional. Muestras herramientas online acercan a las empresas y sus clientes, a través del email." />
    <meta name="distribution" content="global" />
```

- b. JavaScript Variables: Refers to data detection mechanisms that are intentionally added to a site for WebBehavior to capture and link to the source data. They must be embedded within the page, in any of the JavaScript segments prior to the WebBehavior code entry segment.

Example of embedded JavaScript variables for capturing source data (data-language / data-product) on our site at www.masterbase.com:

```
<meta name="application-name" content="Masterbase" />
<meta name="application-name" content="MasterBehavior" data-idioma="Español" data-producto="Corporativo" />
<link href="favicon.ico" rel="shortcut icon" />
<link href="http://www.masterbase.com/" rel="canonical" itemprop="url" />
<link href="files/css/bootstrap.css" rel="stylesheet" />
<script src="https://trkweb.masterbase.com/v1/Script/4804/1/automatabehavior.js" type="text/javascript" async=""></script>
<script id="LeadinJS-1836256" src="https://js.leadin.com/js/v2/1836256.js" type="text/javascript" crossorigin="use-credentials">
```

c. URL Variables: for capturing data in the URL of the web page. Whenever these variables exist in the URL of the page, these values will be added to the feed.

Example of URL variable to capture source data (data-language / data-product)

There is the homepage.html and its URL is

<http://www.es.masterbase.com/inicio.html>

When adding data in the URL take the following into consideration:

- Just before the variables add the **?** symbol
- The structure in which the variable is added is of the type variable = value
- Add the **&** symbol between variables

In our example URL we will incorporate the variables Language:Spanish and Product:Corporate, with the result:

<http://www.es.masterbase.com/inicio.html?Idioma=Español&Producto=Corporativo>

Web Behavior will include: language:Spanish and Product:Corporate within the feed that in turn will be sent to the defined container.