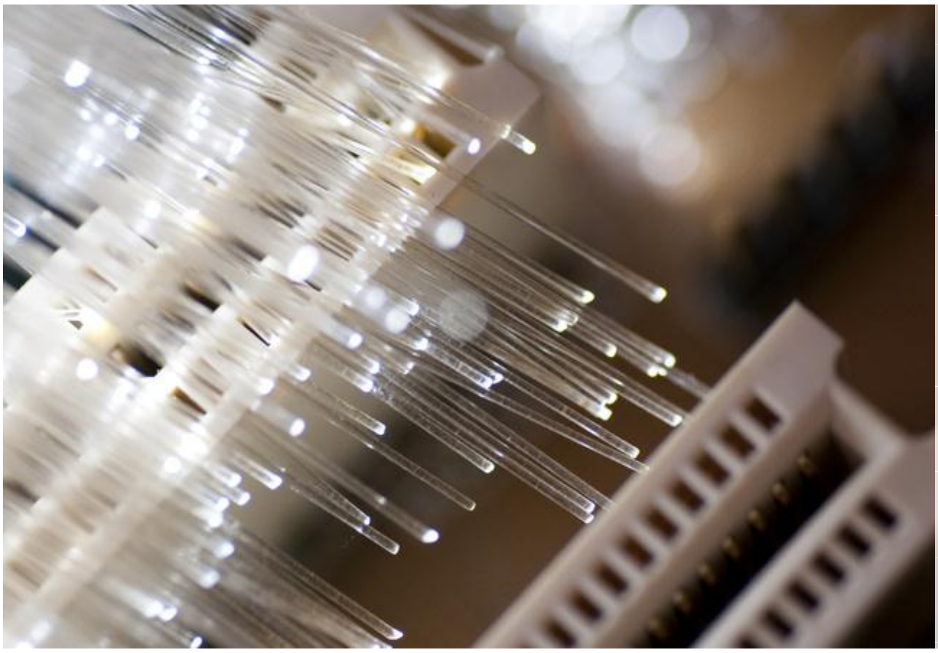
2013

eZ Recommendation Extension manual



David Ben Shimon

YOOCHOOSE - an eZ Systems Company

01.01.2013

Table of Contents

About eZ Recommendation extension

The eZ Recommendation extension enables the administration of the eZ Recommendation service and simplifies the integration into existing eZ Publish 4.7 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 this 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 data type 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 web-shop.
9. Creates a workflow for delete content when a node is deleted
10. Creates a recommendation entry in the support panel to show statistics

This is a documentation which evolved from the first implementation. It is usually kept up-to-date but check https://confluence.ez.no/display/EZRECO for latest updates on the extension development before opening a support request !

1 Enabling the extension

1.1 Requirements

The extension has been tested with eZ Publish 4.7and expected to work also with previous versions. The following documentation has been created with an eZ Publish 4.7 where site package eZ demo with content was installed.

1.2 Installation procedure

1.2.1 Import and install the extension

The eZ Recommendation extension comes as a standard extension folder. The folder should un-zipped, have the name ‘ezrecommendation’ and must be located in the extension directory inside the root folder of your eZ publish installation. For example if your root directory is called MySite then the extension folder should be as illustrated in figure 1. This is all resides in your apache root web directory.

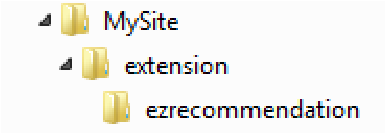
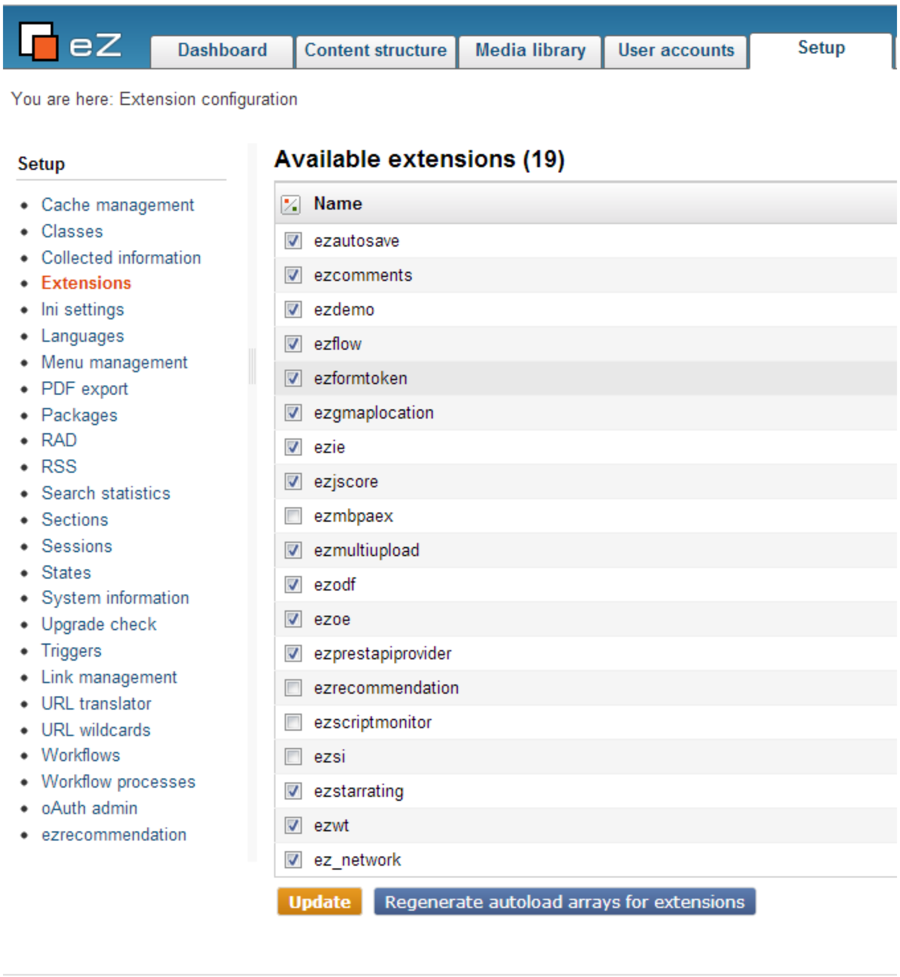


Figure 1. Folders structure of the extension

1.2.2 Activate ezrecommendation

Once the extension is located in place, go to your admin web interface (i.e. http://localhost/ezdemosite/index.php/ezdemo_site_admin/content/dashboard) and choose the setup tab. On the left menu choose the extensions. Your screen should look like the following:



You are here: Extension configuration

Available extensions (19)

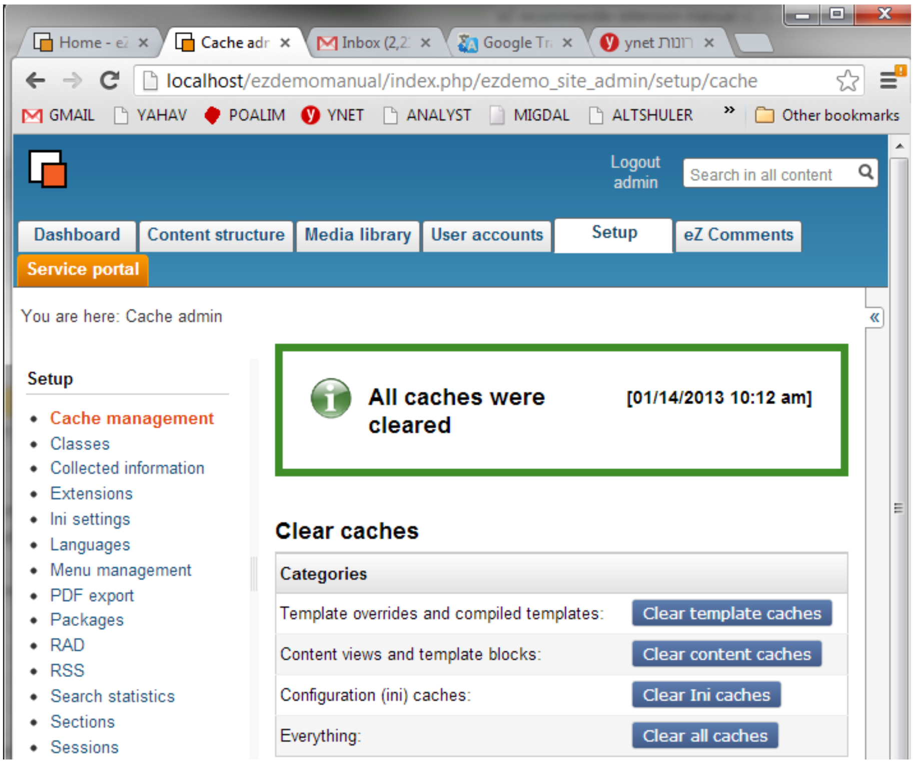
☑	Name
☑	ezautosave
☑	ezcomments
☑	ezdemo
☑	ezflow
☑	ezformtoken
☑	ezgmaplocation
☑	ezie
☑	ezjscore
□	ezmbpaex
☑	ezmultiupload
☑	ezodf
☑	ezoe
☑	ezprestapiprovider
□	ezrecommendation
□	ezscriptmonitor
□	ezsi
☑	ezstarrating
☑	ezwt
☑	ez_network

Figure 2. Activate the extension

Then check the box next to ‘ezrecommendation’ under the available extensions and click the button Update and then afterwards click the button ‘Regenerate autoload arrays for extensions’.

1.2.3 Clear the cache

There is a need to clear the cache after activating the extension. On the left menu at the admin GUI chose the “Cache management” and then click the button “clear all caches” on the right. Figure 3 shows present a snapshot of the related screen.



You are here: Cache admin

All caches were cleared [01/14/2013 10:12 am]

Clear caches

Categories	
Template overrides and compiled templates:	Clear template caches
Content views and template blocks:	Clear content caches
Configuration (ini) caches:	Clear Ini caches
Everything:	Clear all caches

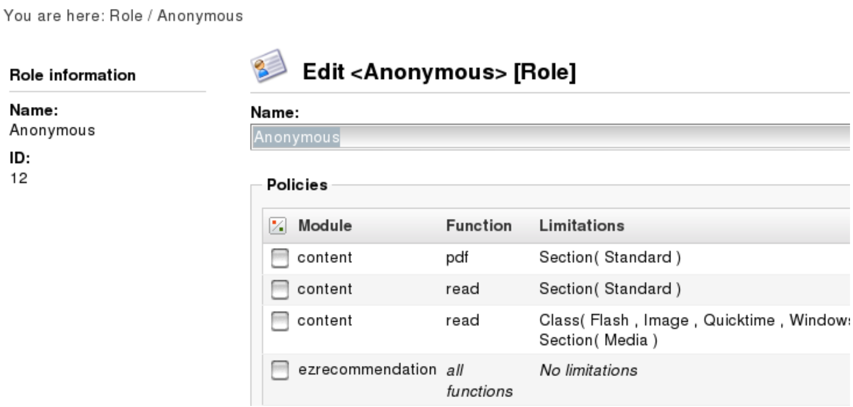
Figure 3. Clear the chace

1.2.4 Grant access for the ezrecommendation module

In the administrator interface, click on 'User accounts’ menu tab and then Click on 'Roles and policies’ on the left menu. In order to allow access to recommendations module for Anonymous role do as follows;

1. Select 'Anonymous' role.
2. Click on the “edit” button
3. Click on the “new policy” button
4. In the “Module” combo box choose “ezrecommendation”
5. In the “Function” combo box choose request.
6. Then click on the button “Grant access to all functions”
7. Click the save button

Apply the same above process for role “Member” and for all other role groups that you wish to provide recommendations. Usually Anonymous and Member are enough. Figure 4 shows a snapshot of the admin GUI while granting access to recommendations for the Anonymous role.



You are here: Role / Anonymous

Role information

Name:
Anonymous

ID:
12

Edit <Anonymous> [Role]

Name:
Anonymous

Policies

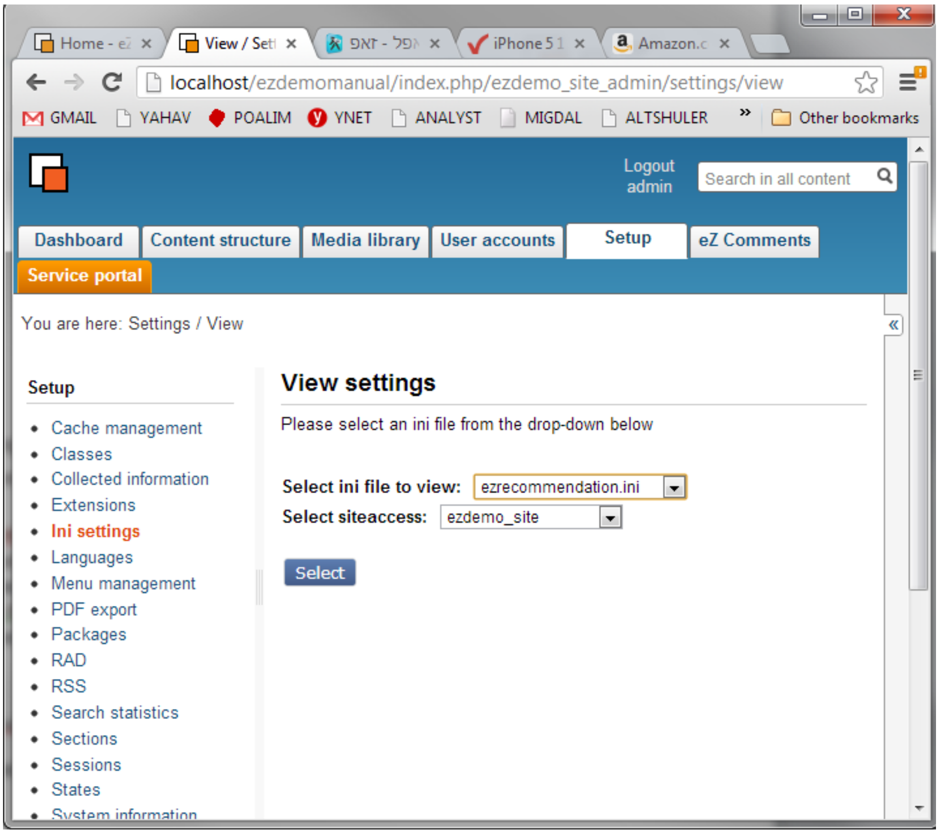
Module	Function	Limitations
□ content	pdf	Section(Standard)
□ content	read	Section(Standard)
□ content	read	Class(Flash , Image , Quicktime , Window Section(Media)
□ ezrecommendation	*all functions*	*No limitations*

Figure 4. Granting the service to roles

At the end of granting access there is a need to **clear the cache**. See „Clear the cache“ section above for instructions.

1.3 Settings in ezrecommendation.ini.append.php

We now start the configuration of the ini settings of the extension. There is an option to edit those settings directly in the file and there is also an option to do it via the admin GUI. We will follow the admin GUI. Click on the “Setup” tab and then click on the “ini settings” link on the left menu. Your screen should look like figure 5.



You are here: Settings / View

Setup

- Cache management
- Classes
- Collected information
- Extensions
- Ini settings
- Languages
- Menu management
- PDF export
- Packages
- RAD
- RSS
- Search statistics
- Sections
- Sessions
- States
- System information

View settings

Please select an ini file from the drop-down below

Select ini file to view: ezrecommendation.ini

Select siteaccess: ezdemo_site

Select

Figure 5. .ini setting view in the admin GUI

In the “Select .ini file to view” combo box choose the ezrecommendation.ini and in the “select site access” combo box choose the site you wish to apply the recommendations; in our case it is the ezdemo_site which comes as part of the eZ publish 4.7 installation. Then click on the “select” button and you should be inside the ini settings page of the extension. All the configuration of the extension is done via this page. Following are description of some of the configuration parameters.

- **SolutionSettings** – The user has to specify here the type of recommendations he wants to use (options are ´shop´ or ´publisher´) depend on the website activities. The default value is publisher.

- **ShopPriceCurrency** – The currency code of the webshop (e.g. EUR or USD). The default value is EUR.

- **ClientIdSettings** – Customer ID and License Key you received for your eZ instance from YOOCHOOSE - **see chapter**

- **Appendixes**

- **Configuration Portal (https://admin.**yoochoose.net) **at the appendixes of this document**. This is crucial as there are no default values for it. Once the CustomerID and the license key is on hands, click the edit icon (✎) of the CustomerID field, and then in the next screen, change the value in the “Change setting type” combo box from String to ‘Numeric’ then write the CustomerID in the “Setting value” text box on the bottom of the page. Now click the save button and you will go back to the previous ini settings page. Do the same for the LicenseKey beside the change of the setting type combo box; here leave the String value in the “Change setting type” combo box in place and fill in the LicenseKey in the text box at the bottom of the page. Click the save button to save the change and to go back to the main ini setting page of the extension.

- **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.

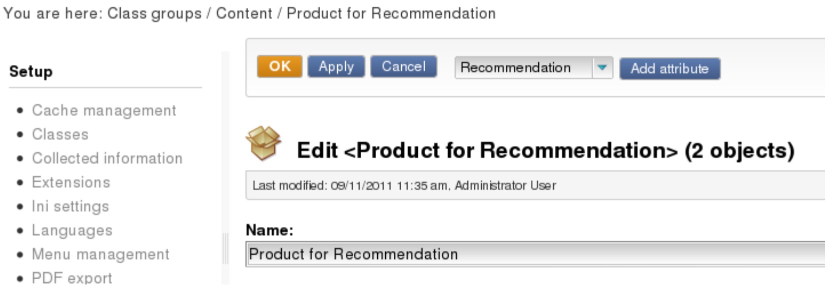
- **SolutionMapSettings** - Maps the type of the site to ezrecommendation product IDs of YOOCHOOSE.

- **ExtensionSettings** - Defines the response form. Json is supported until now.

2 Class definition and datatype

2.1 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 under the Setup Tab:



You are here: Class groups / Content / Product for Recommendation

Edit <Product for Recommendation> (2 objects)

Last modified: 09/11/2011 11:35 am, Administrator User

Name:
Product for Recommendation

Figure 6. Adding attribute to a class

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.

The Recommendation datatype has two mandatory attributes – the publish_date and the unpublish_date. Those attributes enable the proper presentation of items in your system, so the recommendation service will not provide you outdated recommendations. Thus, any class you wish to be able to recommend must have publish_date and unpublish_date attributes, so those attributes could be referenced from the recommendation attribute datatype.

Additionally there are new classes already including the *Recommendation* datatype for demonstration purposes. The new classes are copied from *article (main-page), product* and *image* except that they have additional *Recommendation* data entries. A package with these classes is located in the extensions doc folder. 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.

Now, for enabling the recommendations and view the integrated examples over Article objects from Class Group Content do as follows:

1. Click on classes under setup tab.

2. In the Class groups list choose the Content class group. You will now be able to see all the classes which exist inside the Class Content.

3. Click on the Article class.

4. Scroll down the page and then click on the edit button.

5. Scroll down the page again, and in the combo box choose the Date and Time attribute type. Click the add attribute button. Another attribute has been added in edit mode.

6. Fill in the fields Name the String Unpublish date and in the identifier field the String unpublish_date. Then click the ok button. The attribute unpublish_date just added to the Article class.

2.2 Adding the recommendation entry in the proper class

Comment: The fields in the recommendation entry vary with the solution you selected in the ini settings of the ezrecommendation extension (`solution=shop` or `solution=publisher`, see below). The below screenshot and example is taken with a *publisher* account, which is actually the default.

Inside the Article class (the class we want to recommend items from) add a Recommendation attribute. Figure 8 illustrate the screenshot of this specific attribute. Now we fill up the values for the attribute even though default values are already there.

1. Fill in the fields Name the String Recommendable and in the identifier field the String recommendable.

2. **Recommend check box** - Check the recommended check (by default its checked) box to enable recommendations for all objects of the class. 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.

3. **Item type (for recommendation)** - 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 so the recommendations service can identify the activities related the items from this type. 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 answer from comprising items from that class. Chose the value 2 for demo purposes. We recommend using the following list to choose a suitable number Item type (for recommendation). But it is no requirement to strictly follow this list:

Table 1. Definition of data types for item recommendations

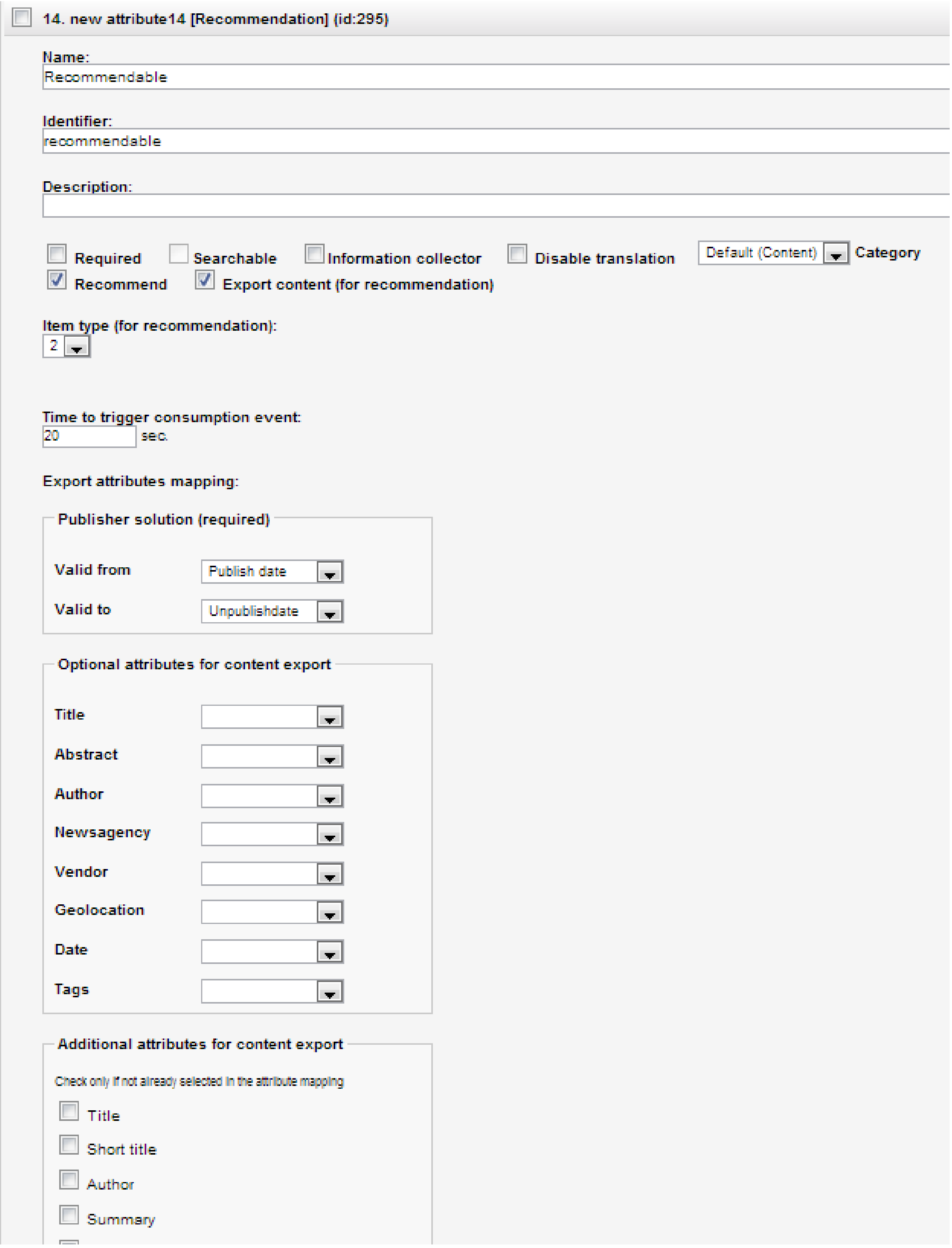
Number	Item type
1	Product
2	Article
3	Image
4	Media
5	User generated content

4. **Time to trigger consumption even -** 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.

5. **Publisher solution -** 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.

6. **Optional attributes for content export**- those are attributes you wish to export to the recommendations service on each item. This is for later processing by the recommendations service and for supporting different variations of models and different types of recommendations, e.g content based recommendations. Map each metadata attribute to the related attribute within your class.

7. **Additional attributes for content export –** here you check only if not already selected in the attribute mapping at the previous optional attributes step. The user interface shows all class attributes in the list. Please does not check attribute that you already mapped in the previous step. This creates redundant sending of information from your web site to the recommendations service. Generally, 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. Those attributes are extremely valuable for the calculation of recommendations and filtering of the results.



□ 14. new attribute14 [Recommendation] (id:295)

Name:
Recommendable

Identifier:
recommendable

Description:

□ Required □ Searchable □ Information collector □ Disable translation Default (Content) Category

☑ Recommend ☑ Export content (for recommendation)

Item type (for recommendation):
2

Time to trigger consumption event:
20 sec.

Export attributes mapping:

Publisher solution (required)

Valid from	Publish date
Valid to	Unpublishdate

Optional attributes for content export

Title	
Abstract	
Author	
Newsagency	
Vendor	
Geolocation	
Date	
Tags	

Additional attributes for content export

Check only if not already selected in the attribute mapping

□ Title
□ Short title
□ Author
□ Summary

Figure 7. Recommend check box attribute

3 Implement tracking

3.1 Enable ‘Click’-events

3.1.1 Default integrated tracking

By default, click events are tracked automatically. Inside the file:

Webroot-dir\ezdemowebsite\extension\ezrecommendation\design\ezdemo\templates\pagelayout.tpl the following code snippet is exists.

```
<!-- Recommendation code: START -->
{* enable recommendation tracking of click events in pagelayout.tpl *}
     {include uri='design:content/ezrecommendation_html.tpl'
     content=$module_result track=true()}
<!-- Recommendation code: END -->
```

Once the extension is activated and the recommendation attribute is properly added to a given class (i.e. Article), then the click events over the items from this class are automatically sent to the recommendation service. That all suggests that tracking over of the click events on the suggested class items is automatically integrated. Those ´Click´ events are created in the moment you click on the link to a recommendable object and in our show case they will be sent whenever you click on item which is from Article type.

Consume event is also integrated automatically for the Article class in our example. However, the code for it is in the file:

Webroot-dir\ezdemowebsite\extension\ezrecommendation\design\standard\override\templates\full\article.tpl

This file is extending the pagelayout.tpl for article objects which already comprising the click tracking. Therefore click and consume events are automatically integrated for Article.

In addition to that, in the above mentioned article.tpl the following code is exists:

```
{*
     <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>
*}
```

By uncommenting those lines and clicking on the relevant div in an article page, we can trigger the relevant events. Thus gives you a hint on how to trigger different events.

3.1.2 Customized design and integrated tracking

If you have a customized pagelayout.tpl template then there is a need to add the tracking code manually. In that case you will need to include it in your relevant pagelayout.tpl. For a default ezdemo installation with content the location of the file you need to edit is here:
extension\ezrecommendation\design\ezdemo\templates

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 under any page which generated from your customized template.

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[1] like the following for each tracking event being sent to the eZ recommender engine depending on the chosen event:

```
[ Sep 11 2011 14:17:52 ]
event.yoochoose.net/news/<CustomerID>/click/1315743450550841122/2/110
?categorypath=%2F2%2F107%2F110
```

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

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

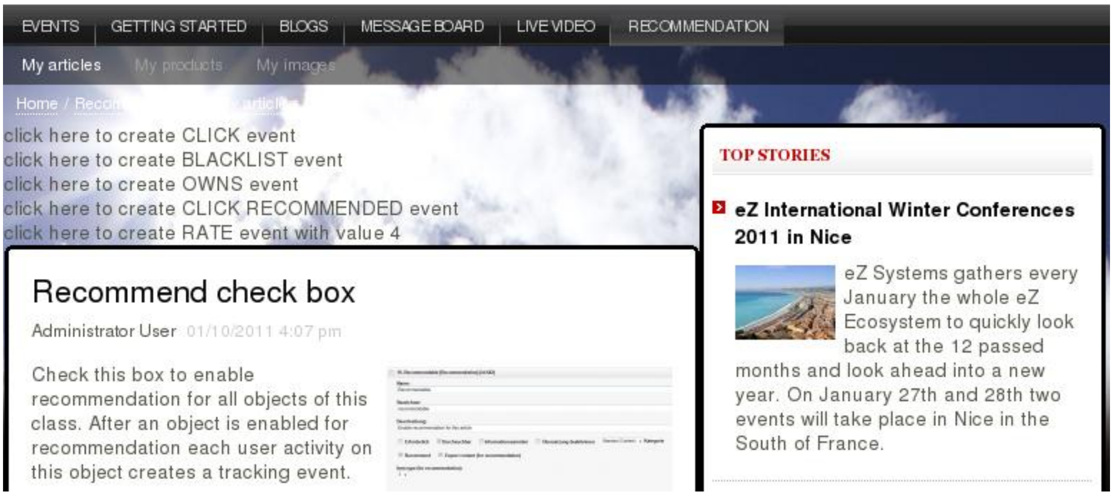
You can create ´click´, ´consume´ and other common events also manually by including the corresponding function call in a selected view template.

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:

[1] Go to the FAQS at the end of this manual if you fail to view the logs

```
{* 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:



click here to create CLICK event
click here to create BLACKLIST event
click here to create OWNS event
click here to create CLICK RECOMMENDED event
click here to create RATE event with value 4

Recommend check box

Administrator User 01/10/2011 4:07 pm

Check this box to enable recommendation for all objects of this class. After an object is enabled for recommendation each user activity on this object creates a tracking event.

TOP STORIES

eZ International Winter Conferences 2011 in Nice

eZ Systems gathers every January the whole eZ Ecosystem to quickly look back at the 12 passed months and look ahead into a new year. On January 27th and 28th two events will take place in Nice in the South of France.

Figure 8. Recommend check box snapshot

Replace the picture with ezdeom picture

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/news/<CustomerID>/blacklist
/131574611641487835/1/239?categorypath =%2F2%2F228%2F241%2F

...
```

3.2.1 ‘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.

3.2.2 ´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.

3.2.3 ´Purchase´-event

The 'purchase'-event is triggered after an 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. The price must be the smallest unit of a currency and can only be an integer. 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>
```

It will send out an event which informs the recommender engine, that one item defined by the node id was bought for 1199 cents which means 11,99 EUR.

3.2.4 ´Delete´-event

The information that objects 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.

3.2.5 ´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 a user.

How often a recommendation was already rendered is counted by ´rendered´-events. If you enable the creation on ´rendered´-events the eZ Recommendation service is informed about each rendering.

No additional code needs to be integrated for sending rendered events. It only has to be enabled in the get recommendation call (see ´The get recommendation call´).

3.2.6 ´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 an anonymous user is lost when he clears his cookies.

3.2.7 ´Click recommended´-event

This event is of outmost 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

The ‘click recommended’ event is sent by default, if a recommended object was clicked.

If you want to customize or use it differently, you can also add it manually. All you need to do is 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.content|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.

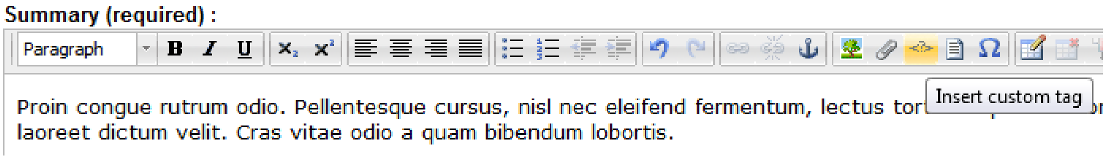
Example (ezimage override):

```
...

{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}

...
```

4 Retrieving recommendations

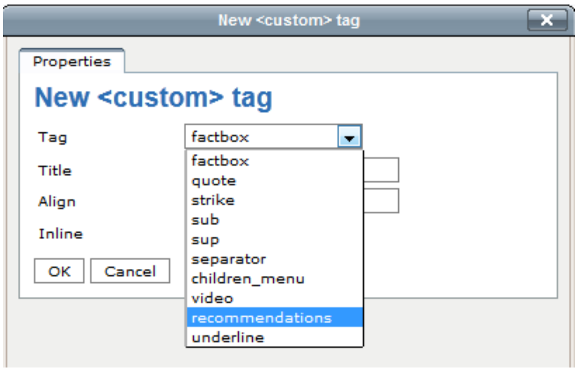
In order to get recommendations, a custom tag needs to be inserted in an article. This can be done in the editor view of an article by simply adding it in the WYSIWIG Editor. While editing an item click on “insert custom tag”



Summary (required) :

Proin congue rutrum odio. Pellentesque cursus, nisl nec eleifend fermentum, lectus tor
laoreet dictum velit. Cras vitae odio a quam bibendum lobortis.

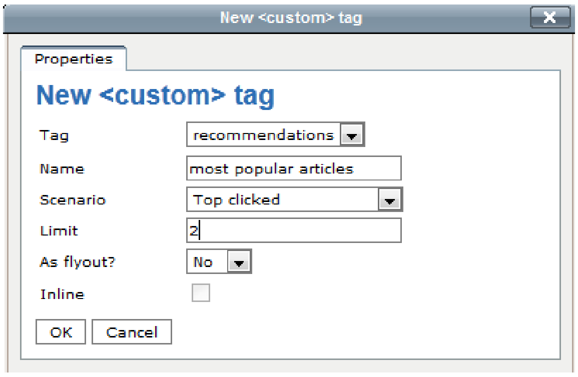
and select “recommendations” from the option-box.



New <custom> tag

Tag: factbox

Select the corresponding recommendation scenario (see also under https://admin.yoochoose.net) and if you want it to be rendered inside the article or as a flyout.



New <custom> tag

Tag: recommendations

Name: most popular articles

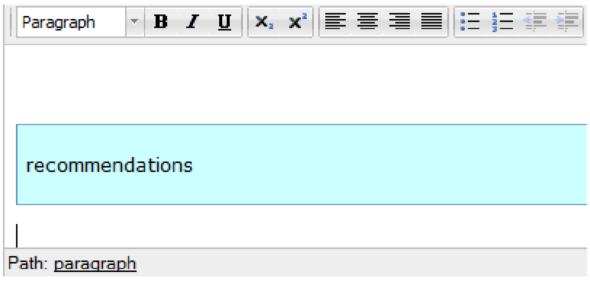
Scenario: Top clicked

Limit: 2

As flyout?: No

Inline: □

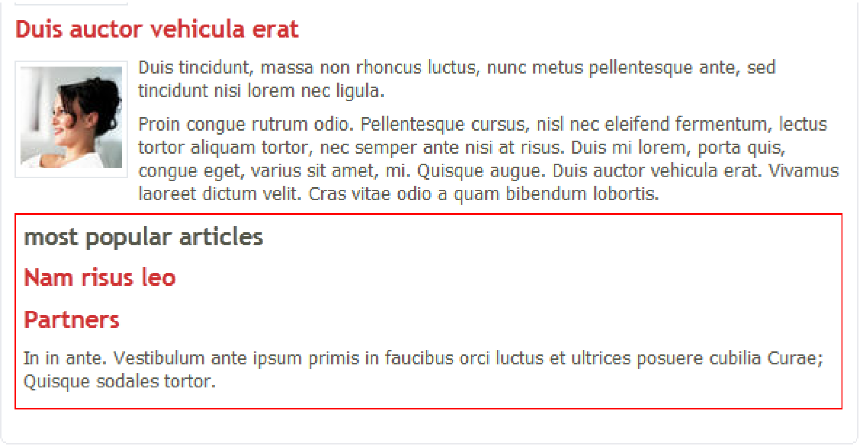
The following placeholder should appear in the editor window



recommendations

Path: paragraph

After publishing and reloading the page you will see recommendations if there's enough tracking events available.



Duis auctor vehicula erat

Duis tincidunt, massa non rhoncus luctus, nunc metus pellentesque ante, sed tincidunt nisi lorem nec ligula.

Proin congue rutrum odio. Pellentesque cursus, nisl nec eleifend fermentum, lectus tortor aliquam tortor, nec semper ante nisi at risus. Duis mi lorem, porta quis, congue eget, varius sit amet, mi. Quisque augue. Duis auctor vehicula erat. Vivamus laoreet dictum velit. Cras vitae odio a quam bibendum lobortis.

most popular articles

Nam risus leo

Partners

In in ante. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Quisque sodales tortor.

If you want to customize the embedding of recommendations you can also follow these steps:

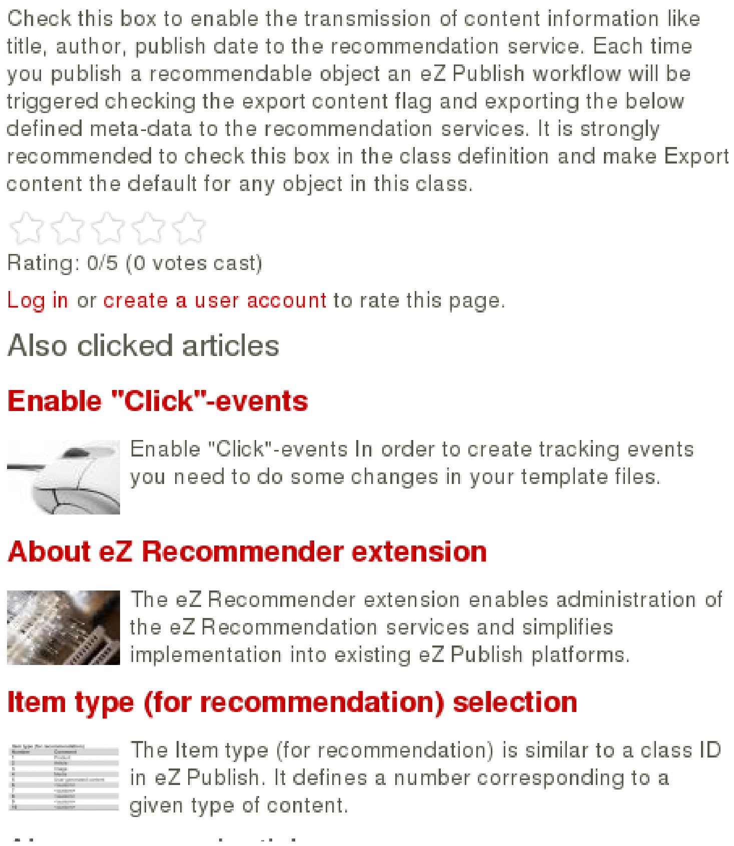
Include the following code wherever you want to see the recommendations

```
{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:



Check this box to enable the transmission of content information like title, author, publish date to the recommendation service. Each time you publish a recommendable object an eZ Publish workflow will be triggered checking the export content flag and exporting the below defined meta-data to the recommendation services. It is strongly recommended to check this box in the class definition and make Export content the default for any object in this class.

☆☆☆☆☆

Rating: 0/5 (0 votes cast)

Log in or create a user account to rate this page.

Also clicked articles

Enable "Click"-events

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

About eZ Recommender extension

The eZ Recommender extension enables administration of the eZ Recommendation services and simplifies implementation into existing eZ Publish platforms.

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.

The top of the above screenshot shows the article full view followed by a star rating and recommendations from the eZ Recommendation service.

To analyze the recommendation workflow in more detail, 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/<CustomerID>/top_clicked.json?itemid=246&numr
ecs=3&itemtypeid=2
[ Sep 17 2011 16:20:24 ] [ezyoochoose] Sending request
reco.yoochoose.net/ebl/<CustomerID>/top_clicked.json?itemid=246&numr
ecs=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/2/179,175,17
7
```

4.1 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 ´

5 Appendixes

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.

5.1 Rendered vs. requested recommendations

You may see only two recommendations although three were requested?

There are two possibilities why this typically happens:

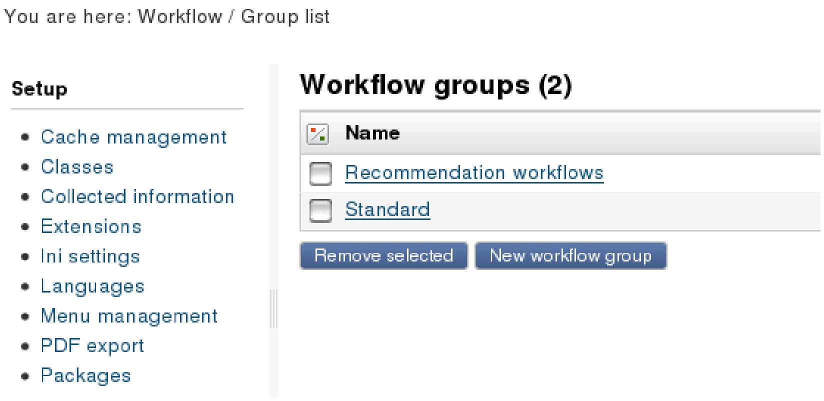
1) The eZ Recommendation service cannot provide enough good recommendations for a given scenario.

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

6 Workflows

6.1 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 the Admin Interface. In our example we name it ´Recommendation workflows´.

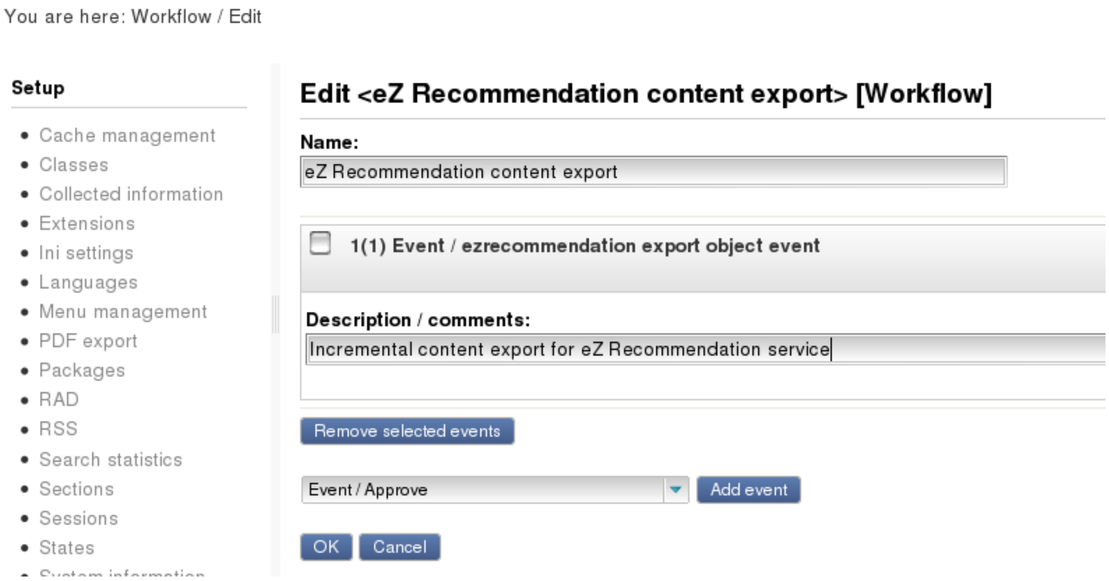


You are here: Workflow / Group list

Workflow groups (2)

Name
□ Recommendation workflows
□ Standard

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.



You are here: Workflow / Edit

Edit <eZ Recommendation content export> [Workflow]

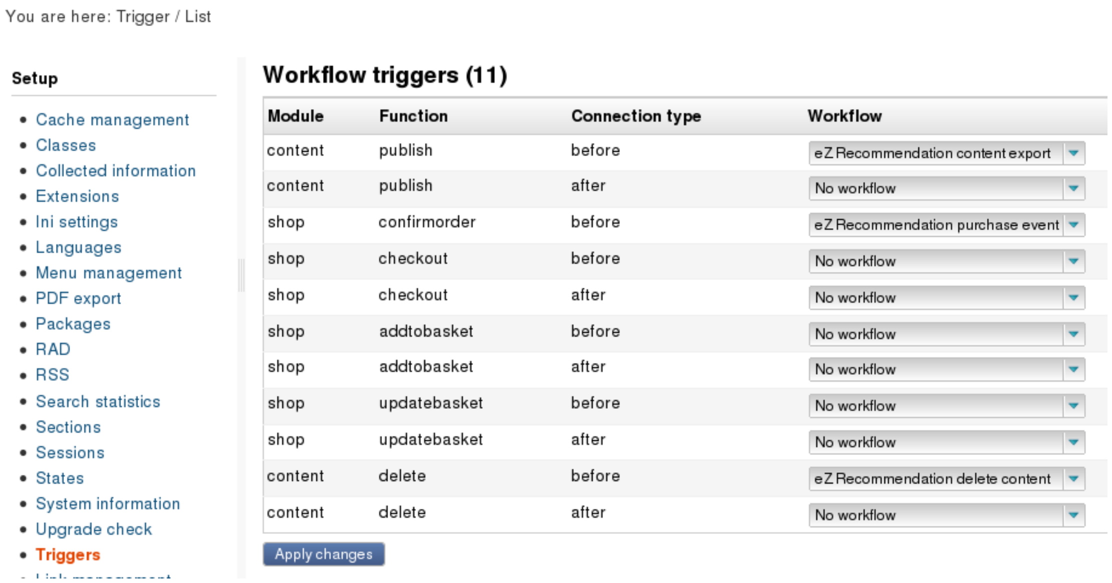
Name: eZ Recommendation content export

□ 1(1) Event / ezrecommendation export object event

Description / comments: Incremental content export for eZ Recommendation service

Event / Approve

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.



You are here: Trigger / List

Setup

- Cache management
- Classes
- Collected information
- Extensions
- Ini settings
- Languages
- Menu management
- PDF export
- Packages
- RAD
- RSS
- Search statistics
- Sections
- Sessions
- States
- System information
- Upgrade check
- Triggers
- Link management

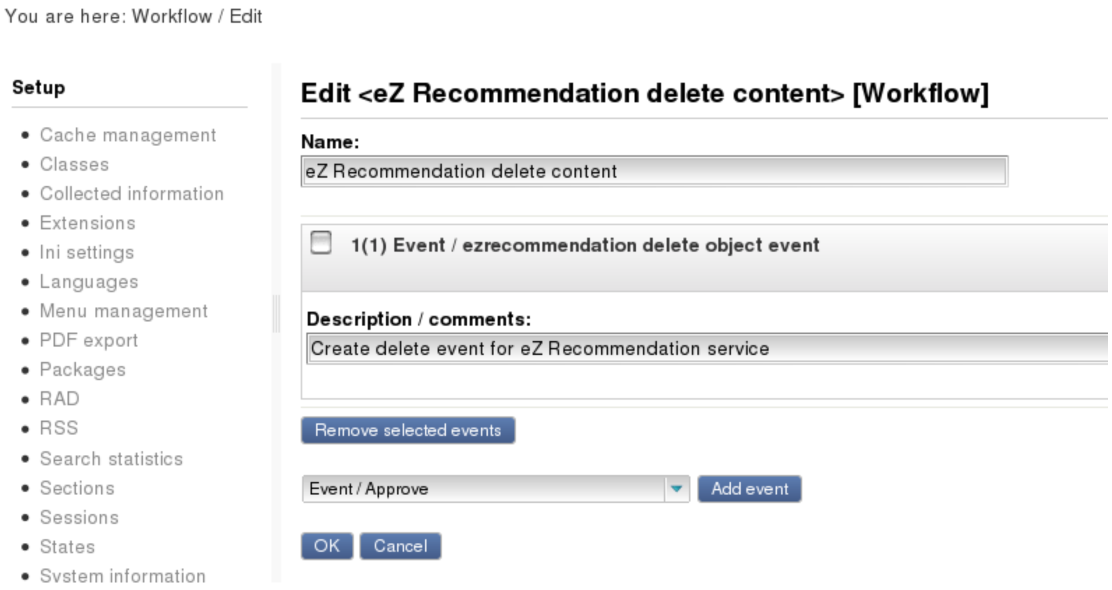
Workflow triggers (11)

Module	Function	Connection type	Workflow
content	publish	before	eZ Recommendation content export
content	publish	after	No workflow
shop	confirmorder	before	eZ Recommendation purchase event
shop	checkout	before	No workflow
shop	checkout	after	No workflow
shop	addtobasket	before	No workflow
shop	addtobasket	after	No workflow
shop	updatebasket	before	No workflow
shop	updatebasket	after	No workflow
content	delete	before	eZ Recommendation delete content
content	delete	after	No workflow

6.2 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.



You are here: Workflow / Edit

Setup

- Cache management
- Classes
- Collected information
- Extensions
- Ini settings
- Languages
- Menu management
- PDF export
- Packages
- RAD
- RSS
- Search statistics
- Sections
- Sessions
- States
- System information

Edit <eZ Recommendation delete content> [Workflow]

Name: eZ Recommendation delete content

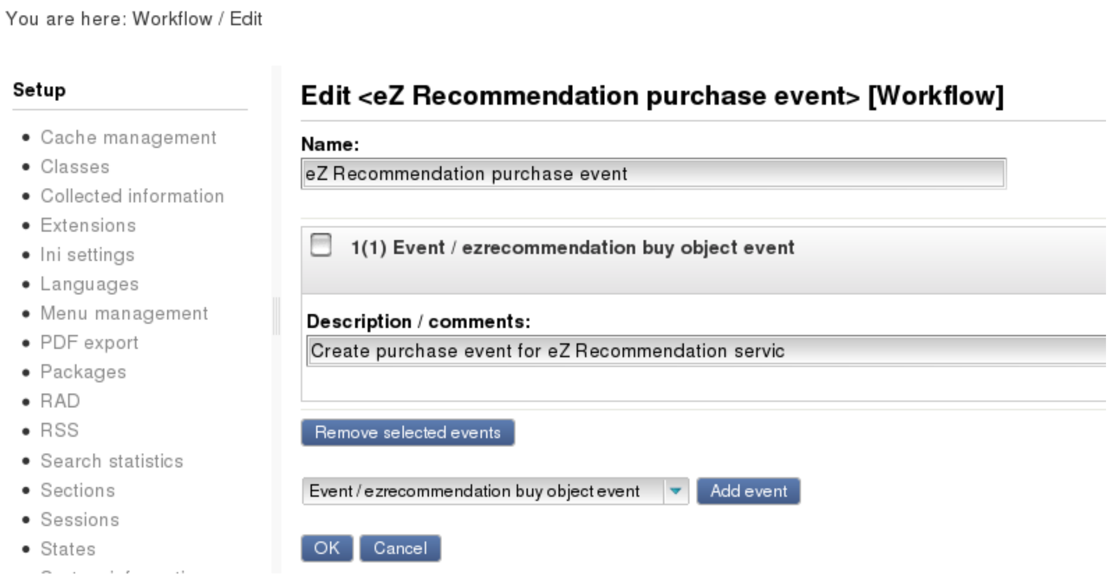
□ **1(1) Event / ezrecommendation delete object event**

Description / comments: Create delete event for eZ Recommendation service

Now that we have created a new workflow we 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.

6.3 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.



Edit <eZ Recommendation purchase event> [Workflow]

Name: eZ Recommendation purchase event

□ 1(1) Event / ezrecommendation buy object event

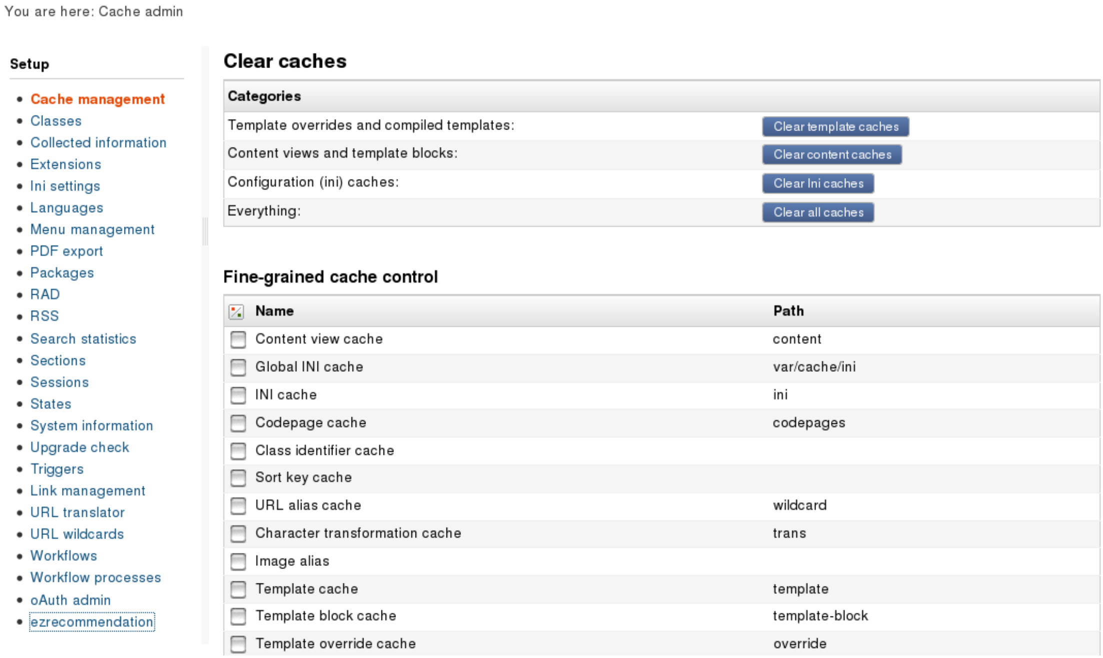
Description / comments: Create purchase event for eZ Recommendation servic

Event / ezrecommendation buy object event

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.

7 Statistics

You can get 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.



You are here: Cache admin

Setup

- Cache management
- Classes
- Collected information
- Extensions
- Ini settings
- Languages
- Menu management
- PDF export
- Packages
- RAD
- RSS
- Search statistics
- Sections
- Sessions
- States
- System information
- Upgrade check
- Triggers
- Link management
- URL translator
- URL wildcards
- Workflows
- Workflow processes
- oAuth admin
- ezrecommendation

Clear caches

Categories	
Template overrides and compiled templates:	Clear template caches
Content views and template blocks:	Clear content caches
Configuration (ini) caches:	Clear Ini caches
Everything:	Clear all caches

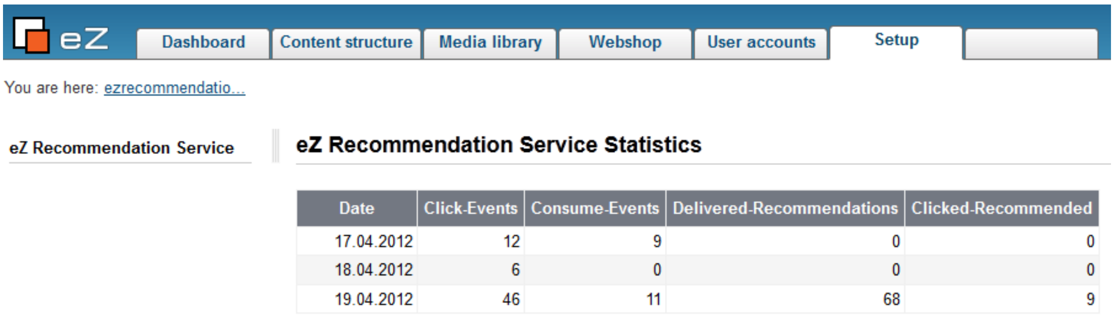
Fine-grained cache control

	Name	Path
□	Content view cache	content
□	Global INI cache	var/cache/ini
□	INI cache	ini
□	Codepage cache	codepages
□	Class identifier cache	
□	Sort key cache	
□	URL alias cache	wildcard
□	Character transformation cache	trans
□	Image alias	
□	Template cache	template
□	Template block cache	template-block
□	Template override cache	override

It provides absolute numbers about

1. Click events that were triggered
2. Purchase events that were triggered (online shop)
3. Delivered recommendations (i.e. 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 if the shop solution is used

The following screenshots are examples that provide statistics for a publisher and a shop solution. Instead of consume events, you have three additional columns in the shop solution named ‘Revenue’, ‘Purchased-Recommended’ and ‘Purchase-Events’.

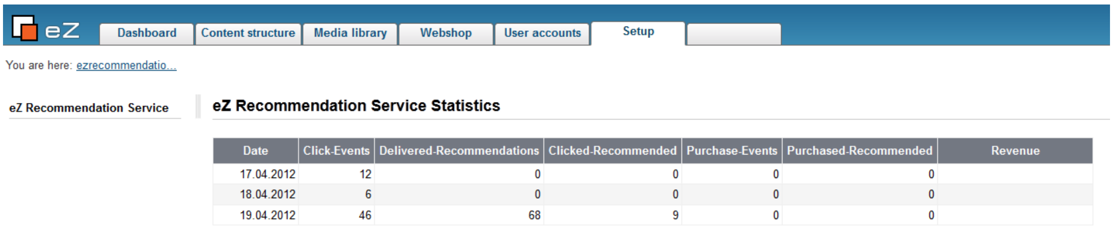


You are here: ezrecommendatio...

eZ Recommendation Service

eZ Recommendation Service Statistics

Date	Click-Events	Consume-Events	Delivered-Recommendations	Clicked-Recommended
17.04.2012	12	9	0	0
18.04.2012	6	0	0	0
19.04.2012	46	11	68	9



You are here: ezrecommendatio...

eZ Recommendation Service

eZ Recommendation Service Statistics

Date	Click-Events	Delivered-Recommendations	Clicked-Recommended	Purchase-Events	Purchased-Recommended	Revenue
17.04.2012	12	0	0	0	0	
18.04.2012	6	0	0	0	0	
19.04.2012	46	68	9	0	0	

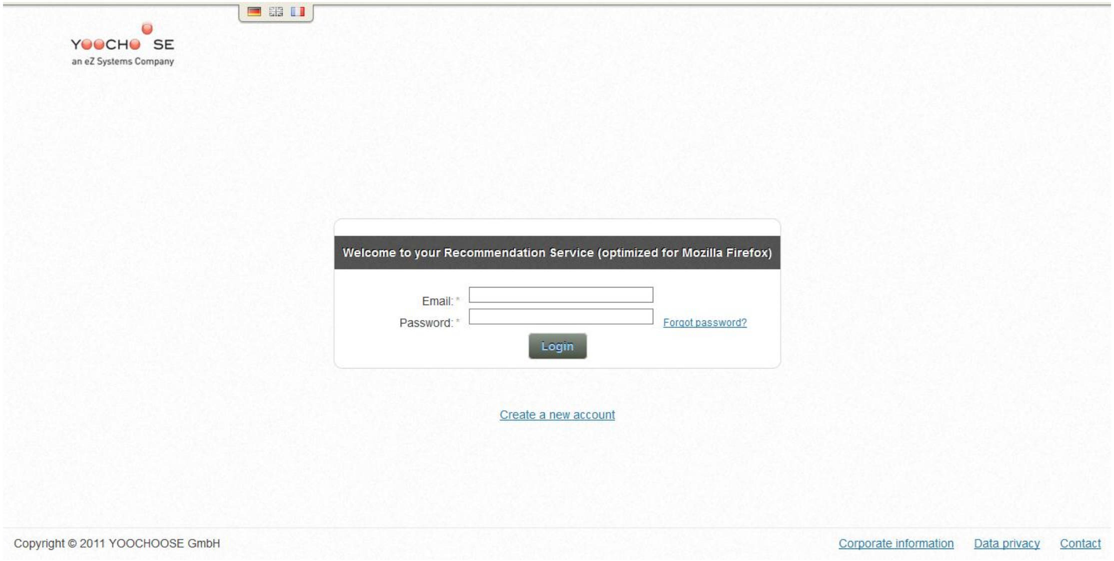
8 Appendixes

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

8.1.1 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:

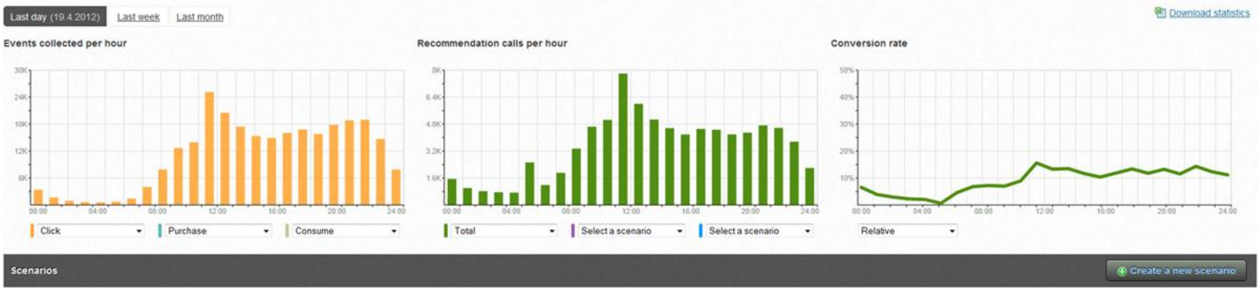


Welcome to your Recommendation Service (optimized for Mozilla Firefox)

Copyright © 2011 YOOCHOOSE GmbH

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*. If not registered, simply add a new account.

You will be redirected to the Welcome page showing available *Recommendation Scenarios* and *Account Data*.



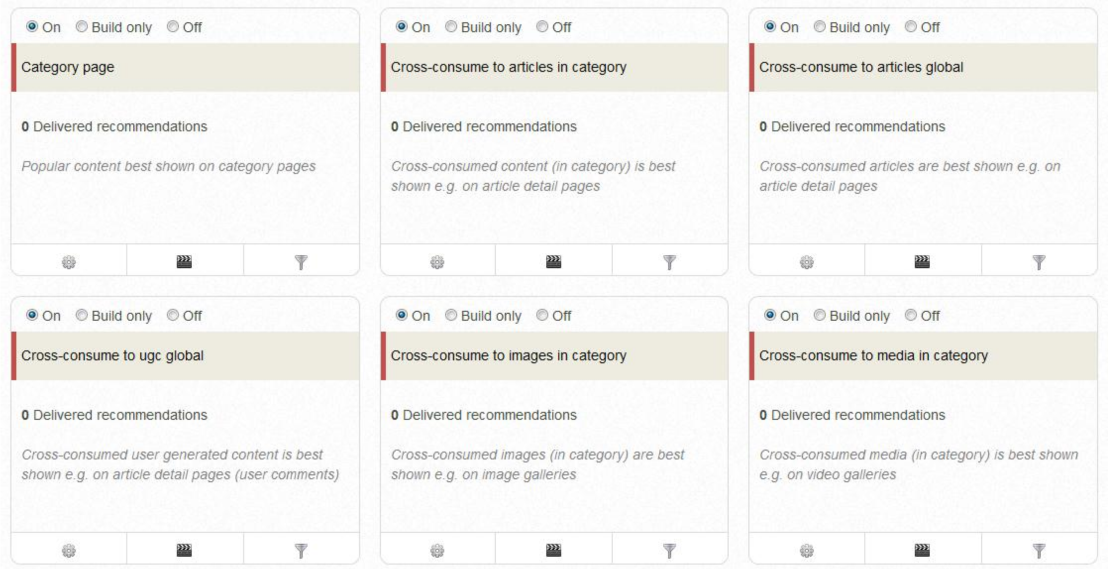
On the upper right part of the screenshot you will find your account detail information:

- 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
```

Below you will find a list of the scenarios which are provided for your recommendation calculation. Most scenarios are self-explanatory e.g.:



● On ○ Build only ○ Off

Category page

0 Delivered recommendations

Popular content best shown on category pages

● On ○ Build only ○ Off

Cross-consume to articles in category

0 Delivered recommendations

Cross-consumed content (in category) is best shown e.g. on article detail pages

● On ○ Build only ○ Off

Cross-consume to articles global

0 Delivered recommendations

Cross-consumed articles are best shown e.g. on article detail pages

● On ○ Build only ○ Off

Cross-consume to ugc global

0 Delivered recommendations

Cross-consumed user generated content is best shown e.g. on article detail pages (user comments)

● On ○ Build only ○ Off

Cross-consume to images in category

0 Delivered recommendations

Cross-consumed images (in category) are best shown e.g. on image galleries

● On ○ Build only ○ Off

Cross-consume to media in category

0 Delivered recommendations

Cross-consumed media (in category) is best shown e.g. on video galleries

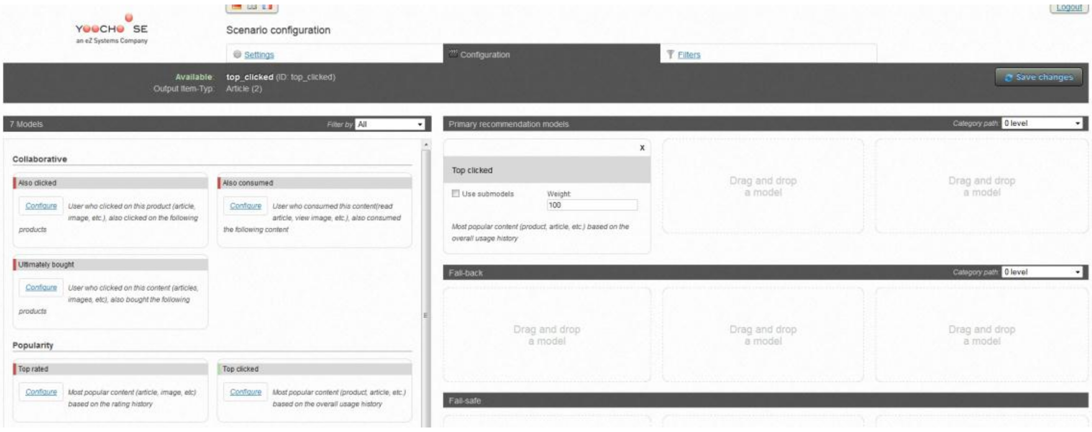
- also_clicked → “Customers who clicked this item, also clicked …”
- also_purchased → “Customers who bought this product, also bought …”
- top_clicked → “Most clicked items” / “Most popular items”
- top_consumed → “Most bought products” / “Most popular products bought”

- ulltimately_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.

8.1.2 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 publisher client with *Basic* license.



Available: top_clicked (ID: top_clicked)
Output Item-Typ: Article (2)

7 Models — Filter by: All

Collaborative

Also clicked — Configure — User who clicked on this product (article, image, etc.), also clicked on the following products

Also consumed — Configure — User who consumed this content(read article, view image, etc.), also consumed the following content

Ultimately bought — Configure — User who clicked on this content (articles, images, etc), also bought the following products

Popularity

Top rated — Configure — Most popular content (article, image, etc) based on the rating history

Top clicked — Configure — Most popular content (product, article, etc.) based on the overall usage history

Primary recommendation models — Category path: 0 level

Top clicked
□ Use submodels — Weight: 100
Most popular content (product, article, etc.) based on the overall usage history

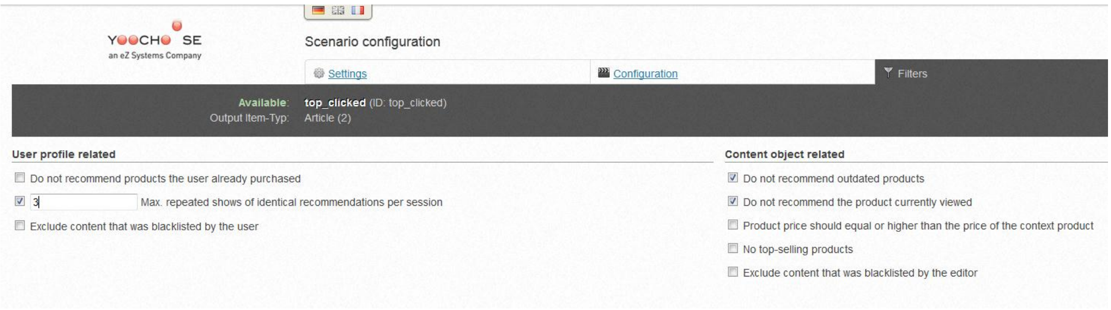
Fall-back — Category path: 0 level

You can configure your scenarios comprising the available models as you like it via drag&drop functionality.

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

8.1.3 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.



Available: top_clicked (ID: top_clicked)
Output Item-Typ: Article (2)

User profile related

□ Do not recommend products the user already purchased
☑ 3 Max. repeated shows of identical recommendations per session
□ Exclude content that was blacklisted by the user

Content object related

☑ Do not recommend outdated products
☑ Do not recommend the product currently viewed
□ Product price should equal or higher than the price of the context product
□ No top-selling products
□ Exclude content that was blacklisted by the editor

8.2 Frequently asked Questions

8.2.1 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.

8.2.2 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.

8.2.3 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.

8.2.4 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.

8.2.5 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.

8.2.6 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://admin.yoochoose.net/<solution>/v3/<CustomerID>/revenue/statistic.xlsx?from_date_time=2012-04-17T00:00:00&to_date_time=2012-04-20T00:00:00&granularity=PT1H

In the above example the granularity is set up to 60 minutes which can be changed to a minimum value of 15 minutes. The format of the request looks like:

https://admin.yoochoose.net/<solution>/v3/<CustomerID>/revenue/statistic.xlsx?from_date_time=[start_time(ISO8601)]&to_date_time=[end_time(ISO8601)]&granularity=PT[Steps in minutes]M

8.2.7 Logging Capabilities

Question: How do I view logging from the ezrecommendations? Where can I check that the communication between my installation and the remote recommendations service is function properly?

Answer:In order to view the activity between your web aite and the recommendations service do as follow:

1. For enabling the general debugging, add the following lines to the file settings/override/site.ini.append.php

```
[DebugSettings]
DebugOutput=enabled
```

2. Under the directory settings/override/ create the file debug.ini.append.php and add the following lines:

```
<?php /*
[DebugSettings]
ConditionDebug=enabled
[GeneralCondition]
extension-ezrecommendation=enabled
[ErrorCondition]
extension-ezrecommendation=enabled
*/
?>
```

Following is an example of information that can be view in the log (under var/log/notice.log) regarding information that has been sent to the recommendations service:

```
[ Dec 21 2012 14:17:25 ] [192.168.56.1] Trying request <extension-
ezrecommendation>:reco.yoochoose.net/news/569/14/also_clicked_article_global.json?itemid=165&numre
cs=3&itemtypeid=2

[ Dec 21 2012 14:17:26 ] [192.168.56.1] Trying HTTP Request <extension-
ezrecommendation>:event.yoochoose.net/news/569/transfer/135609477413953435/14
```

Following is an example of information that can be view in the log (under var/log/debug.log) regarding information that has retrieved from the recommendations service:

```
[ Dec 21 2012 14:03:06 ] [192.168.56.1] Sending HTTP Request to event.yoochoose.net <extension-
ezrecommendation>:GET /news/569/click/14/2/203?categorypath=%2F2%2F201%2F HTTP/1.0

Host: event.yoochoose.net Authorization: Basic NTY5OjEzMTItNzExNi01MzkzLTEwMTA5MTAtNzMxOA==
```

8.3 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).

8.3.1 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´.

8.3.2 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.append.php´ 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/ezrecommendation/design/standard/images/bulkexport.xml), can be accessed from your browser.