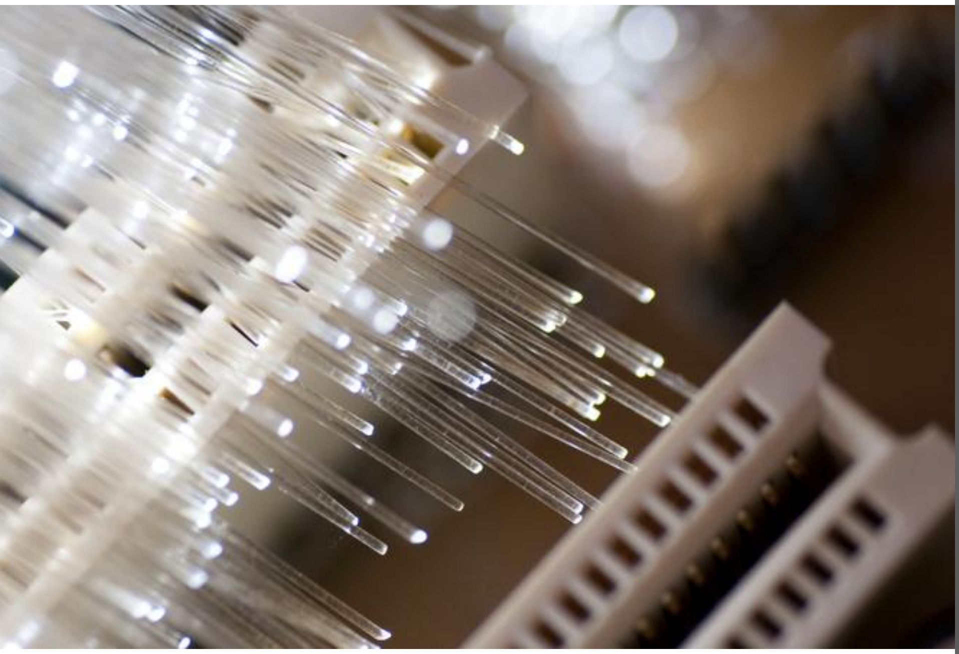
WE RECOMMEND YOOCHOOSE

eZ Recommender Extension Manual



Dr. Uwe Alkemper

19.09.2011

Inhalt

1 About eZ Recommender extension

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

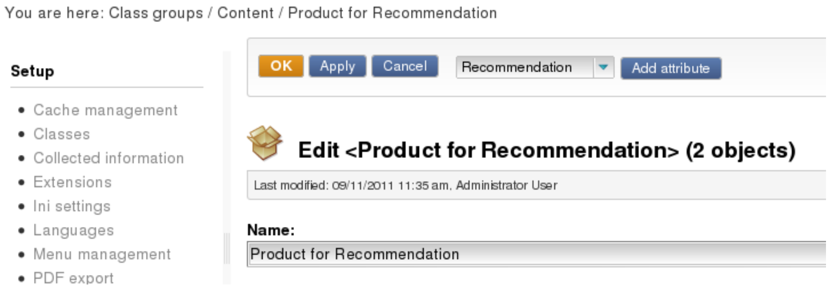
It contains the following:

- Defines a new *Recommendation* datatype with a *recommend* flag
- Provide 4 new classes *Article for Recommendation, Product for Recommendation, Image for Recommendation* and *Folder for Recommendation*
- Creates a workflow *export content to recommendation*
- By default this workflow is triggered *after publishing* of objects marked as *recommend*
- Creates *tracking events* from user activities on objects that are marked as *recommend*
- Creates workflow for *purchase* events when products are bought in the webshop
- Creates automatic *delete* event workflow when a node is deleted
- Creates a *recommendation* entry in the support panel to show *statistics*
- Accesses the YOOCHOOSE cloud-service APIs to retrieve recommendations on demand from the recommendation service
- Provides a script to trigger a full export of all existing content marked as *recommend*
- Supports *publisher* as well as *shop* recommendations

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 recommender extension this datatype *Recommendation* is available as you can see when you edit an existing/new class in the backend of your eZ publish installation



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

Setup

- Cache management
- Classes
- Collected information
- Extensions
- Ini settings
- Languages
- Menu management
- PDF export

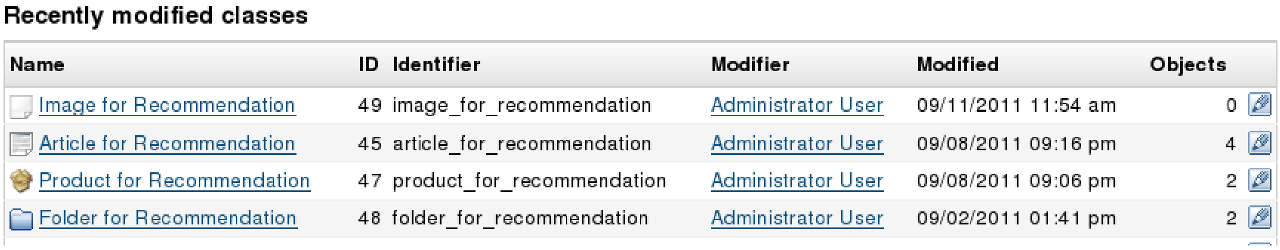
Edit <Product for Recommendation> (2 objects)

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

Name: Product for Recommendation

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 provide you with 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.

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

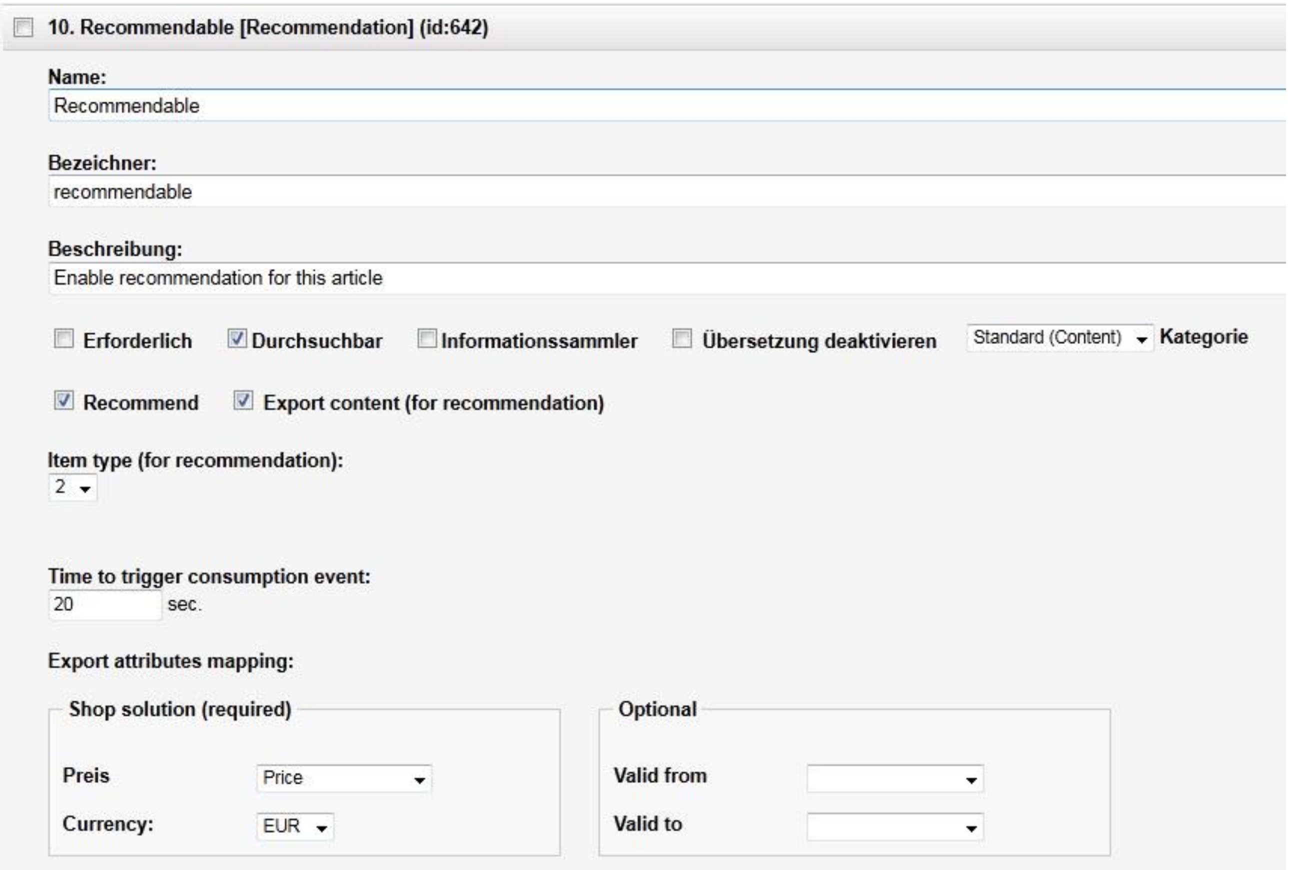


Recently modified classes

Name	ID	Identifier	Modifier	Modified	Objects
Image for Recommendation	49	image_for_recommendation	Administrator User	09/11/2011 11:54 am	0
Article for Recommendation	45	article_for_recommendation	Administrator User	09/08/2011 09:16 pm	4
Product for Recommendation	47	product_for_recommendation	Administrator User	09/08/2011 09:06 pm	2
Folder for Recommendation	48	folder_for_recommendation	Administrator User	09/02/2011 01:41 pm	2

Now edit *article for recommendation* 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 belongs to the shop solution.



□ 10. Recommendable [Recommendation] (id:642)

Name:
Recommendable

Bezeichner:
recommendable

Beschreibung:
Enable recommendation for this article

□ Erforderlich ☑ Durchsuchbar □ Informationssammler □ Übersetzung deaktivieren Standard (Content) Kategorie

☑ Recommend ☑ Export content (for recommendation)

Item type (for recommendation):
2

Time to trigger consumption event:
20 sec.

Export attributes mapping:

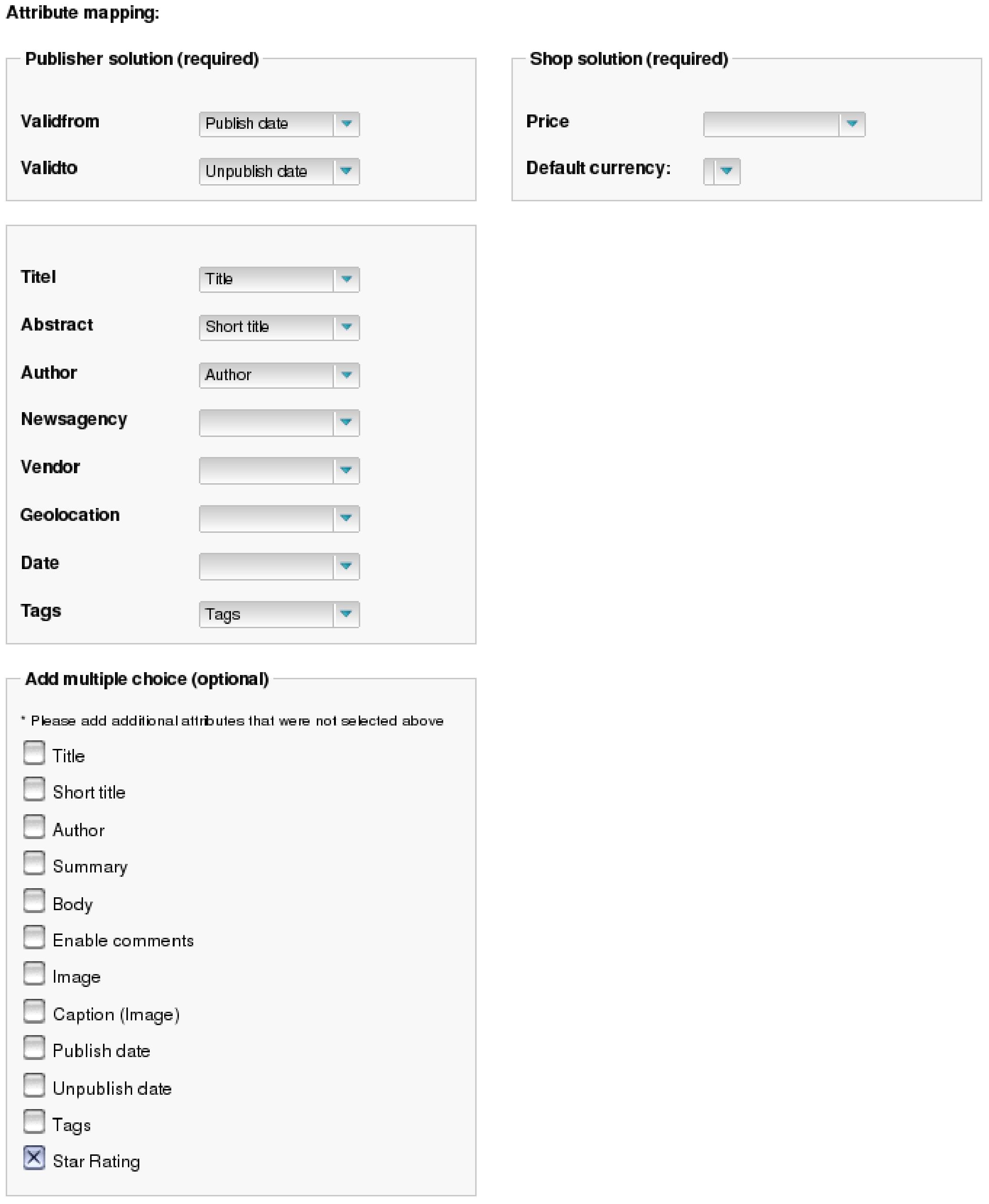
Shop solution (required)

Preis: Price
Currency: EUR

Optional

Valid from:
Valid to:

There are several entries defining how the extension operates, tracks user activities and delivers content information to the eZ recommendation engine. The lower part of the Recommendation entry defines which information about a content object is transmitted to the eZ recommender engine and how the data is mapped between each recommendable class and the datatypes that can be interpreted by the eZ recommender engine.



Attribute mapping:

Publisher solution (required)

Validfrom	Publish date
Validto	Unpublish date

Shop solution (required)

Price	
Default currency:	

Titel	Title
Abstract	Short title
Author	Author
Newsagency	
Vendor	
Geolocation	
Date	
Tags	Tags

Add multiple choice (optional)

* Please add additional attributes that were not selected above

- ☐ Title
- ☐ Short title
- ☐ Author
- ☐ Summary
- ☐ Body
- ☐ Enable comments
- ☐ Image
- ☐ Caption (Image)
- ☐ Publish date
- ☐ Unpublish date
- ☐ Tags
- ☒ Star Rating

2.2 *Recommend* check box

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. 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 recommendation into an existing class of yours e.g. *article (main-page).*

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.

2.3 Export content (for recommendation) check box

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.

2.4 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 class by the recommendation service.

When the extension tracks the user activities it will tell the recommender engine 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.

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

Item type (for recommendation)

Number	Comment
1	Product
2	Article
3	Image
4	Video
5	Event
6	Location
7	Gallery
8	<custom>
9	<custom>
10	<custom>

2.5 Time to trigger consumption event value

This value is only relevant when you want to track if a user reads an article, watches a video or studies a picture. The 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. 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 in a class.

2.6 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.

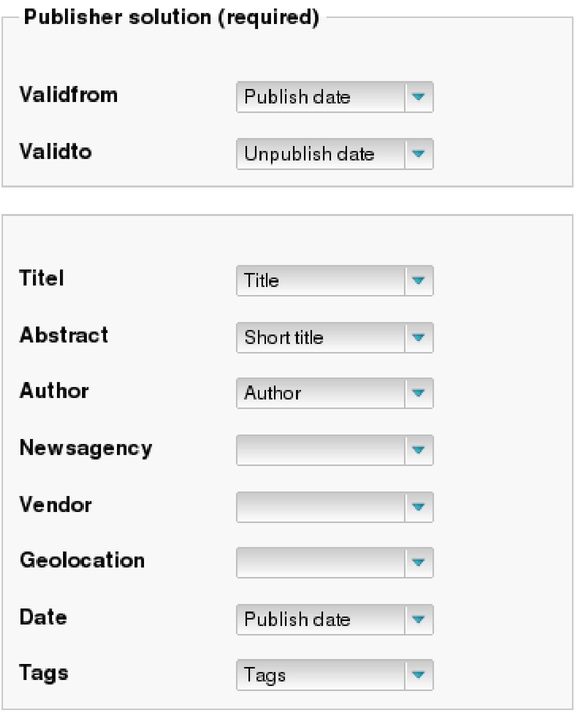
Each time you publish some content and the content is marked as *Export content (for recommendation)* the extension sends some parts of this information to the recommender engine in the cloud. With the content export mapping you define which information is sent to the recommender engine and used for future model calculation.

This is how it works:

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 while *Export content (for recommendation)* is checked to create recommendation only. But providing this information for more attributes 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*).



Publisher solution (required)

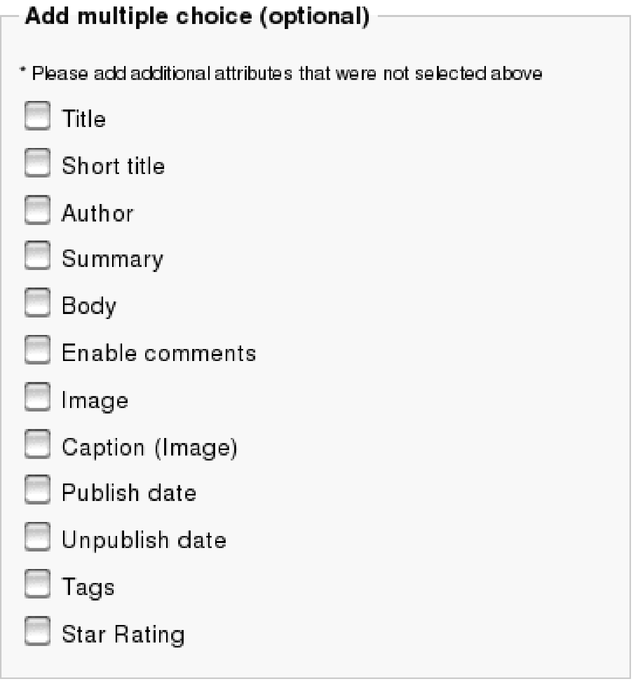
Validfrom	Publish date
Validto	Unpublish date

Titel	Title
Abstract	Short title
Author	Author
Newsagency	
Vendor	
Geolocation	
Date	Publish date
Tags	Tags

On the right-hand side you can choose in a selection box which of your class attributes you want to map to the 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 recommender engine you can make use of the additional information for filter purposes. The Recommendation datatype therefore allows defining *Additional attributes for content export.*



Add multiple choice (optional)

* Please add additional attributes that were not selected above

□ Title
□ Short title
□ Author
□ Summary
□ Body
□ Enable comments
□ Image
□ Caption (Image)
□ Publish date
□ Unpublish date
□ Tags
□ Star Rating

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

3 Implement tracking

3.1 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 eZ recommender tracking in pagelayout.tpl *}

{include uri='design:content/ezyoochoose_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/ebl/XXXXX/click/
131574345055084112/2/110?categorypath=%2F1%2F2%2F107%2F110
```

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

3.2 Create other common events (“Owns”, “Blacklist”, “Recommend”, “Click Recommended”)

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 a product (/full/product.tpl) into our own design and created an override for the class *product for recommendation*. The we added the following code at the beginning of the .tpl file:

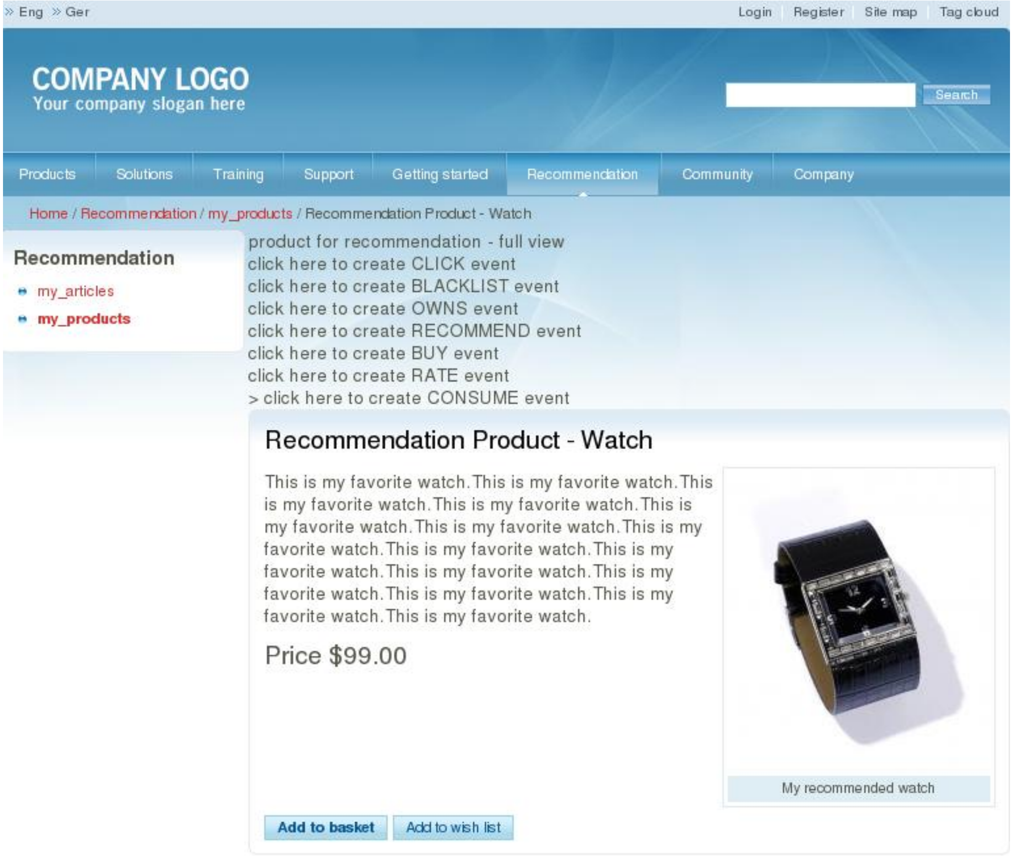
```
{* Product for Recommendation - Full view *}

{* Start - block for eZ recommender testing purposes *}

product for recommendation - full view

{* this creates different events on click *}

<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, 'recommend')}>
click here to create RECOMMEND event </div>

<div {generate_buy_event($node, 1, 11, 'EUR')}>
click here to create BUY event </div>

<div {generate_rate_event($node, 56)}>
click here to create RATE event </div>

<div {generate_consume_event($node)}>
click here to create CONSUME event </div>

{* End - block for eZ recommender testing purposes *}
```

Comment: although not shown in the example you can also use:

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

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



product for recommendation - full view
click here to create CLICK event
click here to create BLACKLIST event
click here to create OWNS event
click here to create RECOMMEND event
click here to create BUY event
click here to create RATE event
> click here to create CONSUME event

Recommendation Product - Watch

This is my favorite watch. This is my favorite watch. This is my favorite watch. This is my favorite watch. This is my favorite watch. This is my favorite watch. This is my favorite watch. This is my favorite watch. This is my favorite watch. This is my favorite watch. This is my favorite watch. This is my favorite watch. This is my favorite watch. This is my favorite watch.

Price $99.00

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

```
debug.log:

[ Sep 11 2011 15:07:05 ] event.yoochoose.net/ebl/XXXXX/blacklist
/131574611641487835/1/239?categorypath=%2F1%2F2%2F228%2F241%2F
```

```
...

[ Sep 11 2011 15:07:09 ] event.yoochoose.net/ebl/XXXXX/owns
/1315746116414878351/1/239?categorypath=%2F1%2F2%2F228%2F241%2F

...

[ Sep 11 2011 15:07:13 ] event.yoochoose.net/ebl/XXXXX/recommend
/1315746116414878351/1/239?categorypath=%2F1%2F2%2F228%2F241%2F
```

3.3 “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 that 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 recommender engine with the next click.

3.4 “Rate”-event

The RATE and the PURCHASE-events need additional parameters to work.

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

```
<div {generate_rate_event($node, 56)}>
click here to create RATE event </div>
```

3.5 “Purchase”-event

The PURCHASE event is triggered after a order has been checked out. Therefore you need to configure the buy event workflow.

Go to the Setup-tab and then to Workflows. Choose the Standard workflow group or create your own one. In the group you have to create a new workflow and give it a name. Add the "yoochoose buy object event" to your workflow.

After this you need to configure the trigger for your workflow. Go to the Setup tab and then to Triggers. Add the your workflow to the "shop checkout before" trigger and apply the changes.

From now on every time a order is checked out ezyoochoose will track every product purchased in this order.

PURCHASE requires 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>
```

3.6 “Delete”-event

The content of an object removed in the eZ publish backend can automatically be sent to the eZ recommender engine. If this is done the content will not be recommended in the future since it is marked as deleted.

Therefore you have to define a workflow and a trigger which causes the workflow.

Go to the Setup tab and then to Workflows. Choose the Standard workflow group or create your own one. In the group you have to create a new workflow and give it a name. Add the "yoochoose delete object event" to your workflow.

After this you need to configure the trigger for your workflow. Go to the Setup tab and then to Triggers. Add your workflow to the "content delete before" trigger and apply the changes.

3.7 “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. Our freqency filter allows not to show recommendations more than 3 times.

How often a recommendation was already rendered can be counted from so-called RENDERED-events. When you enable that events are created for rendered recommendations they will be sent to the recommender engine with each rendering.

No additional code must be integrated. It must be only enabled in the get recommendation call described below.

3.8 “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.

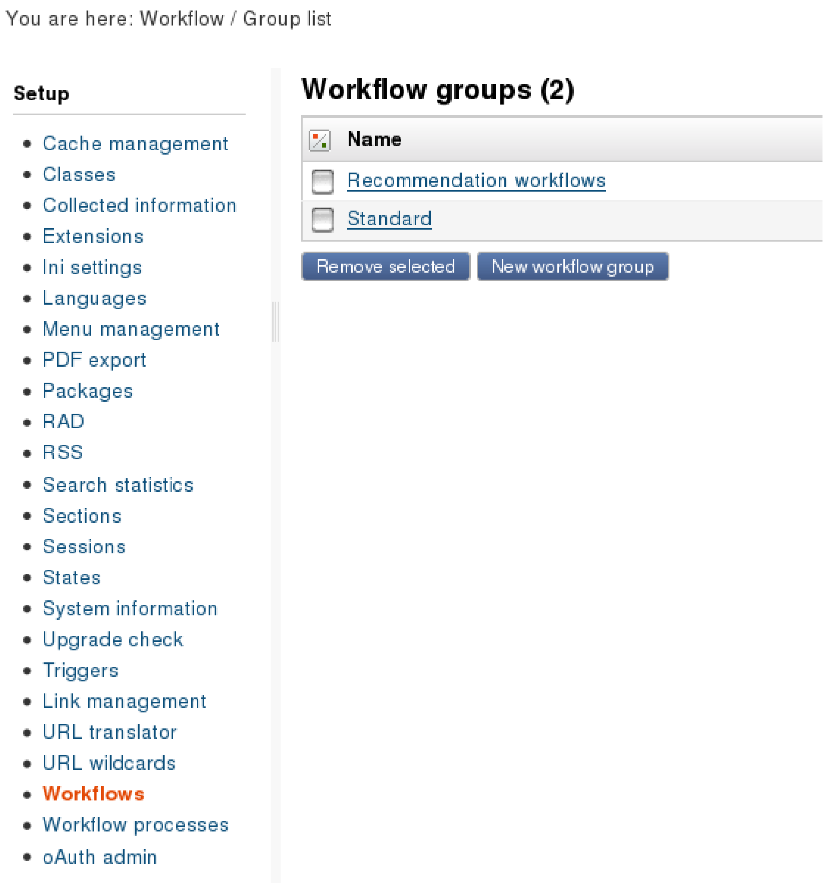
4 Content export

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

To get into effect we must establish an incremental content export that is triggered with each publishing event and we must initiate a one-time export of the whole content from a class when the export conditions of the 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).

4.1 Implement workflow to export content after publishing

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

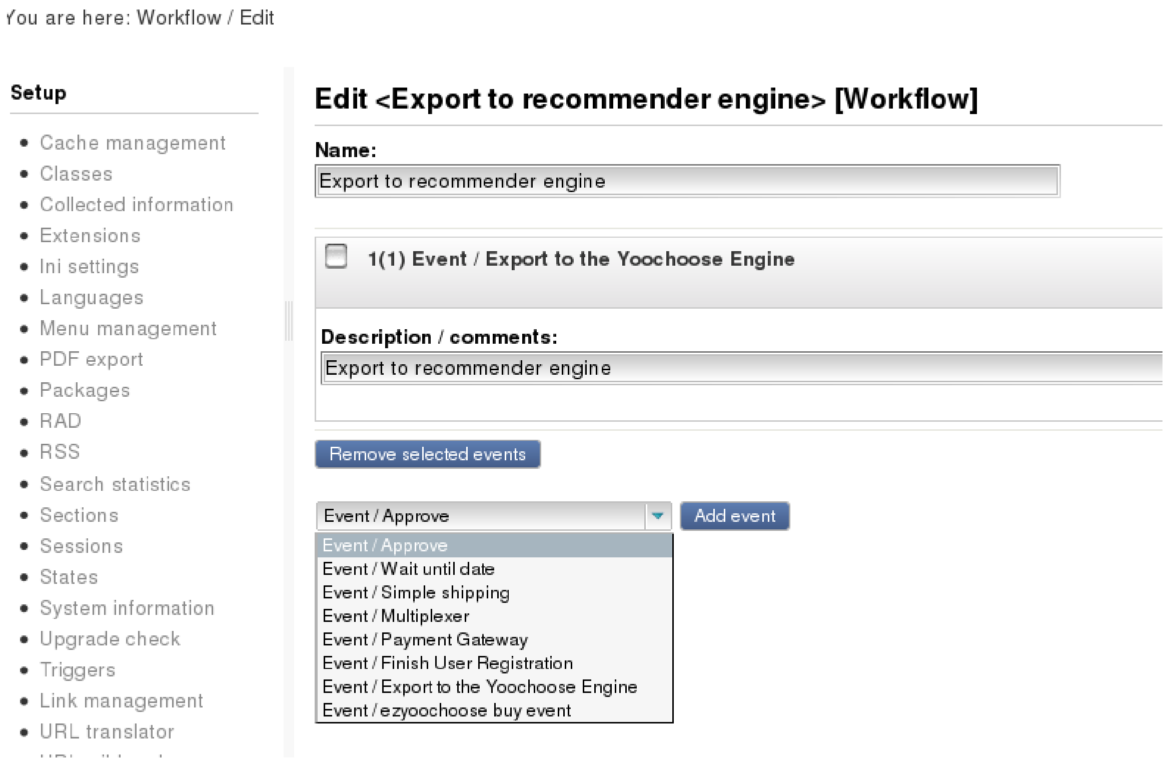


You are here: Workflow / Group list

Workflow groups (2)

	Name
□	Recommendation workflows
□	Standard

Open the workflow group and create a new workflow. In our example we name it „Export to recommender engine“ and select „Event/Export to Recommender Engine“ 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
- Upgrade check
- Triggers
- Link management
- URL translator

Edit <Export to recommender engine> [Workflow]

Name: Export to recommender engine

□ 1(1) Event / Export to the Yoochoose Engine

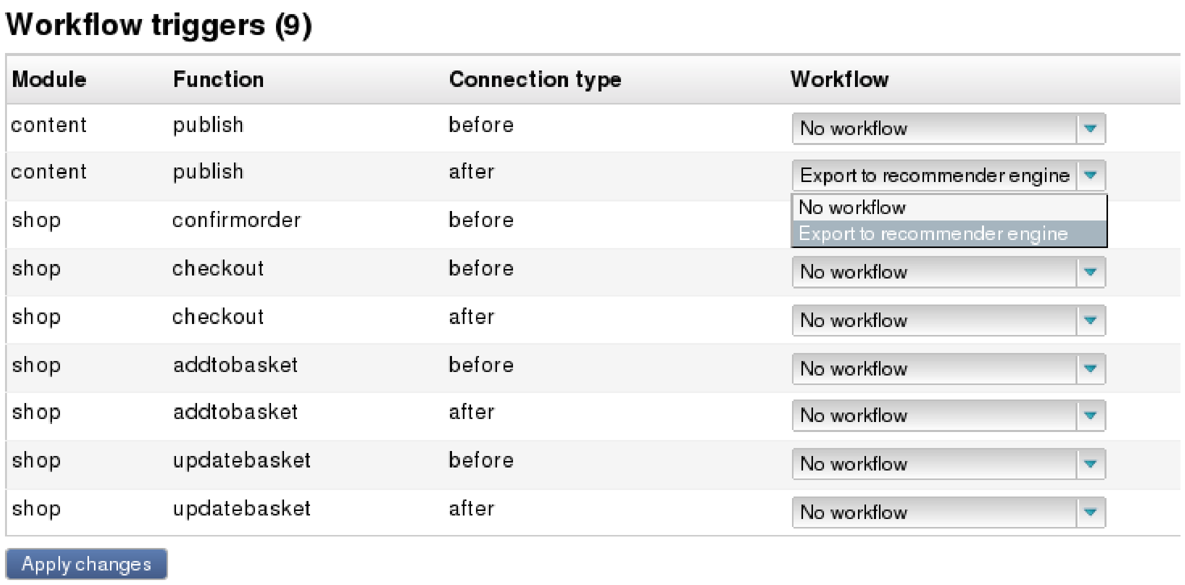
Description / comments: Export to recommender engine

Remove selected events

Event / Approve ▾ Add event

- Event / Approve
- Event / Wait until date
- Event / Simple shipping
- Event / Multiplexer
- Event / Payment Gateway
- Event / Finish User Registration
- Event / Export to the Yoochoose Engine
- Event / ezyoochoose buy event

Now that we have created a new we can use in the Triggers and use it each time when content is exported. To do so select „Event / Export to recommender engine“ in the row content – publish – after and apply the changes.



Workflow triggers (9)

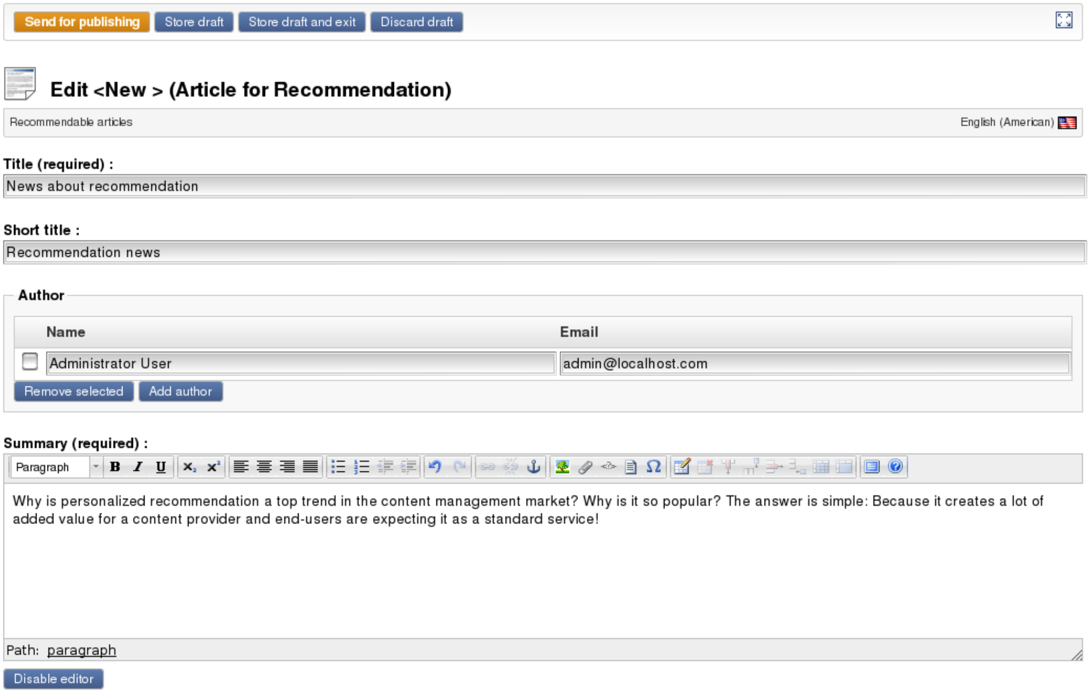
Module	Function	Connection type	Workflow
content	publish	before	No workflow
content	publish	after	Export to recommender engine No workflow Export to recommender engine
shop	confirmorder	before	
shop	checkout	before	No workflow
shop	checkout	after	No workflow
shop	addtobasket	before	No workflow
shop	addtobasket	after	No workflow
shop	updatebasket	before	No workflow
shop	updatebasket	after	No workflow

Apply changes

4.2 Create content for recommendation

Let us now create new content and see whether content export operates as intended.

In our example we create content of the type *article for recommendatiion.*



Edit <New > (Article for Recommendation)

Recommendable articles — English (American)

Title (required) :
News about recommendation

Short title :
Recommendation news

Author

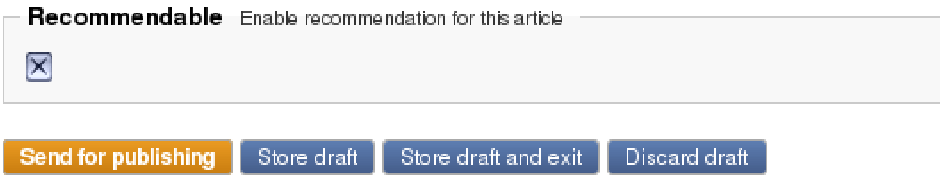
Name	Email
□ Administrator User	admin@localhost.com

Summary (required) :

Why is personalized recommendation a top trend in the content management market? Why is it so popular? The answer is simple: Because it creates a lot of added value for a content provider and end-users are expecting it as a standard service!

Path: paragraph

Fill all required fields of the new article for recommendation. In the bottom of the new content object you can change the *Recommendable* settings of the class definition:



Recommendable Enable recommendation for this article

☒

Finallly press *Send for publishing.*

For a test to verify what has been sent to the recommender engine check again the latest entries in <eZ publish root directory>/var/log/debug.log. You will find an entry like:

```
debug.log:

[ Sep 11 2011 16:49:03 ] import.yoochoose.net/news/XXXXXX
/item<?xml version="1.0" encoding="utf-8"?>

<items version="1"><item id="246" type="2"><validfrom>2011-09-
11T12:00:00</validfrom><validto>2011-09-20T12:00:00</validto>
<categorypaths><categorypath>/2/228/</categorypath>
</categorypaths><content><content-data key="title">
<![CDATA[News about recommendation]]></content-data>
<content-data key="abstract"><![CDATA[Recommendation news]]>
</content-data><content-data key="tags"><![CDATA[recommender,
algorithm,yoochoose,technology,news]]></content-data>
</content><attributes><attribute key="date" value="2011-09-11
T12:00:00"/></attributes></item></items>
```

4.3 Initial content export

When you have a eZ publish installation with lots of content prior to the installation of the eZ recommender extension you need to export the content you want to be recommended. This is done once after you add the *Recommendation* datatype to an existing class. New content is automatically exported as described above.

Before you can start the initial export go to the ezyoochoose.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. http://example.com/wwwdir/).

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 publish date and the mapping 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/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 recommender engine. 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.

5 Retrieving recommendations

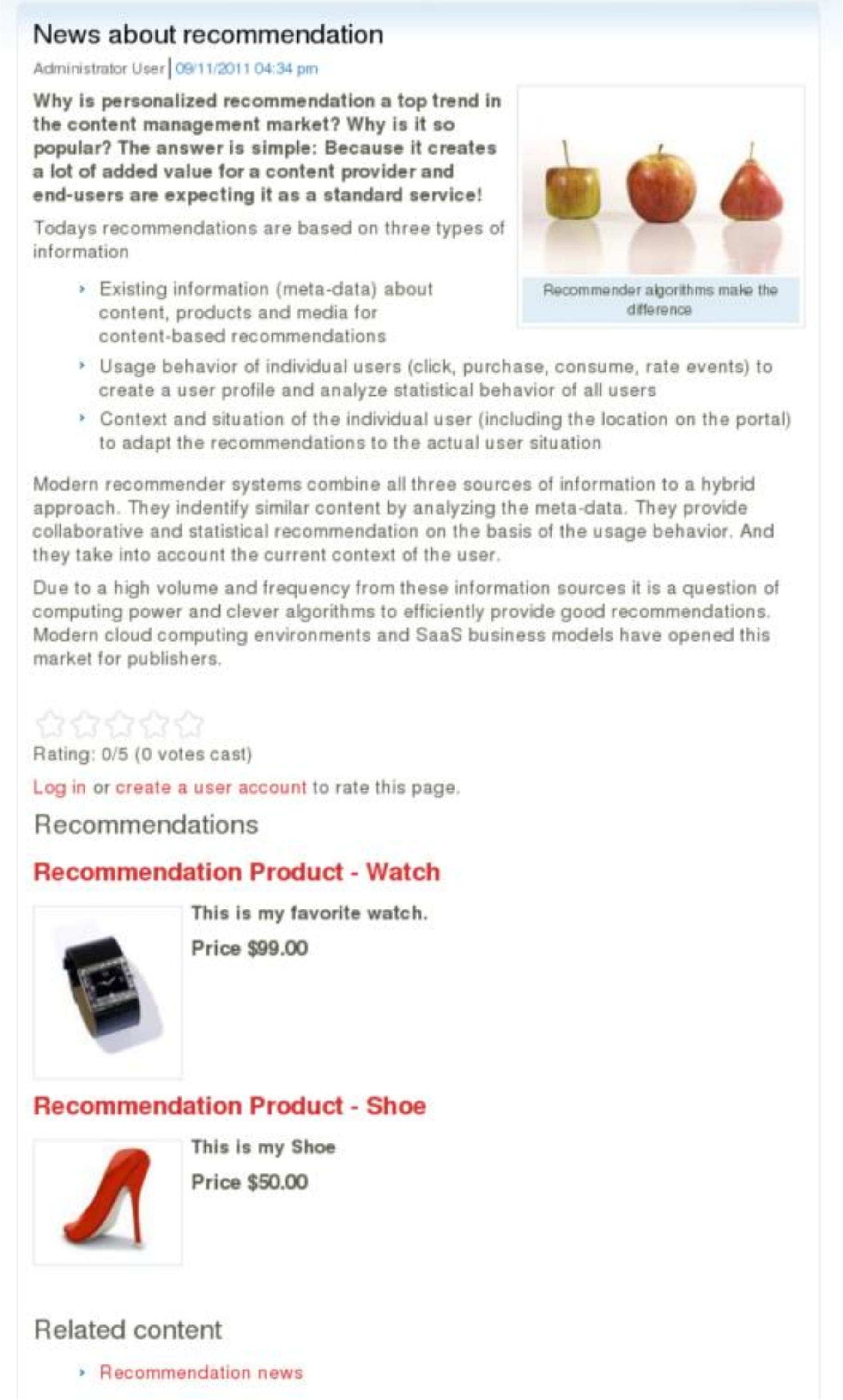
To get recommendations you need to include the following code wherever you want to see the recommendations

```
{include uri='design:content/recommendations.tpl'  node=$node sce-
nario='top_clicked' limit=3 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.

```
Line/ycarticle.tpl

...

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

   {attribute_view_gui attribute=$node.data_map.star_rating}

</div>

{* Start - block for eZ recommender testing purposes *}

<div class="attribute-recommendations">

<h1>{"Recommendations"}</h1>

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

</div>

{* End - block for eZ recommender testing purposes *}

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

...
```

After reloading of the page we see the following:



News about recommendation

Administrator User | 09/11/2011 04:34 pm

Why is personalized recommendation a top trend in the content management market? Why is it so popular? The answer is simple: Because it creates a lot of added value for a content provider and end-users are expecting it as a standard service!

Recommender algorithms make the difference

Todays recommendations are based on three types of information

- Existing information (meta-data) about content, products and media for content-based recommendations
- Usage behavior of individual users (click, purchase, consume, rate events) to create a user profile and analyze statistical behavior of all users
- Context and situation of the individual user (including the location on the portal) to adapt the recommendations to the actual user situation

Modern recommender systems combine all three sources of information to a hybrid approach. They indentify similar content by analyzing the meta-data. They provide collaborative and statistical recommendation on the basis of the usage behavior. And they take into account the current context of the user.

Due to a high volume and frequency from these information sources it is a question of computing power and clever algorithms to efficiently provide good recommendations. Modern cloud computing environments and SaaS business models have opened this market for publishers.

☆☆☆☆☆

Rating: 0/5 (0 votes cast)

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

Recommendations

Recommendation Product - Watch

This is my favorite watch.

Price $99.00

Recommendation Product - Shoe

This is my Shoe

Price $50.00

Related content

- Recommendation news

The top of the article shows the standard full view followed by star rating, recommendations from the eZ recommender engine.

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:

```
[ Sep 17 2011 16:20:24 ] [ezyoochoose] Received recommendations '

{"recommendationResponseList":[{"reason":"POPULARITY_LONG_1 (context: ITEM(s))","itemType":1,"itemId":239,"relevance":9},{"reason":"POPULARITY_LONG_1 (context: ITEM(s))","itemType":1,"itemId":234,"relevance":2},{"reason":"POPULARITY_LONG_1 (context: ITEM(s))","itemType":1,"itemId":233,"relevance":2}]}'

...
```

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

```
[ Sep 17 2011 16:20:25 ] [ezyoochoose] Trying request event.yoochoose.net/ebl/XXXXXX/rendered/1316263418734579 23/1/239,234

[ Sep 17 2011 16:20:25 ] [ezyoochoose] Sending request event.yoochoose.net/ebl/XXXXXX/rendered/131626341873457923/1/239,234

...
```

5.1 The get recommendation call

The get recommendation request has been triggered in the above include from recommendations.tpl. The 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 (config.yoochoose.net). A detailed description of the portal and how you find out all available

scenarios is described in the next chapter. 'top_clicked' represents the most popular clicked (recommendable) content. For a complete default list of scenarios see below.

`limit=3`

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

`track_rendered_items=true()`

This parameter defines what happens when one or more of the recommendations has been rendered. If defined as true() an additional rendered event is sent to the recommender engine. 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 recommender. The two 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 fills the statistics information about usage and acceptance of recommendations.

5.2 Rendered vs. requested recommendations

Why do we see only two recommendations in our example although three were requested?

There are two possibilities why this typiclally happens:

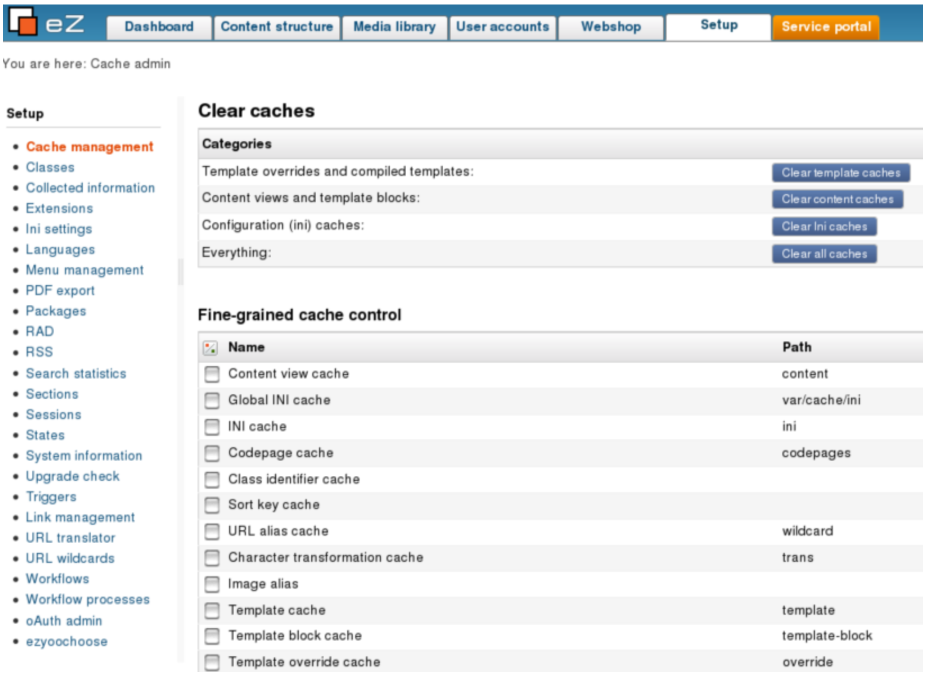
1) The recommender engine cannot provide enough good recommendations for a given scenario.

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

In our example we saw in the debug log that the recommender engine returned three node IDs (233, 234, 239). The solution in this case is therefore 2). The third node has been deleted without noticing the recommender engine.

6 Statistics

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



Clear caches

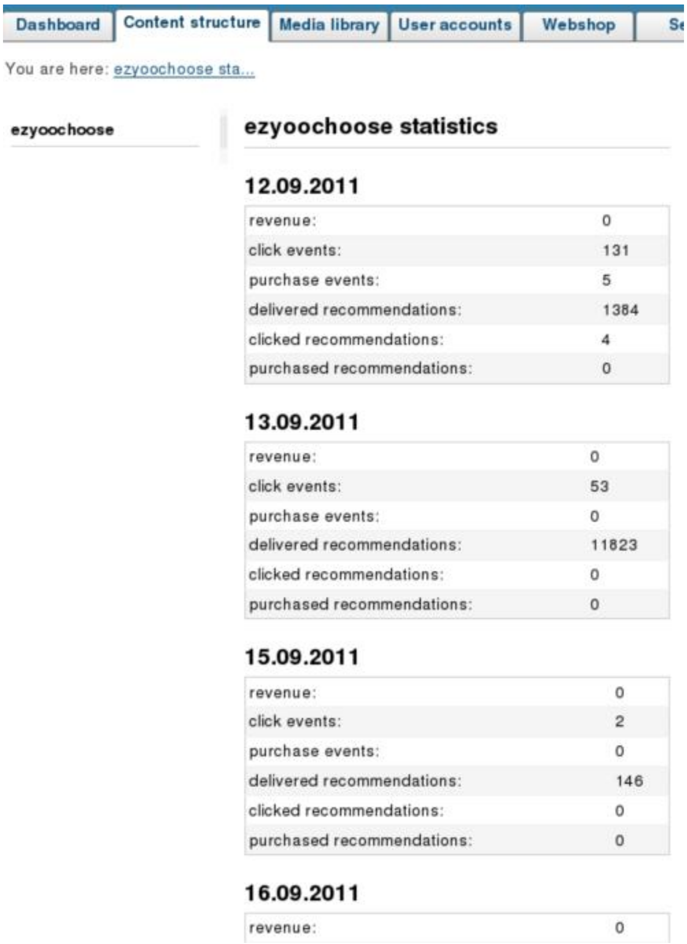
Categories	
Template overrides and compiled templates:	Clear template caches
Content views and template blocks:	Clear content caches
Configuration (ini) caches:	Clear Ini caches
Everything:	Clear all caches

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 (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



You are here: ezyoochoose sta...

ezyoochoose

ezyoochoose statistics

12.09.2011

revenue:	0
click events:	131
purchase events:	5
delivered recommendations:	1384
clicked recommendations:	4
purchased recommendations:	0

13.09.2011

revenue:	0
click events:	53
purchase events:	0
delivered recommendations:	11823
clicked recommendations:	0
purchased recommendations:	0

15.09.2011

revenue:	0
click events:	2
purchase events:	0
delivered recommendations:	146
clicked recommendations:	0
purchased recommendations:	0

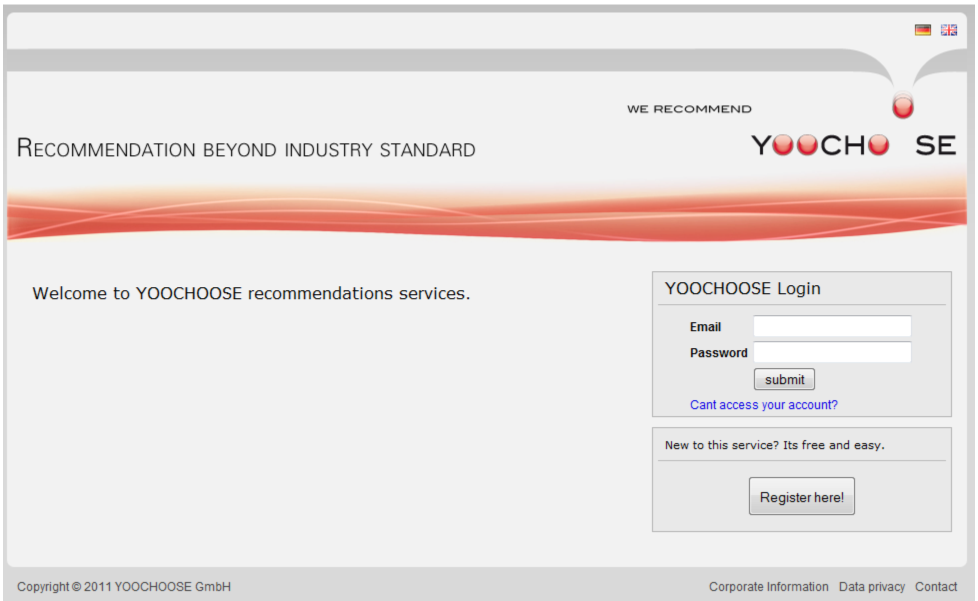
16.09.2011

revenue:	0

7 Configuration Portal (config.yoochoose.net)

7.1 Manage account details

To manage your account details for the eZ recommender engine visit the URL config.yoochoose.net. You will find the following screen:

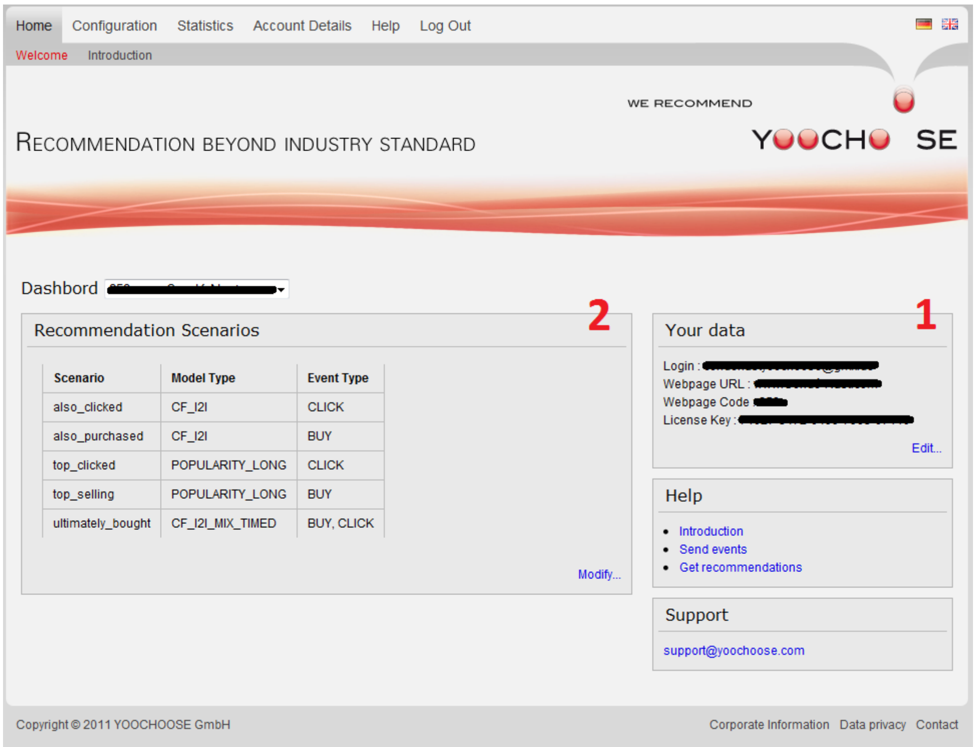


Welcome to YOOCHOOSE recommendations services.

New to this service? Its free and easy.

Copyright © 2011 YOOCHOOSE GmbH

To get access to the configuration Portal fill out the required fields “Email” and “Password” with the data you achieved through the registration process and click “submit”. You will be redirected to the Account-Detail-Page.



Dashbord [redacted]

Scenario	Model Type	Event Type
also_clicked	CF_I2I	CLICK
also_purchased	CF_I2I	BUY
top_clicked	POPULARITY_LONG	CLICK
top_selling	POPULARITY_LONG	BUY
ultimately_bought	CF_I2I_MIX_TIMED	BUY, CLICK

Login : [redacted]
Webpage URL : [redacted]
Webpage Code [redacted]
License Key : [redacted]

support@yoochoose.com

Copyright © 2011 YOOCHOOSE GmbH

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

Login: Your Email address you used for registration, this is needed if you want to log in to our configuration portal.

Webpage URL: Your website the eZ recommender engine will be connected to.

Webpage Code: Your personal authentication number (customer ID)

License Key: Your personal authentication code (license-key)

In order to successfully authenticate your eZ recommender engine, implemented in your eZPublish website, you have to edit the *"ezyoochoose.ini"* file located in the folder */eZPublish/extension/ezyoochoose/settings*

"# Insert here your Yoochoose customerid 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.

Scenario	**Description/Purpose**
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 clicked"*
top_selling	*"Most bought products" / "Most popular products bought"*
umltimatly_bought	*"Customers who bought this product, ultimately bought the following products"*

8 Installation notes

8.1 Requirements

eZ Publish 4.5 or later version.

8.2 Installation

1) Copy ezyoochoose into 'extension' folder or install the ezyoochoose package (Setup->Packages->Import new package).

2) Activate ezyoochoose

   In administrator interface, click 'setup' tab->'extensions' menu,
   select 'ezyoochoose', click button 'Apply Changes'.

   or in settings/override/site.ini.append.php, add activation configuration
   under "ExtensionSettings" section:

   ```
   [ExtensionSettings]
   ActiveExtensions[]=ezyoochoose
   ```

3) Regenerate autoloads

   In administrator user interface, click 'setup' tab->'extensions' menu,
   select 'ezyoochoose', click button 'Regenerate autoload arrays for extensions',

   or in eZ Publish installation folder, run "php bin/php/ezpgenerateautoloads.php -e"

4) Grant access for the ezyoochoose module

   In administrator user interface, click 'User accounts' -> 'Roles and policies'
   then 'Anonymous'. Click on the Role and edit the Policies.

   Add new Policy. Choose the module 'ezyoochoose' and the Function 'request'
   and then grant access to this function.

   Do the same for the Role 'Members' if you have a login area an your site.

5) Clear cache

   Clear INI and template caches. (from admin 'Setup' tab or commandline)

8.3 Settings in ezyoochoose.ini

```
[SolutionSettings]
```

The user has to specify here the yoochoose solution he wants to use (shop or publisher).

```
[ShopPriceCurrency]
```

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

[ClientIdSettings]

These are the authentication string the user gets for his ez instance from yoochoose.

[RequestSettings]

If enabled, the answer from the eZ recommender engine will be logged in debug.log.

The following settings should only be modified in consultation with yoochoose.

[URLSettings]

Settings for the YOOCHOOSE Server URLs.

[ParameterMapSettings]

Maps the ez attribute names to the ezyoochoose parameters.

[SolutionMapSettings]

Maps the type of the site to the ezyoochoose productid.

[ExtensionSettings]

Defines the response form. Json is supported until now.