

Recommendation Bundle

Overview

Recommendation Bundle integrates the YOOCHOOSE recommendation engine into eZ.

The Recommendation Bundle extends the functionality of eZ with a recommendation engine, powered by YOOCHOOSE. It allows you to track the way visitors use your website and suggests recommended content to them based on their behavior.

See <https://yoochoose.com/Personalization-Solution/Documentation> to learn how the recommendation engine works from the YOOCHOOSE side.

Support

Stable versions of this bundle is officially supported by eZ as of version 1.0.0, for users with a eZ Enterprise and Yoochoose subscription.

In this topic

- Installing the Recommendation Bundle
 - Requirements
 - Installation
 - Configuration
 - A
- Using the Recommendation Bundle
 - Initial Setup
 - Indexing
 - Tracking
 - Displaying
 - Troubleshooting
- Other resources

Installing the Recommendation Bundle

The Recommendation Bundle is installed in a similar way as any other Symfony bundle.

Requirements

- PHP 5.4.4 or higher PHP 5.x version
- Symfony 2.6 or higher Symfony 2.x version
- eZ Publish 5.4.1 or eZ Platform/Studio 2015.01 or above, with the REST API configured to use sessions and publicly open to the YOOCHOOSE servers
- A YOOCHOOSE license

This bundle is independent from the ezrecommendation extension used in legacy, and does not require it.

Installation

To install eZ Recommendation Bundle:

1. Run the following from your eZ Publish installation root (*here with most recent 1.0.x release*):

```
php composer.phar require
ezsystems/recommendation-bu
ndle:~1.0.0
```

2. Enable the bundle in `ezpublish/EzPublishKernel.php`:

```
$bundles = array(
    // existing bundles
    new
    EzSystems\RecommendationBun
dle\EzSystemsRecommendation
Bundle()
);
```

3. Import additional routing by adding following lines to your `routing.yml` file:

```
recommendationBundleRestRou
tes:
    resource:
"@EzSystemsRecommendationBu
ndle/Resources/config/routi
ng_rest.yml"
    prefix:
%ezpublish_rest.path_prefix
%
```

Configuration

The bundle's configuration is siteaccess aware. This is an example of settings (`config.yml`):

```

ez_recommendation:
  system:
    default:
      yoochoose:
        customer_id: "12345"
        license_key:
"1234-5678-9012-3456-7890"
        server_uri:
"http://example.com"
        recommender:

included_content_types: ["blog",
"article"]

```

The following parameters need to be included in the settings file:

Parameter	Description
yoochoose.customer_id	Your YOOCHOOSE customer ID.
yoochoose.license_key	Your YOOCHOOSE license key.
server_uri	The URI your site's REST API can be accessed from.
recommender.included_content_types	Content Types on which the tracking script will be shown. See Tracking below for more information.

If content's author or image are stored in different field, you can specify it in **parameters.yml**:

Format

```

ez_recommendation.field_identifiers:
  {field fetched by controller
(image or author)}
  {content type}: {field
with value}

```

For example:

Actual example

```

ez_recommendation.field_identifiers:
  author:
    article: authors
  image:
    article: thumbnail
    blog_post: main_image

```

Advanced configuration

You can select advanced options for YOOCHOOSE backend using the following settings:

```
ez_recommendation:
  api_endpoint:
  'https://admin.yoochoose.net '
  recommender:
    api_endpoint:
    'http://reco.yoochoose.net '
    consume_timeout: 20
  tracking:
    api_endpoint:
    'http://event.yoochoose.net '
    script_url:
    'cdn.yoochoose.net/yct.js '
```

Changing any of these parameters without a valid reason will break all calls to YOOCHOOSE. It can be useful to test the API by mocking the service, or if you have a hosted version of YOOCHOOSE Recommendation service.

Using the Recommendation Bundle

Initial Setup

Your content structure must be mapped to the YOOCHOOSE domain model. This must be done in collaboration with YOOCHOOSE.

Indexing

Public content is automatically indexed. When necessary, eZ Publish will notify YOOCHOOSE of changes to content. Initial import is to be managed with your YOOCHOOSE sales representative. Note that your server's REST API will need to be open to the YOOCHOOSE servers for indexing to be possible.

Tracking

Events from the site need to be sent to YOOCHOOSE so that recommendations can be adapted to visitors. Tracking can be set up in multiple ways, depending on existing constraints.

`EzRecommendationBundle` delivers a Twig extension which helps integrate the tracking functionality into your site.

To enable tracking:

1. Place the following snippet of code somewhere in the HEAD section of your header template (if your bundle is built on top of the DemoBundle this is `page_head.html.twig`):

```
{% if content is defined %}
  {{
  yc_track_user(content.id)
  }}
{% endif %}
```

2. Configure settings under the `recommender.included_content_types` parameter (see the `default_settings.yml` file delivered with this bundle). Here you can define for which content types tracking script will be shown.

You can find more information about [tracking](#) in YOOCHOOSE documentation.

Displaying

In order to allow displaying recommendations on your site you must add portions of scripts which will integrate the recommender engine with your site.

Implementation is very easy and can be performed in just a few steps (assuming that the `EzRecommendationBundle` is properly configured and enabled in `EzPublishKernel.php`):

To enable displaying recommendations:

1. Add the following JavaScript assets to your header template (if your bundle is built on top of the DemoBundle this is `page_head_script.html.twig`):

```
{% javascripts
    ...

    '%kernel.root_dir%/../vendor/
    components/handlebars.js/
    handlebars.min.js'

    '@EzSystemsRecommendationBu
    ndle/Resources/public/js/re
    commendationtemplaterendere
    r.js'

    '@EzSystemsRecommendationBu
    ndle/Resources/public/js/re
    commendationtemplatehelper.
    js'

    '@EzSystemsRecommendationBu
    ndle/Resources/public/js/re
    commendationrestclient.js'
%}
```

2. Place a dedicated Twig helper in the place where you want to display recommendations:

```
{{ yc_show_recommendations(
    contentId = content.id,
    scenario = '',
    limit = '',
    contentType = '',
    template = '',
    fields = []
) }}
```

Parameters

Parameter	Type	Description
-----------	------	-------------

contentId	int	In content-based views the Twig variable holding the content id (the content you want to get recommendations for).
scenario	string	Scenario used to display recommendations. You can create custom scenarios at the YOOCHOOSE dashboard.
limit	int	Number of recommendations to show.
contentType	string	Content Types you are expecting in response.
template	string	HandleBars template name (your templates are stored in the EzRecommendationBundle/Resources/public/views directory. Take a look at default.html.twig which includes a default template that can be used to prepare customized versions).
fields	array	Fields which are required and will be requested from the recommender engine. These field names are also used inside HandleBars templates.

Sample integration can take the following form:

```

{{ yc_show_recommendations(
  contentId = content.id,
  scenario = 'popular',
  limit = 5,
  contentType = 'article',
  template = 'default',
  fields =
  ['ez_publishedDate',
  'ez_url', 'title', 'image',
  'author', 'intro']
) }}

```

You can also bypass named arguments using standard value passing as arguments.

Item ID

The item id is usually set to the viewed ContentId. Depending on requirements, it can be set to a different value, in collaboration with YOOCHOOSE.

Troubleshooting

Most operations are logged via the `ez_recommendation` [Monolog channel](#). To log everything about Recommendation to `dev.recommendation.log`, add the following to your `config.yml`:

```
monolog:
  handlers:
    ez_recommendation:
      type: stream
      path: "%kernel.logs_dir%/%kernel.environment%.recommendation.log"
      channels:
        [ez_recommendation]
      level: info
```

You can replace `info` by `debug` for more verbosity.

Other resources

- [API doc for YOOCHOOSE Service](#)
- [Developer Guides for YOOCHOOSE Service](#)