Tracker

Abstract

eZ Recommender Engine collects information from the web presence of the customer to generate personalized or context based recommendations for auser profile. An event tracking process collects events like clicks, consumes, purchases. On the customer side the event tracking must be implemented by requesting a predefined URL like: **http ://event.yoochoose.net/ebl/00000/click/user00/1/123**

Depending on the event type the URL requires additional parameters. The call itself can be implemented on the client side over the so called tracking pixel or on the server side. See chapter Tracking Implementation for more information.

Click Events

A User clicks on a link that leads to a product/article page or any other page with detailed content related to a single item or a list of items. Click events also deliver information about the structure of the web presence.

The URL to track click-evens has following format: **http://event.yoochoose.net/\[solutionid\]/\[clientid\]/click/[userid]/[itemtypeid]/[itemid]** All parameters embedded in the URL are described in the Table 1.

Table 1. Embedded parameters for click-event

Parameter Name	*Description*	*Values*

solutionid	Code of the product edition	ebl, ebh, newsl, newsh
clientid	Your customer ID, a.k.a. "mandator ID" (for example "00000").	alphanumeric
userid	The ID of the user in the web presence of the customer. It could be internal customer code or session code for anonymous users. See the chapter User ID for more information.	alphanumeric
itemtypeid	Item type ID. See the chapter Item Type ID for more information.	1 – 5
itemid	Id of the item the user has clicked.	1 to 2147483647

All these parameter are required for the request. Some optional parameters can be set over query string parameters (a.k.a. GET parameters) at the end of the request.

http://event.yoochoose.net/\[solutionid\]/\[clientid\]/click/[userid]/[itemtypeid]/[itemid]?parameter1=value1&parameter2=value2

Table 2. Query string parameter for click-event

Parameter Name	*Description*	*Values*
categorypath(optional)	The Path of Categories that categorize the position of the item in the category tree, e.g. *"/Cameras&Foto/Compact Cameras/Canon".* See chapter Cathegory Path for more information.	alphanumeric

All the parameter values must be encoded using a percent-encoding (a.k.a. URL encoding) as specified in RFC 3986. For example the category path in the example above must be encoded as **%2FCameras%26Foto%2FCompact%20Cameras%2FCanon**

Click-Recommended Events

If a user clicks on a recommended link the following event should be sent to gather statistics over the acceptance of recommended items. Click-recommended events are also known as follow-events. The URL has the following format:

http://event.yoochoose.net/\[solutionid\]/\[clientid\]/clickrecommended/[userid]/[itemtypeid]/[itemid]

All the embedded parameters are the same as for a click event. The query string parameters are specified in the Table 3.

Table 3. Query string parameter for click-recommended-event

Parameter Name	*Description*	*Values*
scenario (required)	Name of the scenario, where recommendations originated from.	alphanumeric

The scenario parameter identifies the originating scenario to gain detailed statistics about which scenario motivated the user to click a recommendation. This information comes with the recommendation from the recommendation controller. See chapter Fetching Recommendation for more information.

Consume Event

The consume event is important only for publisher web pages (like a newspaper or news agency). For web shops this event makes sense only in very special cases.

The event should be sent if the user stays on the page for some predefined period of time. In this case it is assumed, that the user consumed the item (read an article or finished watching a video).

The URL for a consume event has the following format: **http://event.yoochoose.net/\[solutionid\]/\[clientid\]/consume/[userid]/[itemtypeid]/[itemid]**

The consume event has the same embedded parameters as the click event and no additional query string parameters. As mentioned above the simplest implementation is a JavaScript timer, which sends this event after some predefined time is elapsed (and the user did not change the page). The time the event is triggered after must be set dependent on the content. For example it could be 30 for a small newspaper article or a timespan calculated with the amount of words. In some special cases the event could be sent based on other criteria, like the finishing of watching a video over the integrated flash player or finishing a level of an on-line game.

Buy Events

The buyevent is only relevant, if the user is charged per item like it happens in the classical shop. If items are sold on subscription base or the web presence is ad-sponsored, this event type is not needed.

As the name suggests this event must be used if the user buys an item. It must be sent to the event tracker at the end of the check-out process of a shopping basket to ensure that no further action of the user can result in an abort.

The URL for a consume event has the following format:**http://event.yoochoose.net/\[solutionid\]/\[clientid\]/buy/\[userid\]/[itemtypeid]/[itemid]**

Additionally to the fact of buying this event can provide information about the product price. Query string parameters for the buy event are specified in Table 4. For further analysis and to avoid recommendations of articles not in the price range of the user the purchase price and the quantity should be submitted.

Table 4. Query string parameter for buy-event

Parameter name	*Description*	*Values*
quantity (optional)	The number of items the user bought. It is possible to send several events instead of setting this parameter.	integer(default is "1")

fullprice (optional)	A price for a **single** item. It contains of price in decimal format plus the currency ISO 4217 code. If the price has decimal part, the point must be used. No space must be between price and currency. Use either fullprice or the combination of price and currency	For example "12.45EUR" or"456JPY "
price (optional, deprecated)	Price in cents, if cents are defined in the currency used according to ISO 4217. For example 123 can stand for €1.23 or ¥123.	integer
currency (optional, deprecated)	The currency of the sales price formatted according the ISO 4217 (e.g. USD, CHF, etc.).	ISO 4217(default is "EUR")

For example, if a user bought 4 pens for €10 Euro, the parameter "fullprice" must be set to "2.25EUR" and parameter "quantity" to 4.

Prices in Buy Events

As mentioned above every buy event can contain a price. If the price is set, it is stored with the events and used for calculating revenue for statistics. The price must be a price the user paid for the item including all taxes and discounts. The current version of the recommendation engine doesn't sum revenues in different currencies. Every currency is presented in an own column in the statistic export. Filtering in different currencies could be difficult to manage. It is often a good idea to select a base currency and convert prices before sending the tracking event. Additionally to calculating of revenue the price set in the buy event updates the current item price. The item price is used for filtering the recommendations. If the paid price differs from user to user (usually because of discounts), it is possible that the current price will differ over the time, which usually it is not a problem. The price itself is never shown to a user and the only drawback is that sometimes too much or too few items will be filtered out based on some threshold. In order to avoid it, the price can be imported separately over the import interface. Automatic update of prices with buy events can be disabled in such case. See chapter Item Information Import for more information.

Render Event

Render events must be sent, if the web presence uses the recommendation provided by the recommendation engine and renders it on the web-page. In a combination with a predefined threshold it allows the recommender engine to exclude this item from a future result and not recommending the same item to the same user multiple times during the session.

The URL for a render event has the following format: **http://event.yoochoose.net/\[solutionid\]/\[clientid\]/rendered/\[userid\]/[itemtypeid]/[itemid[,itemid]]**

The render event has the same embedded parameters as a click event except of the item ID. It is common that recommendations are rendered as a block with multiple items. To save traffic and reduce latency it is possible to bundle multiple recommendations in one request. Several item IDs must be comma separated.

Transfer event

Recommendations rely on the fact that user actions can be correlated over a longer period of time. Moreover, recommendations like "users who viewed this item ultimately bought it" require correlating click events with subsequent purchase events. In general, users tend to browse a site anonymously and add items to their shopping cart. Up to this point, the user is identified by a visit-scoped variable (e.g. a session ID). On check-out of the shopping cart, a user may log in to an existing account. As consequence, the user identifier changes from an anonymous session ID to a persistent account ID. It is highly desirable to correlate both IDs in order to correlate the buy events (account ID) with the preceding click events (session ID). The administrative event "transfer" serves exactly this purpose.

The format of the URL is: **http://event.yoochoose.net/\[solutionid\]/\[clientid\]/transfer/[sourceuserid]/[targetuserid]**

Table 5. Embedded parameters for click-event

Parameter name	*Description*	*Values*
sourceuserid	User identifier valid up to now(usually some anonymous session ID)	alphanumeric
targetuserid	User identifier valid from now on (usually an account ID or login name)	alphanumeric

Special Events

For a more sophisticated integration into the web pages of the customer, an explicit rating of items and user interaction with the recommendation engine, the following events can be implemented. The user has different possibilities to submit user ratings or to actively influence the results of the recommendation engine.

Rate-Item

The news portal or shop allows commenting on articles, products or other content. If a user comments on an item this indicates a special interest on this topic that has to be treated separately.

The format of the URL is: **http://event.yoochoose.net/\[solutionid\]/\[clientid\]/rate/\[userid\]/[itemtypeid]/[itemid]**

This can also be used for explicit ratings like a five star rating or any other liquor scale based rating. The predefined rating can be submitted when the user comments on an item.

Table 6. Query string parameter for rate-event

Parameter name	*Description*	*Values*
rating	Value from 0 to 100	0 to 100(default "0")

User blacklists item

If the web-page offers a link or button that allows feedback like "do not recommend this item anymore" the user could express that he might bought this item on behalf of somebody else or changed his opinion after reading.

The format of the URL is: **http://event.yoochoose.net/\[solutionid\]/\[clientid\]/blacklist/\[userid\]/[itemtypeid]/[itemid]**

There are no query string parameters for this event.

Request Format and Parameter descriptions

Events can be forwarded to the recommender engine as RESTful-Requests using "http" or "https" schema using GET or POST method. Make sure that all embedded and query string parameters are URL encoded. See RFC 3986 for more information. Some of the tracking parameter will be described below. Please read the event description first for understanding a use case.

userid

All the tracking events require a user ID. This ID must define a user in the web presence non ambiguously. Every alphanumeric combination can be used as the user ID. User login or email can also be used as a user ID. Usually a user starts a session without logging in. Until the user signs up the session ID (no problem, if this ID is very long) or some other generated temporary ID can be used as a user ID. After a user signs in the temporary ID should be linked to the permanent user ID using the transfer event. See chapter_ *Transfer event* for more information. Internally the recommendation engine creates a hash of every user ID. The original ID is not saved by the recommendation engine and is not used for calculating recommendations. (It is still possible, that the original ID and IP address of the requesting host appear in the log files for debug purpose. Log files are not stored for a long time and are purged regularly.) The user ID is case sensitive.

itemtypeid

Item types were developed to be used in content provider web presences like newspapers etc.. If you are managing a classical web shop most likely you don't need different item types. Just use "1" for your products.

The recommendation engine can handle different domains of items. It is especially useful, if different types of the context are provided by the web presence, e.g. articles and images. There are some predefined item types defined and listed in the table below.

Table 7 Predefined item types

Item Type ID	*Description*
1	Product
2	Article
3	Image
4	Media
5	User Generated Content

itemid

An item ID must identify the product in the web presence non ambiguously. It must be an integer between 1 and 2147483647 (31 bit). Every item type uses its own ID scope. It is possible to have different items of different item

types but with the same item ID. Do not use thousand separator or other symbols in the item ID, only integers are allowed.

categorypath

It defines the category where the item is located in. The category path is used for item filtering. For example "recommend only items from the same category" or "exclude category 18+ from the recommendations". The categorypath parameter plays an important role as it offers the possibility to provide category based recommendations without an explicit export of the structure of a customer's website. It is used for on-the-fly updating of items. If you exportthe item database over the import interface, you should not provide the category path in the event tracking process. During recommendation requests the category path must be provided in any case!. The category path is a forward slash separated list of categories from the root like "/Cameras&Foto/Compact/Cameras/Canon". First slash (if set) is ignored. Do not forget to escape the slashes before creating a parameter value. If an item is accessible over multiple categories and multiple category paths are sent with the click-event, all the paths will be preserved. If the item is moved to another category , it will be handled as being in both categories until the old category ages out.

scenario

It is the identifier of the scenario that is called to get recommendations from (the originating scenario). It is a parameter to gather and create detailed statistical information.

Response Handling

The following HTTP response codes are used by the event tracker.

Table 8 HTTP response codes of event tracker

HTTP Status Code	*Description*
200 OK 204 No Content	Request was successfully processed.
400 Bad Request 414 Request-URI Too Long	The request is wrongly formatted. See response body for more information.
401 Unauthorized	Illegal authentication credentials (not implemented yet). Huhu
403 Forbidden	Access denied (not implemented yet)
404 Not Found	The customer ID (a.k.a. mandator ID) was not found.The event code was not found.
500 Internal Server Error	Some unspecified error. Please inform support, if it you get this error repeatedly.

The body of the response (if available) can be ignored. In case of errors it contains human readable error messages. Error messages can be changed and should not be used for automated processing.

Tracking Integration

There are several ways how tracking events can be sent. The simplest way of integration is to include an image tag within the page and use an http GET: **** An other solution would be to make the tracking request using the XMLHttpRequest object. JavaScript can access only the server the web page is loaded from. To request a recommendation server over JavaScript, one must create a proxy on the server side, which forwards the tracking request to the recommendation server. Apache's "mod_proxy" provides such functionality. A third way is sending events from the server side during the webpage preparation. In Table 9 one can find a small overview of possible caveats for different implementation types. It is possible to combine different implementation types in one project. For example one can send click events over an image tag, consume events over an XML request in JavaScript and buy events from the server side.

Table 9 Different tracking implementations

Problem	IMG Tag	XMLHttpRequest	Server Side Tracking
It is not blocked by ads blockers or no-track plug-ins.		✅	✅
It works, if JavaScript is disabled.	✅		✅
No configuration is needs on the server side.	✅		
It is compatible with frontend caching on the server.	✅	✅	
It does not delay the page rendering.	✅	✅	*
It supports authentication for tracking request (not supported yet).			✅

* To avoid delays linked to fetching recommendations, multithreaded request processing or a messaging system is needed on the server side. Dependent on the implementation language it could be quite easy (for example in Perl or Java) or a bit tricky (for example in PHP).

Event Tracking Example

To help integrating the event tracking some examples are given in this section. Out of this reason 'ebl' is used as solution ID. The client ID (a.k.a. mandator ID) used here is '00000' which is not a valid one and must be replaced with a real one.

Basic Event Use Cases

The click event is probably the most important event for enabling the recommendation engine system to provide recommendations for a customer's system. Based on the collected statistics recommendations are calculated and stored. It is important to provide as much information as possible in the request in order to store implicitly additional

information about the item's attributes with the tracking events if the explicit import functionality is not used. An example of a **click** use case is given here:

1. User "Js79009234YU7" navigates to an item with the id 123 and type 1 which is located under Shoes -> Children
2. The following event should be sent from the detail page of the item 123: **http://event.yoochoose.net/ebl/00000/click/Js79009234YU7/1/123?categorypath=%2FShoes%2FChildren**

If recommendations provided by the recommendation engine are rendered on this detail page of item 123, the engine should be informed about the rendered recommended items to avoid displaying these recommendations too often. An example of a **rendered** use case is given here:

1. Items 128, 129 and 155 of type 1 are rendered as recommendations for user "Js79009234YU7" on the detail page of item 123.
2. The following event should be sent: **http://event.yoochoose.net/ebl/00000/rendered/Js79009234YU7/1/128,129,155**

If a recommendation is clicked an event should also be sent due to statistical reasons and calculating the conversion or acceptance rate. It is then possible to gain revenue statistics that can be fetched via public REST interfaces. An example of a **clickrecommended** use case is given here:

1. User "Js79009234YU7" clicks on recommended item 155 that was delivered by the call of scenario "also_bought".
2. The following event should be sent: **http://event.yoochoose.net/ebl/00000/clickrecommended/Js79009234YU7/1/155?scenario=also_bought**

After putting items in a shopping basket with the items 128 (one piece and the price of EUR 19.99) and 129 (2 pieces with the price of each EUR 4.44) the user finally decides to buy. In order to track the purchase the following events should be sent. An example of a **buy** use case is given here:

1. User "Js79009234YU7" clicks on 'buy' in the shop
2. The following events should be sent (iterating over the items in the shopping basket) :
    - **http://event.yoochoose.net/ebl/00000/buy/Js79009234YU7/1/128?quantity=1&fullprice=19.99EUR**
    - **http://event.yoochoose.net/ebl/00000/buy/Js79009234YU7/1/129?quantity=2&fullprice=4.44EUR**

Make sure that the buy event is sent from a point where the user cannot abort the process any more (e.g. after verifying the payment or when displaying an order confirmation page).

Advanced Event Use Cases

There are some examples for other events:

1. User "Js79009234YU7" is interested in an item (e.g. he reads a news article or watches a video). For example an article with the id 452
2. The following event should be sent after the threshold defined for consumption is reached: **http://event.yoochoose.net/ebl/00000/consume/Js79009234YU7/1/452**

1. User "Js79009234YU7" likes the item and wants to rate it e.g. with 5 stars (remember that a rating between 0 and 100 is supported)
2. The following event should be sent: **http://event.yoochoose.net/ebl/00000/rate/Js79009234YU7/1/133?rating=5**

1. User "Js79009234YU7" decides that he wants no more recommendations for item 666.
2. The following event should be sent: **http://event.yoochoose.net/ebl/00000/blacklist/Js79009234YU7/1/666**

Transfer Event Use Case

In order to keep track of a user's history a transfer event from anonymous to pseudonymous (e.g. if a user logs in on a website to pay the basket) is offered from YOOCHOOSE. What therefore happens is that all the events he did as an anonymous user are transferred to his 'real' profile. For example all the items he declared as blacklisted in the anonymous session are now transferred to his pseudonymous profile and can be filtered after logging in again on another day or when the session has timed out. An example of a **transfer** use case for the anonymous user id Js79009234YU7 to the pseudonymous with ID of registered user Max Mustermann is given here.

1. The User with the current user ID (Session ID) "Js79009234YU7" logs in and receives a new unique identifier max@mustermann.de
2. The following event should be sent: **http://event.yoochoose.net/ebl/00000/transfer/Js79009234YU7/max%40mustermann.de** Do not forget, the User ID is case sensitive and in case of an email address it should be converted to lower case.