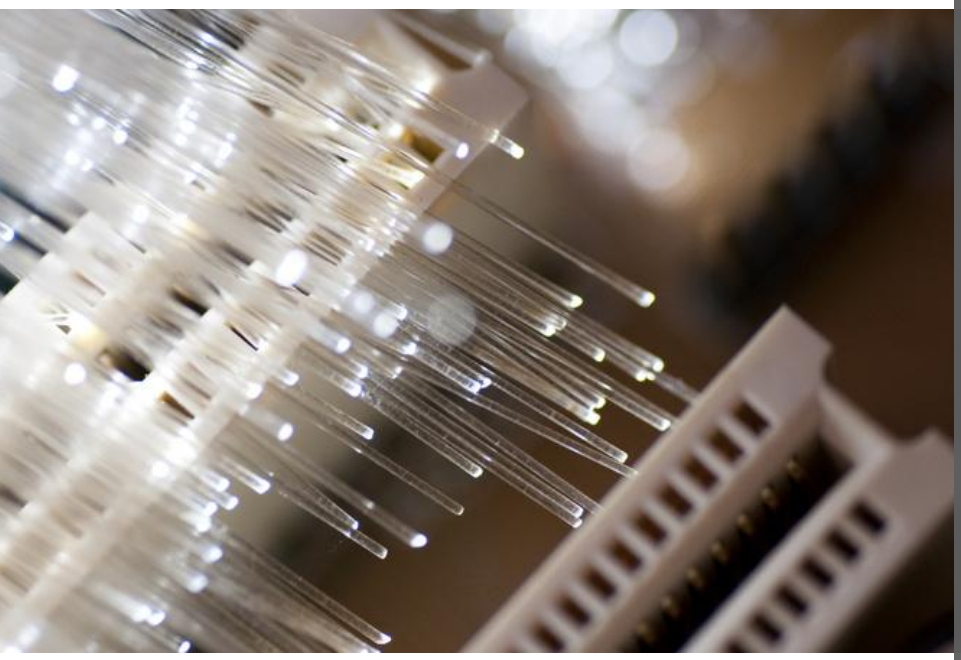
YOOCHOOSE
an eZ Systems Company

eZ Recommendation Extension Manual



Dr. Uwe Alkemper
YOOCHOOSE an eZ Systems Company

Table of contents

About eZ Recommendation extension 4
Installation 5
Requirements 5
Installation procedure 5
Settings in ezrecommendation.ini 6
Class definition and datatype 8
New datatype for recommendation 8
'Recommend' check box 9
'Export content (for recommendation)' check box 10
Item type (for recommendation) selection 10
Time to trigger consumption event value 10
Content export mapping 11
Implement tracking 14
Enable 'Click'-events 14
Create other common events ('owns', 'consume', 'blacklist') 14
'Consume'-event 15
'Rate'-event 16
'Purchase'-event 16
'Delete'-event 16
'Rendered'-event 17
'Transfer user'-event 17
'Click recommended'-event 17
Content export 19
Incremental content export 19
Initial content export 19
Retrieving recommendations 20
The get recommendation call 22
Rendered vs. requested recommendations 23
Workflows 24
Export content after publishing 24
Remove content before delete 25
Create purchase event 26
Statistics 27

Configuration Portal (https://admin.yoochoose.net) 29
Manage account details 29
Create new scenarios 31
Apply filter rules to scenarios 32
Frequently asked Questions 33
Content-delete-after not triggering delete workflow 33
How-to reset recommender statistics 33
Initial content export not exporting all recommendable content 33
Possible conflict with ezstyleeditor extension 33
Recommendation in multiple eZ publish installations 33
Processing requests for statistics data through a *.csv-file 34

About eZ Recommendation extension

The eZ Recommendation extension enables the administration of the eZ Recommendation service and simplifies the integration into existing eZ Publish platforms.

The main value of the extension is to automatically create tracking events from user activities and export relevant content to the eZ Recommendation service. It feeds the cloud-based recommender engine with these data and retrieves on-demand recommendations to be rendered in the eZ Publish frontend.

It contains the following functions:

1. Accesses the YOOCHOOSE cloud-service APIs to retrieve recommendations on demand from the recommendation service
2. Supports publisher as well as online shop recommendations
3. Creates a workflow to export content to recommendation
4. By default this workflow is triggered after publishing of objects marked as recommendable
5. Provides a script to trigger a full export of all existing content marked as recommendable
6. Defines a new Recommendation datatype with a recommend flag
7. Creates tracking events from user activities on objects that are marked as recommendable
8. Creates a workflow for purchase events when products are bought in the webshop
9. Creates a workflow for delete content when a node is deleted
10. Creates a recommendation entry in the support panel to show *statistics*

Installation

Requirements

The extension has been tested with eZ Publish 4.3, 4.4 and 4.5. It is expected to work also with 4.6. The following documentation has been created with an eZ Publish 4.5.

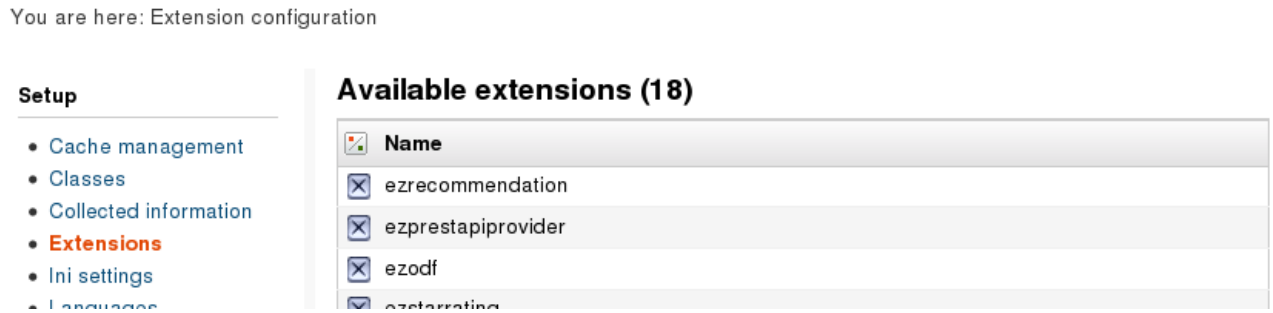
Installation procedure

1. **Install package**

   Extract and copy the ezrecommendation extension into 'extension' folder.

2. **Activate ezrecommendation**

   In the administrator interface, click 'setup' tab->'extensions' menu,



   or in *settings/override/site.ini.append.php*, activate the extension in the 'ExtensionSettings' section:

```
[ExtensionSettings]
ActiveExtensions[]=ezrecommendation
```

3. **Regenerate autoloads**

   In administrator interface, click 'setup' tab->'extensions' menu, select 'ezrecommendation', Click 'Regenerate autoload arrays for extensions'.

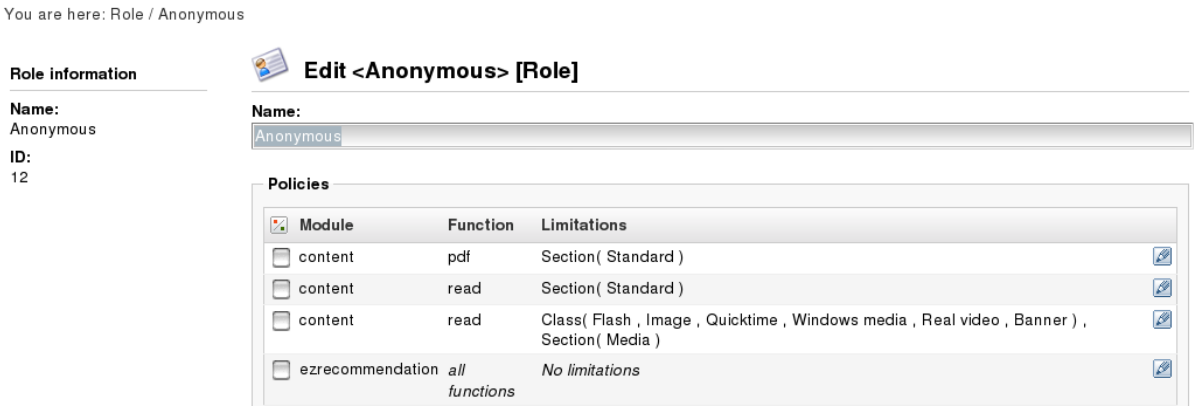
   or in eZ Publish installation folder, execute:

```
php bin/php/ezpgenerateautoloads.php -e
```

4. **Grant access for the ezrecommendation module**

   In the administrator interface, click 'User accounts' -> 'Roles and policies' then select 'Anonymous'. Click on the *role* and edit the *policies*.

   Add new policy. Choose the module 'ezrecommendation' and the function 'request'
   and then 'Grant full access'.



Do the same for the role 'Members' if you have a login area an your site and all other user groups using recommendations.

5. Clear cache

Clear INI and template caches. (from the administration interface use 'Setup' tab or commandline).

Settings in ezrecommendation.ini

`[SolutionSettings]`

The user has to specify here the type of recommendations he wants to use (options are ‘shop’ or ‘publisher’).

`[ShopPriceCurrency]`

The default currency code of the webshop (e.g. EUR or USD)

`[ClientIdSettings]`

Customer ID and License Key you received for your eZ instance from YOOCHOOSE (see chapter ‘Configuration Portal (https://admin.yoochoose.net)‘).

`[RequestSettings]`

If enabled, the answer from the eZ recommendation service will be logged in debug.log.

The following settings should only be modified in consultation with eZ Systems or YOOCHOOSE.

`[URLSettings]`

Settings for the YOOCHOOSE Server URLs.

`[ParameterMapSettings]`

Maps the ez attribute names to the ezrecommendation parameters.

`[SolutionMapSettings]`

Maps the type of the site to ezrecommendation product IDs of YOOCHOOSE.

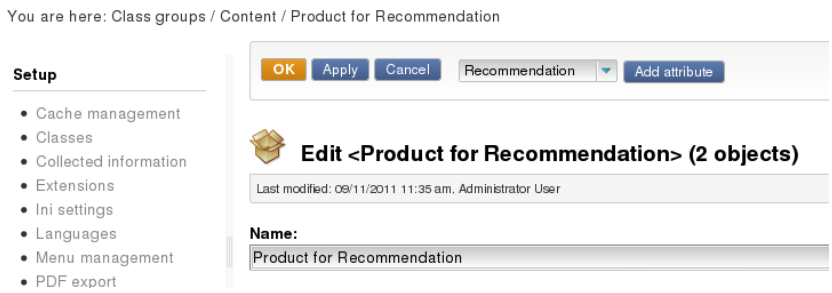
`[ExtensionSettings]`

Defines the response form. Json is supported until now.

Class definition and datatype

New datatype for recommendation

A new datatype has been created for the recommendation service. When you install the eZ Recommendation extension the datatype *Recommendation* is available. You can see it when you edit an existing/new class in the backend of your eZ Publish installation:



You can edit any existing class and add a *Recommendation* entry. This enables this class to be tracked and recommended by the eZ recommender service in the future.

We created new classes including the *Recommendation* datatype named 'Article for recommendation' etc. (see below). The new classes are copies of *Article (main-page), Product* and *image* except that they have additional *Recommendation* data entries. We created one more class named *folder for recommendation*. This class is a copy of the folder class without changes. This serves to create dedicated templates overrides for this class as described later.

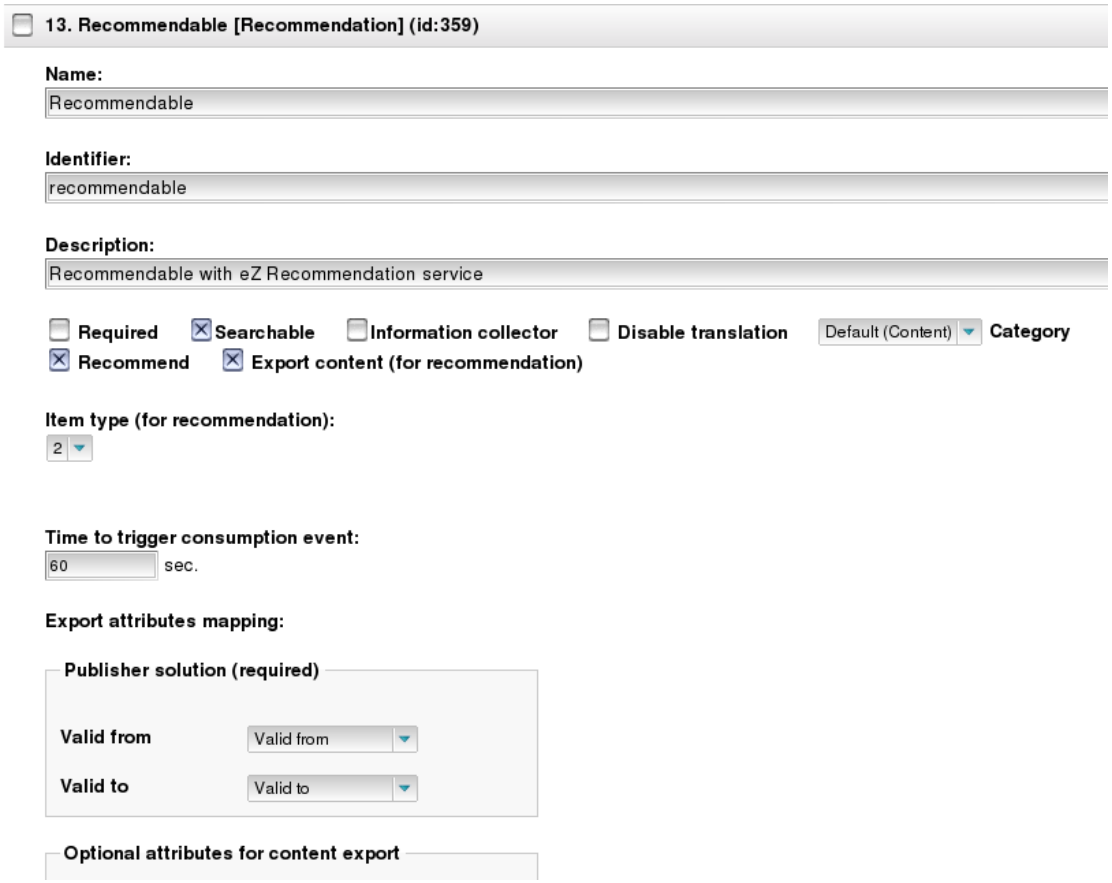
You can see the new recommendation classes in the list of available classes.

Recently modified classes

Name	ID	Identifier	Modifier	Modified	Objects
Product for recommendation	51	product_for_recommendation	Administrator User	02/10/2011 2:23 pm	6
Article for recommendation	48	article_for_recommendation	Administrator User	01/10/2011 4:41 pm	9
Folder for recommendation	52	folder_for_recommendation	Administrator User	30/09/2011 6:27 pm	3
Image for recommendation	49	image_for_recommendation	Administrator User	30/09/2011 6:18 pm	7
Article	16	article	Administrator User	29/09/2011 7:15 pm	29
Article (main-page)	17	article_mainpage	Administrator User	29/09/2011 7:15 pm	0
Article (sub-page)	18	article_subpage	Administrator User	29/09/2011 7:15 pm	0

Edit e.g. *Article for recommendation* (or the class you want to be recommended) and scroll down until you find the recommendation entry in this class.

Comment: The fields in the recommendation entry vary with the solution you selected in the ezyoochoose.ini settings (solution=shop or solution=publisher, see below). The below screenshot is taken with a *pubisher* account.



There are several entries defining how the extension operates, tracks user activities and delivers content information to the eZ Recommender engine. The lower part of the *Recommendation* entry (see below) defines which information about a content object is transmitted to the eZ Recommender engine and how the data is mapped between the eZ Publish class and the datatypes that can be interpreted by the eZ Recommendation service.

'Recommend' check box

It is strongly recommended to check this box in the class definition and make *Recommend* the default for any object in this class. You can later change this flag for each individual object if you do not want to include a specific node into the recommendation service.

Check this box to enable recommendations for all objects of the class as a default. After an object is enabled for recommendation each user activity on this object creates a tracking event. This means that you can only see tracking events after you included recommendation into at least one class and created objects of this class.

If you want to test this you now either create objects of the class *Article for recommendation* or you include the *recommendation* datatype into an existing class of yours e.g. *article (main-page).*

‘Export content (for recommendation)’ check box

It is strongly recommended to check this box in the class definition and make ‘Export content’ the default for any object in this class.

Check this box to enable the transmission of content information like ‘title’, ‘author’, ‘publish date’ to the eZ Recommendation service. Each time you publish a recommendable object an eZ Publish workflow will be triggered (see ‘Export content after publishing‘) checking whether the export content flag is set. If ‘yes’ it will export the below defined meta-data to the eZ Recommendation service.

Item type (for recommendation) selection

The *Item type (for recommendation*) is similar to a class ID in eZ Publish. It defines a number corresponding to a given *type* of content. If you attribute the number “2” as Item type for recommendation to multiple classes e.g. *article (main-page)* and *article (sub-page)* then these both classes are treated as one type by the recommendation service.

When the extension tracks the user activities it will tell the eZ Recommendation service that all these articles are of type “2” and should be treated in the same way. And when you ask for recommendations of the same type you will get node IDs from both classes as an answer.

We recommend to use the following list to choose a suitable number Item type (for recommendation). But it is no requirement to strictly follow this list:

Number	Item type
1	Product
2	Article
3	Image
4	Media
5	User generated content

Time to trigger consumption event value

This parameter is relevant when you want to track if a user reads an article, watches a video or studies a picture. A default value of 20 sec is a good starting point to define the typical consumption time of an article.

When a user stays more than 20 sec on the same page the page will send a so-called ‘consume’-event with the next click of the user (see ‘‘Consume’-event`). There is some analogy to a ‘purchase’-event in an online shop regarding the relevance of such an event to the user profile.

Change this value to a value that suits your content class. You may want to change this value to 5 sec for images and 60 sec for videos. Choose 0 sec to disable 'consume'-events completely for a given class.

Content export mapping

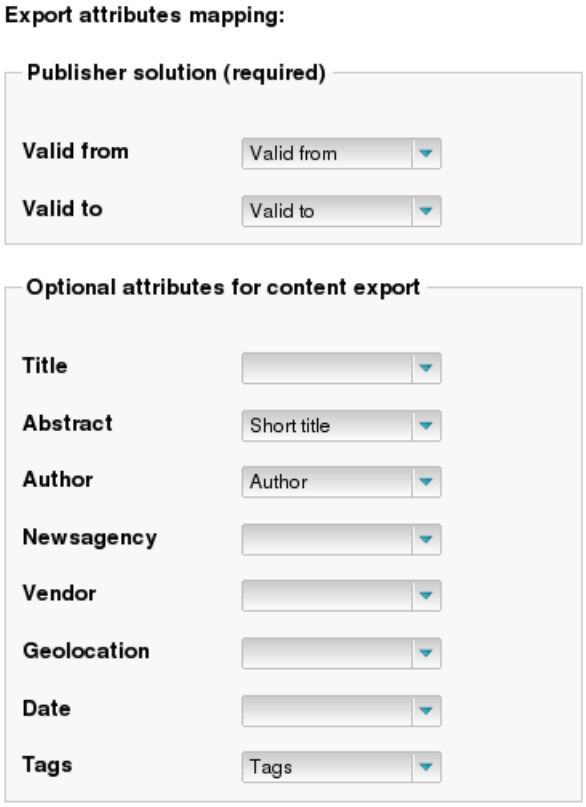
We stated earlier that the recommendation service is a cloud-based service. It uses pre-calculated models that are based not only on the user's activities but also on information about the content of your eZ Publish installation.

Each time you publish some content and the content is marked with 'Export content (for recommendation)' in the class definition the extension sends the defined meta-data to the recommender engine in the cloud. With the content export mapping you define which information is sent to the recommender engine/used for future model calculation.

There are 10 content attributes that are extremely valuable for the calculation of recommendations and filtering of the results. You find the keys to these attributes on the left-hand side of the attribute mapping in your class definition. Two of these attributes are mandatory. But exporting more content information may significantly improve the quality.

Example:

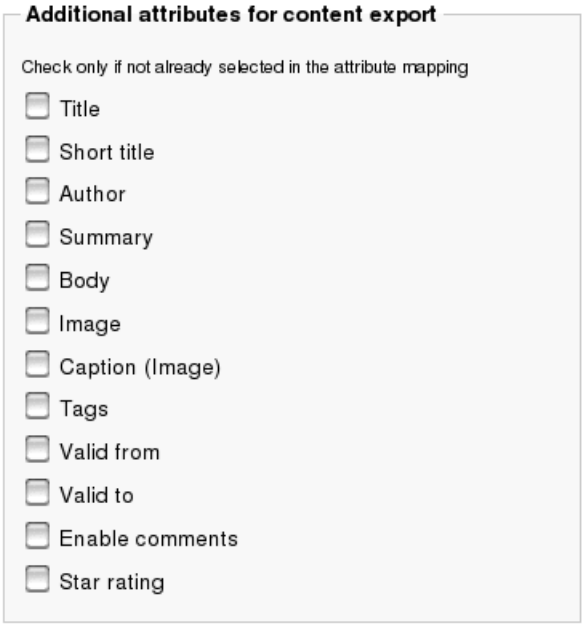
When your individual article class possesses a number of date/time entries like *publish date, unpublish date, created on, validated on* it is important to tell the recommender engine the time window for recommendation of each article (probably between *publish date* and *unpublish date*). In this case you map *publish date* with *Valid from* and *unpublish date* with *Valid to*).



On the right-hand side you can choose in a selection box which of your class attributes you want to map to these recommendation keys.

You may have additional content information that is not interpreted by default like the color or size of a product, the category of an article or the resolution of an image. Such information can be useful to create filters on recommendation results prior to presentation.

In the *Advanced* version of the eZ Recommendation service you can make use of the additional information for such filter purposes. The *Recommendation* datatype therefore allows defining *Additional attributes for content export.*



The user interface shows all class attributes in the list. Please do not check attributes twice which you mapped already to other attributes above. This creates more traffic from redundant content export information only.

Implement tracking

Enable ‘Click’-events

In order to create tracking events you need to do some changes in your template files.

The first step is to include the following code in your pagelayout.tpl. The code must be included in the body of the pagelayout.tpl but outside of any caching block. You can place it e.g. right before the </body> tag:

```
{* enable tracking of click events in pagelayout.tpl *}

{include uri='design:content/ezrecommendation_html.tpl'
content=$module_result track=true()}
```

After this step you will see that ‘click’-events will be created for any node that is marked as recommendable through the class definition. In our case all content of the classes ‘Article for recommendation’, ‘Product for recommendation’ and ‘Image for recommendation’.

‘Click’-events are created in the moment you click on the link to a recommendable object.

You can verify whether an event has been created by checking the ‘debug.log’ in the directory ‘<eZ publish root directory>/var/log’. You will find entries like the following for each tracking event being sent to the eZ recommender engine:

```
[ Sep 11 2011 14:17:52 ]
event.yoochoose.net/news/xxxxx/click/
131574345055084112/2/110?categorypath=%2F2%2F107%2F110
```

If you change from track=true() to track=false() in the above include you will obviously disable tracking for any page.

Create other common events (‘owns’, ‘consume’, ‘blacklist’)

You can create ‘click’, ‘consume’ and other common events also manually by including the corresponding function call in selected view templates.

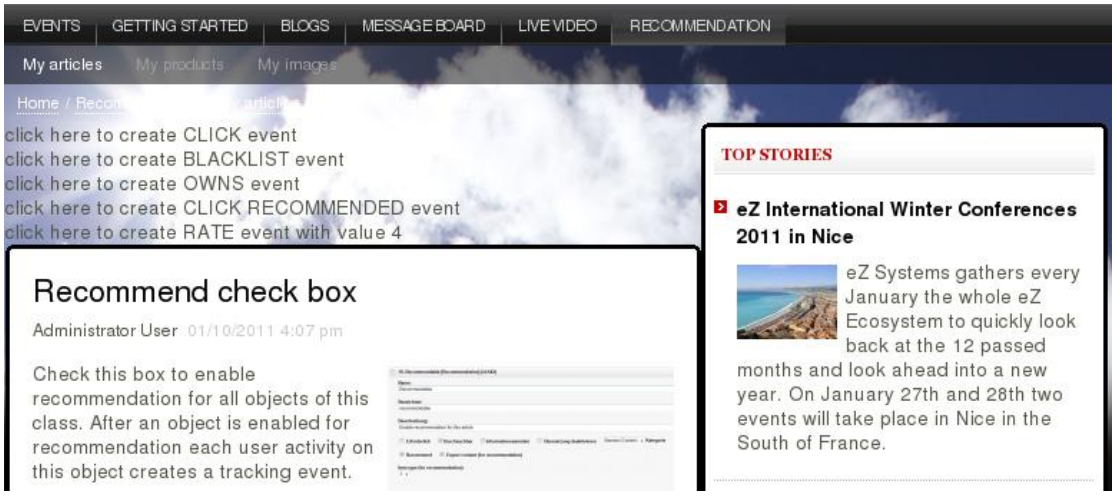
Example:

we copied the full view templates of an article (‘/full/article.tpl’) into our own design and created an override for the class ‘Article for recommendation’. Then we added the following code at the beginning of the .tpl file:

```
{* Article for recommendation - full view *}

<div {generate_common_event($node, 'click')}>
click here to create CLICK event </div>
```

```
<div {generate_common_event($node, 'blacklist')}>
click here to create BLACKLIST event </div>

<div {generate_common_event($node, 'owns')}>
click here to create OWNS event </div>

<div {generate_common_event($node, 'clickrecommended')}>
click here to create CLICK RECOMMENDED event </div>

<div {generate_rate_event($node, 4)}>
click here to create RATE event with value 4</div>
```

After clearing caches and visiting content of the class 'Article for recommendation' in full view you will find a slightly changed view that you can use for temporary testing:



Again you can initiate and test an event type by clicking e.g. on 'BLACKLIST' and check the result in '<eZ publish root directory>/var/log/debug.log':

```
[ Sep 11 2011 15:07:05 ]
event.yoochoose.net/ebl/XXXXX/blacklist
/13157461164148783 5/1/239?categorypath
=%2F2%2F228%2F241%2F

...
```

'Consume'-event

When you want to be notified about consumption of content (e.g. reading an article for more than 20 seconds, watching a video for more than 60 seconds) you need to generate consume events.

To do this you must include the following code into each page that shall be enabled to trigger such an event, e.g. an article for recommendation full view template:

```
<div> {generate_consume_event($node)} </div>
```

This will just create a hidden div with some information in it. After this go in the backend to the *setup* tab and then click on *classes*.

‘Consume’-events are created only in the case when you stay on the page for the pre-defined ‘Time to trigger consumption event’ from the class definition.

Choose the class where you want to configure the *Time to trigger consumption event* attribute and click on edit. Search the recommendation attribute and insert the preferred time in the ‘Time to trigger consumption event’ field.

From now on every time a user uses more than the time to trigger value on a site the ‘consume’-event will be sent to the eZ Recommendation service with the next click.

‘Rate’-event

The ‘rate’- and the ‘purchase’-event need additional parameters to work.

‘Rate’ requires an integer rating value between 0 and 100. An exemplary function call looks like this:

```
< {generate_rate_event($node, 4)}>
click here to create RATE event </div>
```

In the future we will integrate it with the ‘ezstarrating’ extension. Today you have to trigger such an event manually.

‘Purchase’-event

The ‘purchase’-event is triggered after a order has been checked out. Therefore you need to configure the buy event workflow. Follow the chapter ‘Create purchase event‘ to create the workflow and set the triggers.

From now on every time an order is checked out the Recommendation extension will track each product purchased in this order.

‘Purchase’-events require a volume, price and currency. An exemplary function call (in case you do not use the above described workflow) looks like this:

```
<div {generate_buy_event($node, 1, 1199, 'EUR')}>
click here to create BUY event </div>
```

‘Delete’-event

The information that object that is removed in eZ Publish can be automatically sent to the eZ Recommendation service. If this is done, the content will not be recommended in the future after deletion.

Therefore you have to define a workflow and a trigger that initiates the workflow. Follow the chapter ‘Remove content before delete‘ to create the workflow and set the triggers.

‘Rendered’-event

When you present a recommendation more than e.g. 3 times to a user but he never clicks on it he may not be interested in the recommendation. The Recommendation services ‘frequency’-filter allows not showing recommendations more than 3 times to one user in.

How often a recommendation was already rendered is counted from so-called ‘rendered’-events. When you enable the creation on ‘rendered’-events the eZ Recommendation service is informed about each rendering.

No additional code must be integrated. It must be only enabled in the get recommendation call (see ‘The get recommendation call‘).

‘Transfer user’-event

This is another special event that is automatically transmitted when a user logs in. This helps collecting data from anonymous users that belong to one account of a member in an online shop or portal. Without such an event all information about a user is lost when he clears his cookies.

‘Click recommended’-event

This event is of utmost importance if you want to measure the acceptance of your recommendations. In the statistics we use the ratio of ‘Click recommended’-events” divided by the total number of recommendation calls to calculate the so-called conversion rate:

conversion rate = # of ‘click recommended’-events/ # recommendation calls

To create a ‘click recommended’-event you have to do the following:

1) When you retrieve recommendations (see chapter ‘The get recommendation call‘) you set the parameter

```
create_clickrecommended_event=true()
```

2) The following code must be included into an <a>-tag of the templates that you use to show the recommendations (usually the line-view):

```
{if eq($create_clickrecommended_event, true())}
{generate_common_event($node, 'clickrecommended')}{/if}
```

Example (article line view):

```
<div class="content-view-line">

   <div class="class-article float-break">

   <h2><a {if eq($create_clickrecommended_event, true())}
   {generate_common_event($node, 'clickrecommended')}{/if}
```

```
   href={$node.url_alias|ezurl}>{$node.data_map.title.conten
t|wash}
   </a></h2>

...
```

3) In many cases you include an image into the view. In this case you have to inherit the *create_clickrecommended_event* variable and the *node* to the corresponding image.tpl.

Example (article line view):

```
...

   {section show=$node.data_map.image.has_content}

   <div class="attribute-image">

         {attribute_view_gui image_class=articlethumbnail
         href=$node.url_alias|ezurl
         attribute=$node.data_map.image
         create_clickrecommended_event=
         $create_clickrecommended_event ycnode=$node}

   </div>

   {/section}

...
```

4) Finally you have to modify your override of the ‘ezimage.tpl’ in an <a>-tag similar to step 2) and include the code to generate the ‘click recommended’-event. Note that we used the additional variable ‘ycnode’ to inherit the node.

Example (ezimage override):

```
...

{if $href} <a
   {if eq($create_clickrecommended_event, true())}
   {generate_common_event($ycnode, 'clickrecommended')}{/if}
   href={$href}
   {if and( is_set( $link_class ), $link_class )}
   class="{$link_class}"{/if}
   {if and( is_set( $link_id ), $link_id )}
   id="{$link_id}"{/if}
   {if $target} target="{$target}"{/if}
   {if and( is_set( $link_title ), $link_title )}
   id="{$link_title|wash}"{/if}>
{/if}

...
```

Content export

The capability of content export for recommendable content has been enabled in the class definition.

To get into effect we must establish an incremental content export that is triggered with each publishing activity and we must initiate a one-time export of the whole content when the export conditions of a class are changed within the class definition (e.g. you decide to export tags for all *articles for recommendation* and therefore change the class definition).

Incremental content export

Step one is to create a workflow to export content. You find a detailed description about the implementation of the incremental content export workflow in the chapter ‘Incremental content export‘.

Initial content export

When you have an eZ Publish installation with many content objects prior to the installation of the eZ Recommendation extension you need to export the content once. This is done once after you add the *Recommendation* datatype to an existing class. New content is automatically exported incrementally as described above.

Before you can start the initial export go to the ‘ezrecommendation.ini’ and fill in the field "SiteURL" in the block "BulkExportSettings". There you have to fill in the url of your site. If you use a www-dir, than it should be also entered (e.g. SiteURL=http://example.com/wwwdir and BulkPath=extension/ezrecommendation /design/standard/images/).

The second important thing is to make sure you added the *Recommendation* datatype in every class you want to export to the eZ recommender engine. Otherwise the objects from those classes won't be exported.

In the shop solution you should enter the price in every node you want to be exported. In the publisher solution the mapping of a datetime-field to the ‘valid-from’ and ‘valid-to’ attributes is obligatory. Nodes, which do not fulfill these specifications, will not be exported.

The initial export is executed in the terminal. Go to your ez root directory and type in:

```
php extension/ezrecommendation/bin/initialdataexport.php
```

Observe the output of the script for errors or warnings. If the script finishes successfully open the ‘debug.log’ and search for the answer of the eZ Recommendation extension. If you got an error e.g. a 404 answer, make sure that the folder in which the XML was saved (standard is the image folder in the standard design (extension/ezyoochoose/design/standard/images/bulkexport.xml), can be accessed from your browser.

Retrieving recommendations

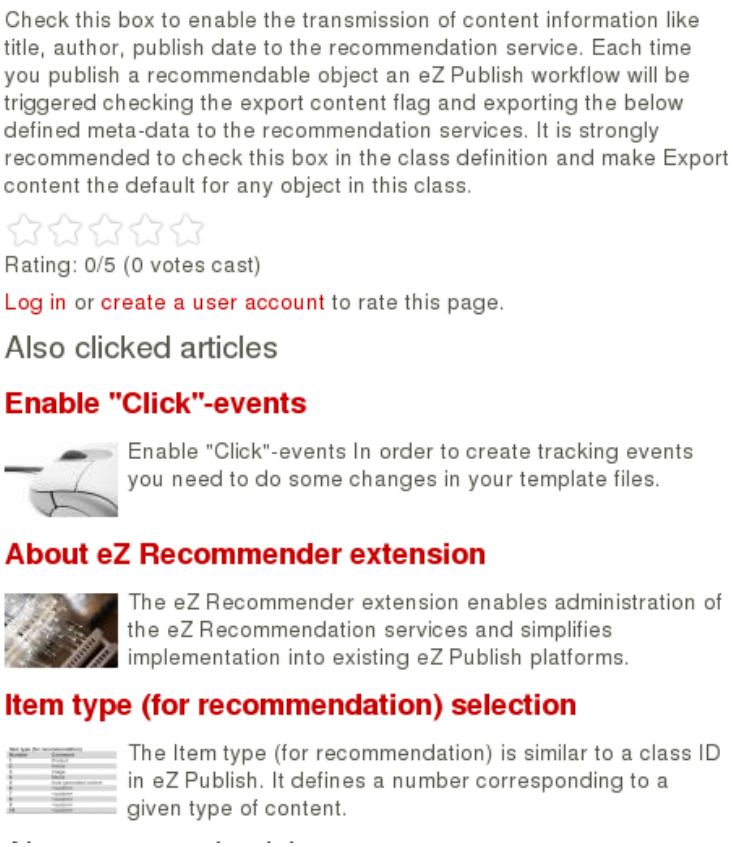
To get recommendations you need to include the following code wherever you want to see the recommendations

```
{include uri='design:content/recommendations.tpl'
node=$node scenario='top_clicked' limit=3
category_based=false() track_rendered_items=true()
create_clickrecommended_event=true()}
```

As an example we included the code in our full view template of ‘articles for recommendation’ between the star rating and the related content.

```
...

<div class="attribute-star-rating">

   {attribute_view_gui
attribute=$node.data_map.star_rating}

</div>

{* Start of eZ Recommendation service *}

<div class="attribute-recommendations">

<h1>{"Also clicked articles"}</h1>

{include uri='design:content/recommendations.tpl'
node=$node scenario='also_clicked_article' limit=3
category_based=false() track_rendered_items=true()
create_clickrecommended_event=true()}

</div>

{* End of eZ Recommendation service *}

   {include uri='design:parts/related_content.tpl'}

...
```

After reloading of the page we see the following:



The top of the above snapsho shows the article full view followed by star rating and recommendations from the eZ Recommendation

service.

Now let us analyze what happens. To do this we check again the latest entries in <eZ publish root directory>/var/log/debug.log.

We start with the request for recommendations:

```
debug.log:

[ Sep 17 2011 16:20:23 ] [ezyoochoose] Trying request
reco.yoochoose.net/ebl/XXXXXX/top_clicked.json?itemid=246
&numrecs=3&itemtypeid=2

[ Sep 17 2011 16:20:24 ] [ezyoochoose] Sending request
reco.yoochoose.net/ebl/XXXXXX/top_clicked.json?itemid=246
&numrecs=3&itemtypeid=2

[ Sep 17 2011 16:20:24 ] [ezyoochoose] Received answer
'HTTP/1.1 200 OK

...
```

If the installation is working properly and recommendations are available you will also see an answer in the following form:

```
[ Nov 17 2011 18:59:29 ] [ezrecommendation] Received
recommendations

{"recommendationResponseList":[{"reason":"CF_I2I_CLICK
(context: ITEM(s))","itemType":2,"itemId":179,"relevance"
:93},{"reason":"CF_I2I_CLICK (context: ITEM(s))",
"itemType":2,"itemId":175,"relevance":91},{"reason":
"CF_I2I_CLICK (context:ITEM(s))","itemType":2,
"itemId":177,"relevance":76}]}'...
```

Scrolling down you will see tracking events showing that some of the recommendations have been rendered:

```
[ Nov 17 2011 18:59:29 ] [ezrecommendation] Sending
request
event.yoochoose.net/news/mycustomerID/rendered/mycookie/2
/179,175,177
```

The get recommendation call

The get recommendation request has been triggered in the above include from recommendations.tpl. This include comes with six parameters. These parameters control the way recommendations are created. They also control what happens after a recommendation is delivered to a portal:

`node=$node`

the node is obviously the node of the currently rendered content. The recommender engine required this information to define the context of the recommendation e.g. for “users who clicked this also clicked …”

`scenario='top_clicked'`

the scenario string must exactly match with the scenario ID in the configuration portal. A detailed description of the portal and how you find out all available scenarios is described in chapter ‘Configuration Portal (https://admin.yoochoose.net)‘. ‘top_clicked’ represents the most popular clicked (recommendable) content. For a complete default list of scenarios check the configuration portal.

`limit=3`

Tells the eZ Recommendation service to deliver 3 recommendations if possible. Depending on the scenario it is possible that the number of requested recommendations cannot always be fulfilled.

`category_based=false()`

This parameter allows you to limit the results to the current branch of the node tree. In the example we set this parameter to false(). In this case the category path (node path) is not sent to the eZ Recommendation service. The current context/position of the user in the website is ignored. Top-clicked articles for

example take into account the whole website. If you change this parameter to true() you will see only those objects that are in the same folder or below.

`track_rendered_items=true()`

This parameter defines what happens when one or more of the recommendations have been rendered. If defined as true() an additional 'rendered'-event is sent to the eZ Recommendation service. If you want to apply a filter to show recommendations no more than e.g. 5 times to the same user you must set this to true() and provide this information to the eZ Recommendation service. The three rendered events in our example vanish if you set this parameter to false().

`create_clickrecommended_event=true()`

When this is defined as true() the template will create a 'click recommended'-event if one of the rendered recommendations is clicked by a user. This allows measuring conversion rates and filling the statistics information with usage and acceptance of recommendations.

Rendered vs. requested recommendations

You ma see only two recommendations although three were requested?

There are two possibilities why this typically happens:

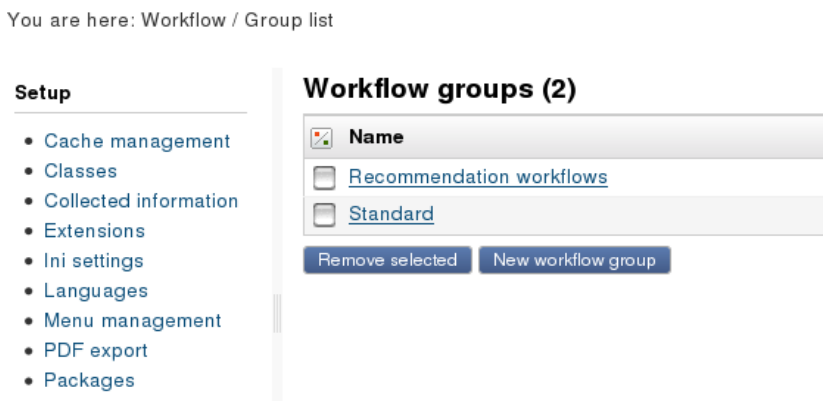
1) The eZ Recommendation service cannot provide enough good recommendations for a given scenario.

2) The resulting node ID cannot be rendered in the current view

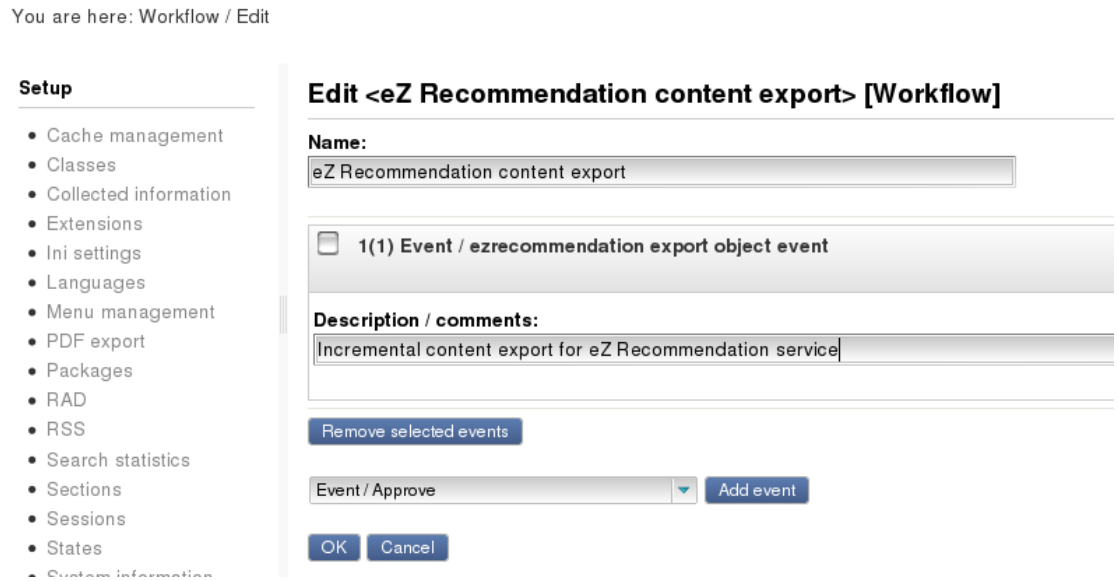
Workflows

Export content after publishing

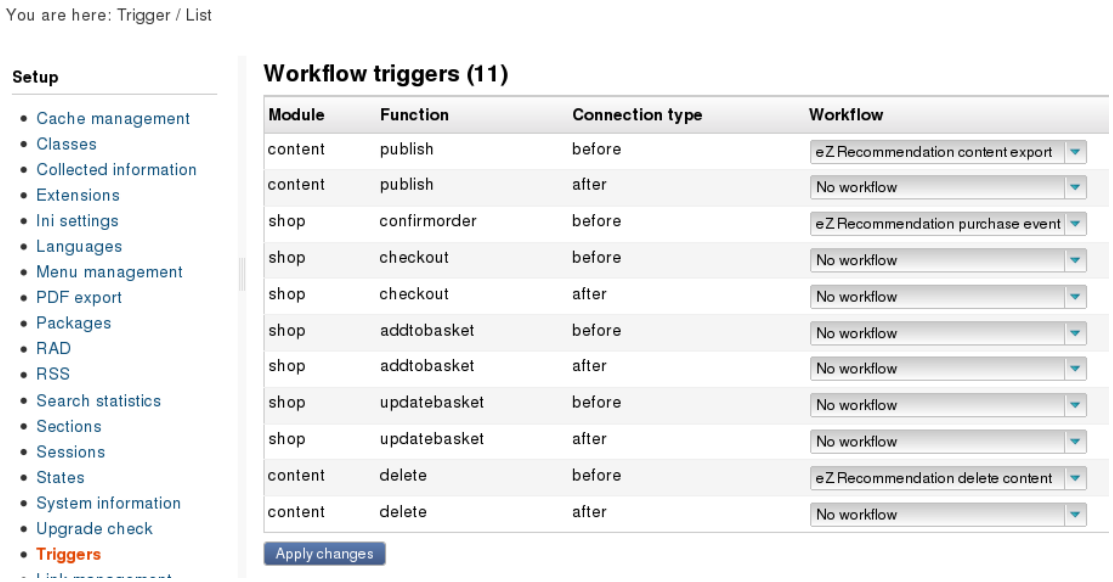
Step one is to create a workflow to export content to the eZ Recommendation service. To do so you create a new Workflow group in Workflows of the Setting-tab. In our example we name it ‘Recommendation workflows’.



Open the workflow group and create a new workflow. In our example we name it ‘eZ Recommendation content export’ and add the event ‘Event / ezrecommendation export object event’ from the drop-down menu.



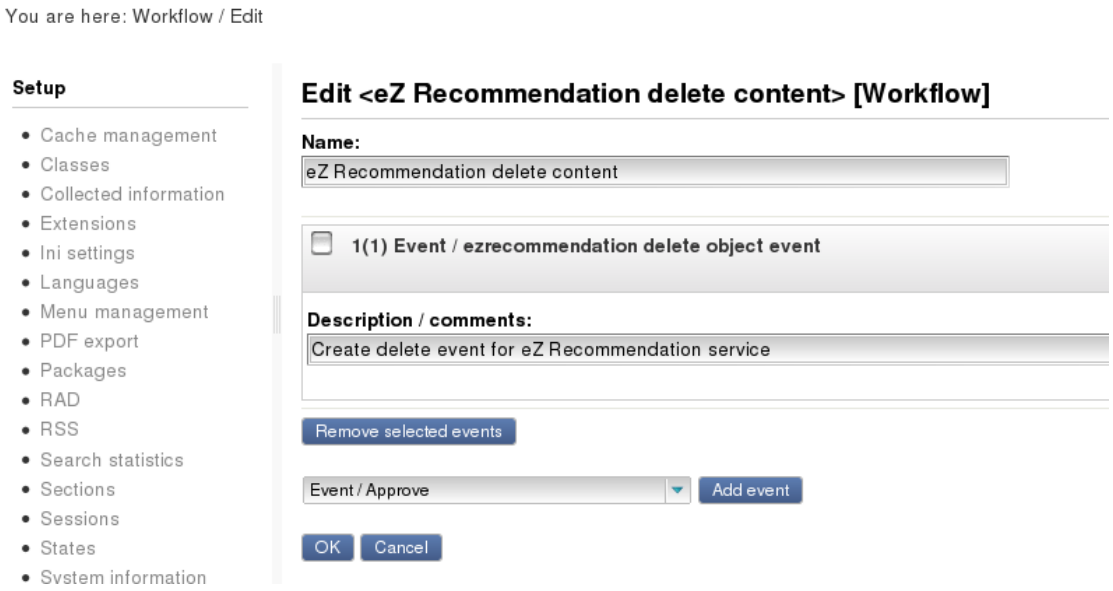
Now that we have created a new workflow we can use in the *Triggers* and apply them each time when content is published. To do so select ‘eZ Recommendation content export’ in the row ‘*content – publish – before’* and apply the changes.



Remove content before delete

We do not want to receive recommendations for content that has already been removed from eZ Publish. The next workflow informs the eZ recommender about each deletion of content in eZ Publish.

Open the workflow group Recommender workflows and create a new workflow. In our example we name it ‘eZ Recommendation delete content’ and select ‘Event / ezrecommendation delete object event’ from the drop-down menu.

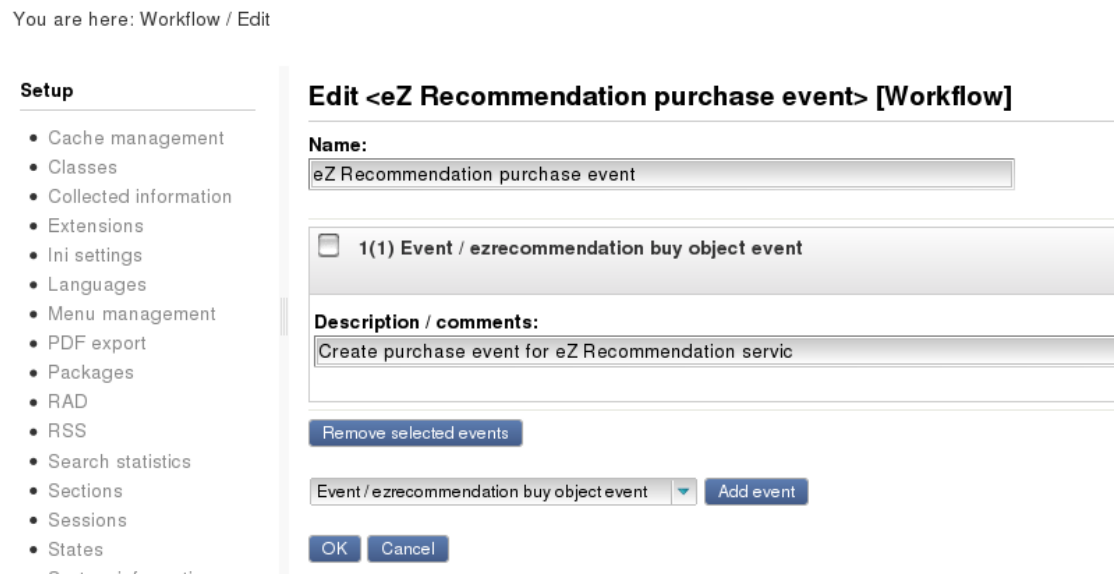


Now that we have created a new we workflow can use in *Triggers* and use it each time when content is deleted. To do so select ‘eZ Recommendation delete content’ in the row *content – delete –* ***before*** and apply the changes.

Create purchase event

This workflow is relevant only if you want to create purchase events each time a shop activity takes place.

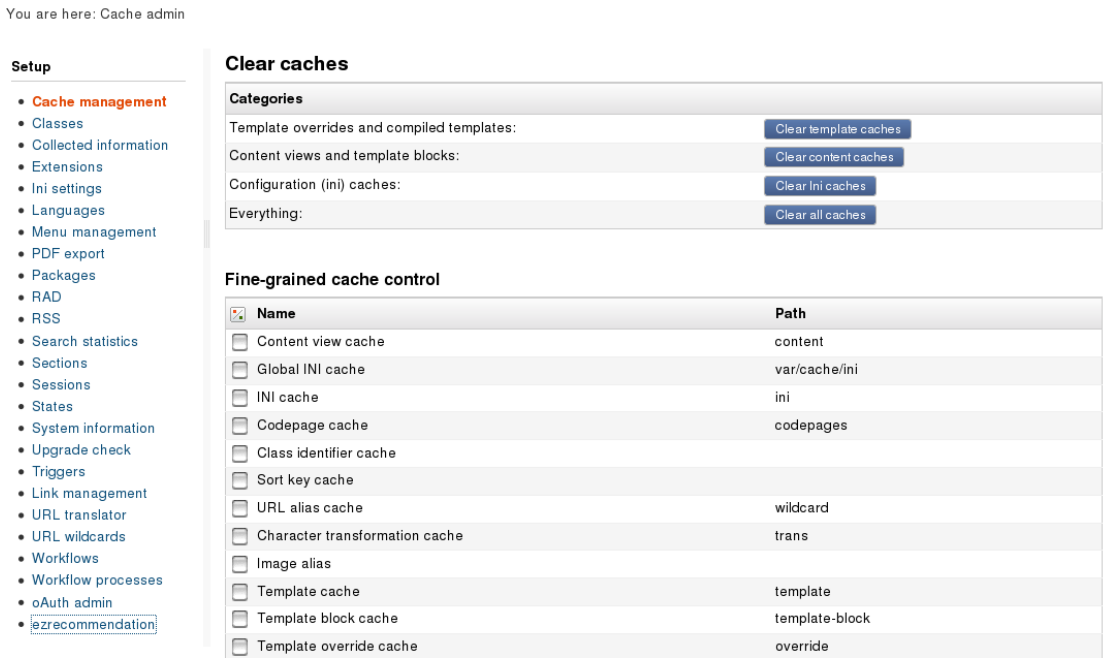
Open the workflow group ‘Recommender workflows’ and create a new workflow. In our example we name it ‘eZ Recommendation purchase event’ and select ‘Event / ezrecommendation buy object event’ from the drop-down menu.



Now that we have created a new workflow we can use it in *Setup->Triggers* and apply it each time when a product is purchased. To do so select ‘eZ Recommendation purchase event’ in the row *shop – confirmorder – before* and apply the changes.

Statistics

You can get a brief statistical information about the eZ Recommendation services usage and results. To see this information in the backend open the new ezrecommendation entry in the left row of the Setup tab.



It provides absolute numbers about

1) Click events that were triggered
2) Purchase events that were triggered (online shop)
3) Delivered recommendations (number of recommendation calls)
4) Recommendations that were clicked by users
5) Recommendations that were purchased
6) Revenue that can be calculated from purchased recommendations and the price of each individual product

ezyoochoose statistics

28.09.2011

revenue:	0
click events:	42
purchase events:	0
delivered recommendations:	191
clicked recommendations:	6
purchased recommendations:	0
currency:	

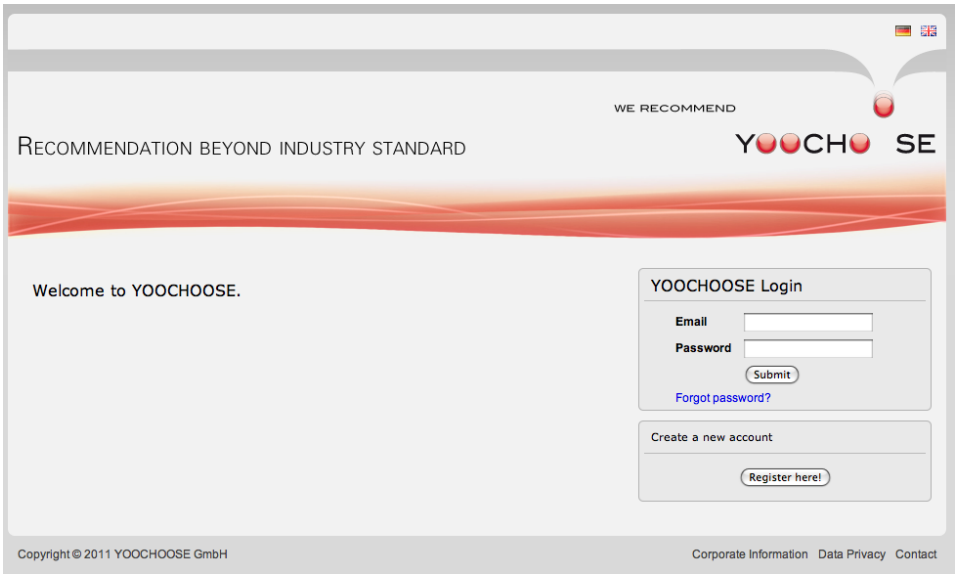
23.09.2011

revenue:	0
click events:	45
purchase events:	18
delivered recommendations:	122
clicked recommendations:	15
purchased recommendations:	0
currency:	

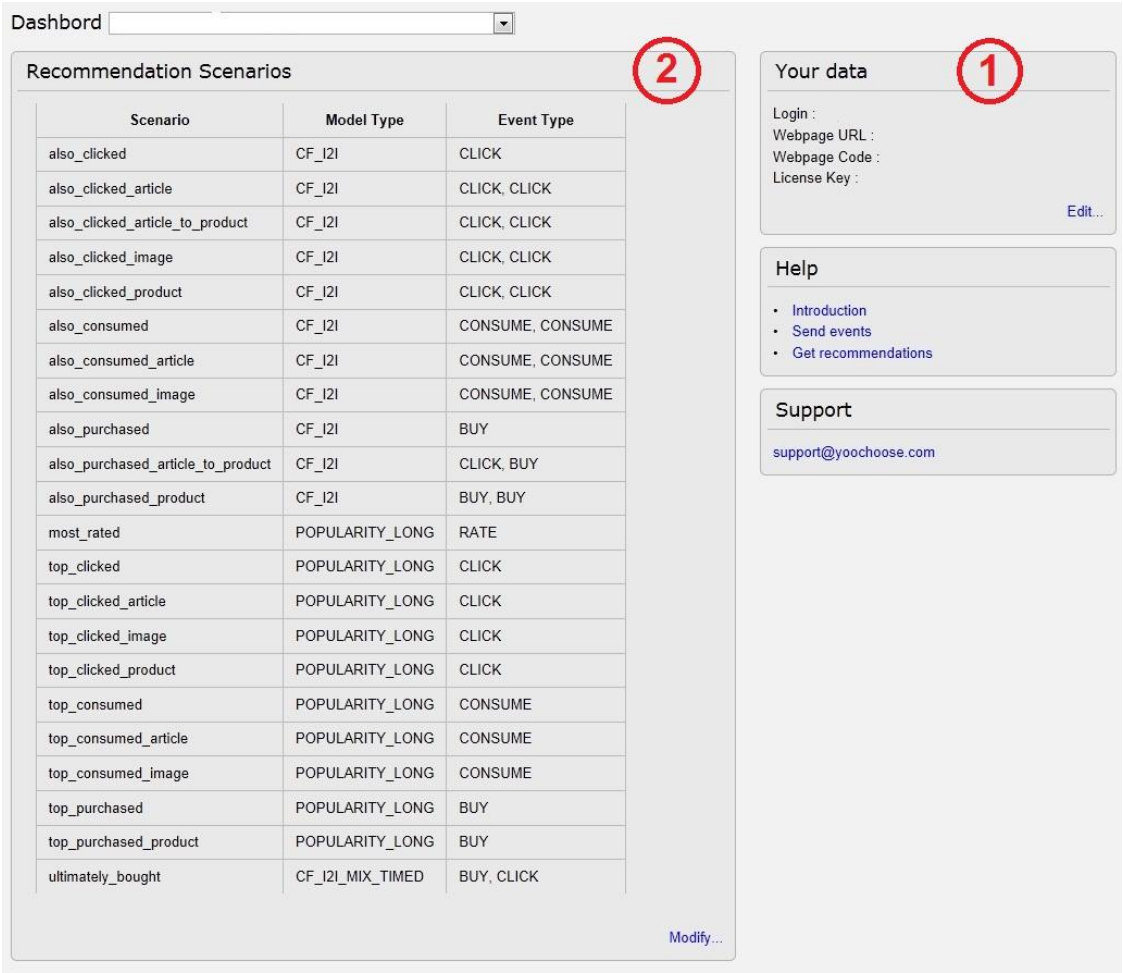
Configuration Portal (https://admin.yoochoose.net)

Manage account details

To manage your account details for the eZ Recommender Service visit the URL https://admin.yoochoose.net You will find the following screen:



To get access to the configuration Portal fill out the required fields *Email* and *Password* with the data you created during registration and click *Submit*. You will be redirected to the Welcome page showing available *Recommendation Scenarios* and *Account Data*.



Under point (**1**) which is marked in the snapshot your will find your account detail information:

- Login: Your Email address used for registration
- Webpage URL: Your website the eZ Recommender service will be connected to.
- Customer ID: Your **Customer ID**
- **License Key:** Your personal authentication code

You need the *Customer ID* and den *License Key* to connect eZ Publish with your Recommendation service account.

Please edit the ezrecommendation.ini file located in the directory <eZ publish root directory>/extension/ezrecommendation/settings:

```
# Insert here your ezrecommendation customer id and your
license key.

[ClientIdSettings]
```

```
CustomerID= your customerID (Webpage Code)”

LicenseKey= your license key
```

“

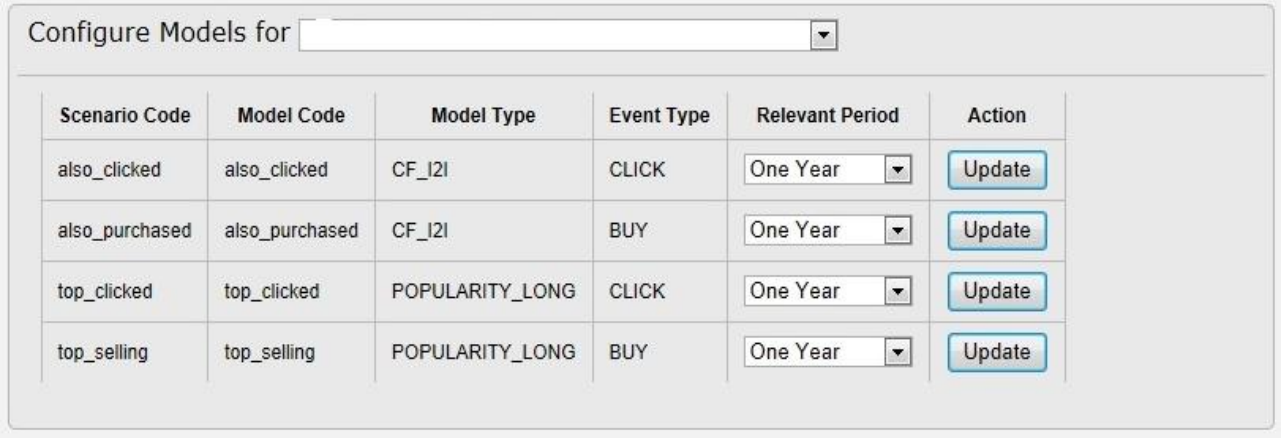
Under point (**2**) which is marked in the snapshot you will find a list of the scenarios which are provided for your recommendation calculation. Most scenarios are self-explanatory e.g.:

- also_clicked
  “Customers who clicked this item, also clicked …”
- also_purchased
  “Customers who bought this product, also bought …”
- top_clicked
  Most clicked items” / “Most popular items”
- top_consumed
  “Most bought products” / “Most popular products bought”
- umltimatly_bought
  “Customers who clicked this item, ultimately bought the following products with probability of e.g. 53% (product 1), 22% (product 2) and 9% (product 3)”

The list of scenarios depends on your type of account. The Recommendation service differentiates between online shops and publishers. It also varies from *Basic* license to *Advanced* license.

Create new scenarios

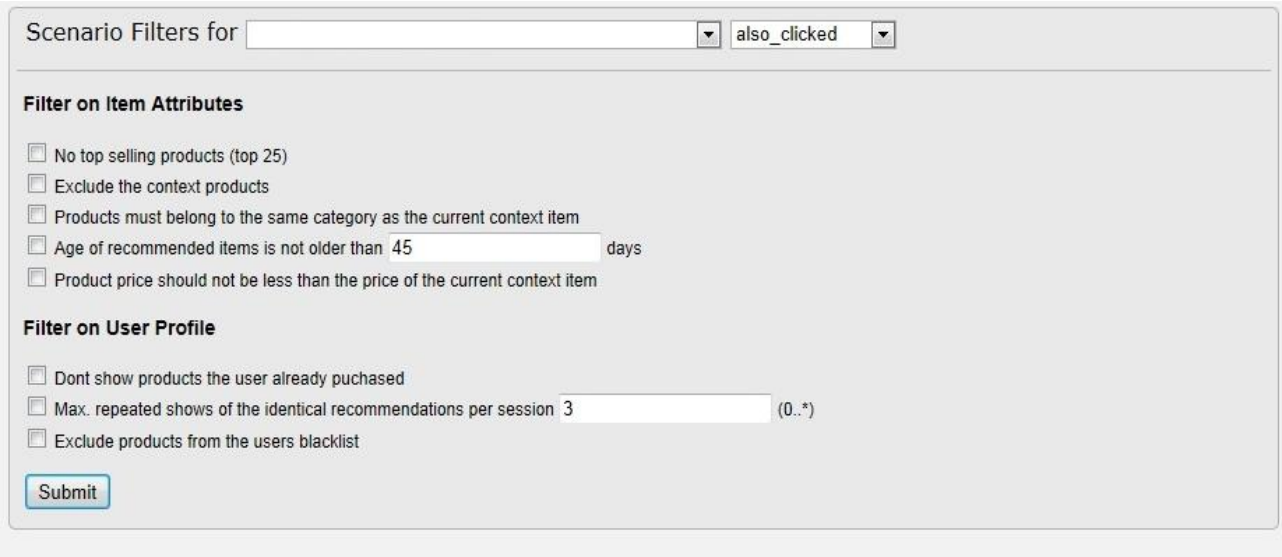
The Basic license allows only changing some parameters in your existing scenarios. With the Advanced license you can create new scenarios for later use and reference in eZ Publish. The screenshot shows an out-of-the-box online shop client with *Basic* license.



Remember: The scenario ID (scenario code) is the link between the scenarios shown here and the eZ Publish recommendation call in your templates.

Apply filter rules to scenarios

Try out the standard filters that you can apply with each of your scenarios. Again, most of the filters are self-explanatory.



Frequently asked Questions

Content-delete-after not triggering delete workflow

Question: I created a workflow to trigger a delete event for the recommendation service after each delete in eZ Publish. This workflow seems not to be working. What can I do?

Answer: The delete workflow requires the trigger content-delete-before. Then it should work.

How-to reset recommender statistics

Question: I played around with the eZ Recommendation service on a test environment. Now I would like to flush all collected events and reset the recommendations. How can I do that?

Answer: Currently the YOOCHOOSE staff can do a reset only. So please send us an e-mail with the customer ID that you want to reset. Then we remove all your past events and inform you. Afterwards you need to create new events to model new recommendations. You will receive new recommendations as soon as new models replace your old models.

Initial content export not exporting all recommendable content

Question: Some of my articles are not exported in the initial export although they are marked as recommendable. Why does that happen?

Answer: Some fields are mandatory when you want to export an object to the eZ Recommendatio service. In the publisher solution it is mandatory to have a *valid from* and *a valid to* (often mapped to publish date/unpublish date) different from 0. In the shop solution the *price* must be different from 0. Fill the *price* or the *valid from/to* fields and the content export should work also for this object.

Possible conflict with ezstyleeditor extension

Question: Tracking does not seem to work in my eZ Publish installation although I followed the installation guide and implemented tracking in the pagelayout.tpl.

Answer: There seems to be a conflict with the ezstyleeditor extension. Try to disable the extension and check the debug.log for click events.

Recommendation in multiple eZ publish installations

Question: I installed the eZ Recommendation extension on my development installation as well as on my productive environment. I get recommendations from my development installation on the productive system and vice versa. How can I change that?

Answer: You should not use one YOOCHOOSE customer ID with multiple installations. If you have a development installation besides a productive installation ask YOOCHOOSE for two separate customer IDs to avoid interference between recommendations.

The second option is to reset the recommender engine after finishing the testing and then use the customer ID for the productive installation only.

Processing requests for statistics data through a *.csv-file

Question: Can I get more statistical information about the eZ Recommendation service usage than provided in the backend.

Answer: It is possible to retrieve your statistical data with a configurable interval through a *.csv file which can be downloaded in the way it is described below. To do so, replace your customer ID in the link, determine the starting/ending time and the granularity which should be used for the time slices. After you edited the link copy/paste it to your browser and confirm your entry.

Example:

https://config.yoochoose.net/ebl/v3/your_customer_id/revenue/summary.csv?from_date_time=2011-10-31T00:00:00&to_date_time=2011-10_31T23:00:00&granularity=PT60M

In the above example the granularity is set up to 60 minutes. It can be changed. The minimum value is 15 minutes.

https://config.yoochoose.net/[Product]/[API_version]/[your_customer_id]/revenue/summary.csv?from_date_time=[start_time(ISO8601)]&to_date_time_[end_time(ISO8601)]&granularity=PT[Steps in minutes]M