


# Import

- [Introduction](#)
- [Content interface](#)
  - [Pushing a single item](#)
  - [Pushing an editorial list](#)
  - [Bulk interface](#)
    - [Export of all items](#)
    - [Upload an export file](#)
- [Web Interface](#)
- [Configuration interface](#)
  - [License information](#)
  - [Statistics](#)
  - [Event Summary](#)

 The following document mentions productId and customerId in the links. The productId must be replaced by the registered solution (news or ebl) and the customerId with the [technical login](#) of the registered customer.

## Introduction


The YOOCHOOSE Recommender Engine needs updated information from the web presence of the customer to generate personalized recommendations for the user profile. To get such information an event tracking process is required to collect events like clicks, purchases, consumes etc. This is described in detail in the [eZ Recommender Tracker API](#).

In addition to events eZ Recommender can also use information about the provided content in the portal or the products and their attributes in an eCommerce shop. The third chapter describes how to provide an interface that can be used to provide information about how to introduce new, update or delete content.

Providing content can be basically done in two different ways. The two alternatives for the "how" are either pushing the content by using [Import](#) API or providing a service on the customer side that allows to pull the content. The alternatives for the "what" are either providing a full dump e.g. the full product catalog or providing a single item or product.

If you are a partner or programmer who is interested in integrating configuration and reporting information from the eZ Recommender into the backend of your Content Management System or shop software the fourth chapter will show how to access some basic reporting information. Nevertheless we recommend using the [Administration Interface \(GUI\)](#) to have an insight into statistics.

## Content interface

 In order access the service endpoints the HTTP request needs to include **BASIC Authentication**. Use the [technical login](#) to access the services.

The following describes the interface which allows inserting and administrating single items or editorial lists stored in the eZ Recommender data store. Also the necessary functions that have to be provided from the customer's system to allow a pull on single items will be described in this section. The second part focuses on the bulk interface that allows either the upload or the download of a full item export from a customer's system.

## Pushing a single item


eZ Recommender provides a REST interface that accepts a single item in XML-format. The following URL describes the interface.

[https://admin.yoochoose.net/\[productid\]/\[customerid\]/item](https://admin.yoochoose.net/[productid]/[customerid]/item)

It can be used to POST item information within the request's body into the store and

[https://admin.yoochoose.net/\[productid\]/\[customerid\]/item/\[itemtypeid\]/\[itemid\]](https://admin.yoochoose.net/[productid]/[customerid]/item/[itemtypeid]/[itemid])

to display, update or delete items directly. The parameters that are used in the call are described in the table.


 The above mentioned interfaces are not officially supported and we do not guarantee full functionality and availability. They should help customers to get an insight view of their metadata

Parameter name	Description	Values
productid	This is a reference to the YOOCHOOSE product. It will be provided by YOOCHOOSE together with the client id.	String
clientid	This is a reference to the account of the customer. It will be provided by YOOCHOOSE.	String
itemid (optional)	A unique identifier for the item that is used as a reference to identify the item in the database of the customer	Numeric
itemtypeid (optional)	Describing the type of the item id. Usually it is fixed to 1 but if the customer does use more than one type of item this has to refer to the corresponding item type	Numeric

Different HTTP-methods can be used to create, update, delete or retrieve items located in the YOOCHOOSE data store. The following table gives an overview about the different methods and their function.

HTTP Method	Description	Result
POST	If the body contains valid xml data, the item will be persisted. The item is not directly available but scheduled to be inserted. If the content cannot be validated, the server will send a Bad Request status code.	202 (Accepted) 400 (Bad Request)

GET	This method retrieves all information that is stored in the database for the given item id. If not found the status code 404 is returned	200 (OK)404 (Not Found)
DELETE	Deletes all information that is related to the item id that has been sent. There is no need to send a body element. The item is not deleted directly but scheduled to be removed from the data store.	202 (Accepted)404 (Not Found)

 In general, the client should check the response code of any push service call. In case of an unexpected response code (other than denoted in the table above), one should

1. check the request (in case of status codes 4xx)
2. retry the service call later (in case of status codes 5xx)

The body of a request to import data using the above interface must contain a valid XML document. Here's a simple example:

```

<?xml version="1.0" encoding="UTF-8"?>
<items version="1">
  <item id="102" type="1">
    <description>the item's description (optional) </description>
    <!-- price as integer value (e.g. EUR-Cent) -->
    <price currency="EUR">122</price>
    <validfrom>2011-01-01T00:00:00</validfrom>
    <validto>2021-01-01T00:00:00</validto>
    <!-- category path(s) that lead to items -->
    <categorypaths>
      <!-- one item can have n category paths -->
      <categorypath>/8/4/5</categorypath>
      <categorypath>/84</categorypath>
      <categorypath>/1/847</categorypath>
    </categorypaths>
    <content>
      <content-data key="title">
        <![CDATA[;;;<<>> values need to be escaped]]>
      </content-data>
      <content-data key="abstract"> <![CDATA[<html><head><title>abstract of the
text</title></head><body>body ä </body></html>]]>
      </content-data>
    </content>
    <attributes>
      <attribute key="author" value="Max Mustermann" />
      <attribute key="agency" value="dpa" />
      <attribute key="vendor" value="BOSCH" />
      <!-- if NUMERIC is chosen, the value must be numeric as well. Otherwise the
attribute will be discarded -->
      <attribute key="size" value="5" type="NUMERIC" />
    </attributes>
  </item>
</items>


```

Depending on the product, there are several mandatory fields defined that will be semantically interpreted by the YOOCHOOSE recommender engine (e.g. filtering based on the price). The following table provides an overview about the predefined attributes.

Name/Attribute	Description	Type	Mandatory for
id	The id of an item	Integer	publisher, shop
type	The type of an item	Integer	publisher, shop
description	Some additional information about an item	String	
price	The price of an item (e.g. in EUR cents)	Integer	shop
currency	Currency used, by default EUR is assumed	ISO 4217	shop

validfrom	Defines together with validto the "lifespan" of an item.If NULL or not available, the item is valid	ISO 8601	publisher
validto	Defines together with validfrom the "lifespan" of an item.If NULL or not available, the item is valid	ISO 8601	publisher
categorypath	A navigation path how the item can be reached in the customer's system	String	publisher, shop

In order to improve recommendations, item data can be enriched with meta data that is not mandatory. This data could for example include information about an agency, author, geo location, vendor etc. To insert these values in the YOOCHOOSE data store, the following keys can be used in the **<attribute>** and **<content>** elements. eZ Recommender will interpret this information during the calculation of models.

 The value of the key attribute in the elements **<attribute>** and **<content-data>** must contain only letters, numbers and underscores. Otherwise these attributes will be **discarded!**

Key	Description	Type	Section
author	The name of an author	String	attributes
agency	The content provider	String	attributes
geolocation	geo location data	String	attributes
vendor	The vendor of an item	String	attributes
date	The creation/import date of an item	ISO 8601	attributes
title	The title of an item	String	content
abstract	The abstract of an item	String	content
tags	Comma separated list of tags for an item	String	content

It is also possible to define custom (numeric or nominal) attributes in the **<attributes>** section. The default type of every attribute is "NOMINAL", i.e. the values of an attribute are treated as distinct when compared while calculating a content based model, which is available in the ADVANCED solution. If you add another attribute in the attribute element named *type="NUMERIC"*, the recommender engine will treat the values as ranges. This means that a size of 4 is closer to a size of 5 than to 1. If the attribute price were of type NOMINAL, they were both just different and had no "distance based similarity".

```
<attribute key="size" value="4" type="NUMERIC" />
```

Another typical example is the color of an item. To insert it in the store, the following line should be a child of the <attributes> element.

```
<attribute key="color" value="green" />
```

## Pushing an editorial list

Editorial lists are predefined lists that can be referenced while fetching recommendations. If no recommendations can be provided due to too less events or not yet calculated models these lists can be used to provide predefined recommendations. Another useful way of using editorial lists is e.g. pushing the selling or reading of certain items. Items configured in this list can be rendered on certain landing pages.

To import editorial lists, YOOCHOOSE provides the following URL to insert lists in the datastore.

[https://admin.yoochoose.net/\[productid\]/\[customerid\]/editoriallist](https://admin.yoochoose.net/[productid]/[customerid]/editoriallist)

As an example, the following content could be used in a request's body to insert an editorial list in the YOOCHOOSE data store.

```
<?xml version="1.0" encoding="UTF-8"?>
<editoriallist version="1" referencecode="top_products_list">
  <listitem itemid="123" itemtype="3" relevance="13" />
  <!-- by default the relevance is 1 -->
  <listitem itemid="1234" itemtype="2" />
</editoriallist>
```

The relevance allows the client to prioritize its recommendations. If e.g. an item has a relevance of 50, half of the recommendations based on this list will contain this item. In order to access or delete editorial list data directly, the following URL can be used

[https://admin.yoochoose.net/\[productid\]/\[customerid\]/editoriallist/\[referencecode\]](https://admin.yoochoose.net/[productid]/[customerid]/editoriallist/[referencecode])

The parameters that are used in the call are described in the table.

Parameter	Description	Values
productid	This is a reference to the YOOCHOOSE product. It will be provided by YOOCHOOSE together with the client Id.	String
clientid	This is a reference to the account of the customer. It will be provided by YOOCHOOSE.	String

referencecode	The identifier of a category	String
---------------	------------------------------	--------

After an editorial list import, YOOCHOOSE will respond with the following status codes.

HTTP Method	Description	Result
POST	If the body contains valid xml data, the editorial list will be persisted. If the request is accepted, the editorial list is not directly available but scheduled to be inserted. If the content cannot be validated, the server will send a Bad Request status code.	202 (Accepted) 400 (Bad Request)
GET	This method retrieves all information that is stored in the database for the given editorial list. If not found the status code 404 is returned	200 (OK)404 (Not Found)
DELETE	Deletes the editorial list defined by the reference code. There is no need to send a body element. The editorial list is not deleted directly but scheduled to be removed from the data store.	202 (Accepted)404 (Not Found)

**i** In general, the client should check the response code of any push service call. In case of an unexpected response code (other than denoted in the table above), one should

1. check the request (in case of status codes 4xx)
2. retry the service call later (in case of status codes 5xx)

## Bulk interface

The bulk interface is mainly used to provide an initial data import to initialize the eZ Recommender system. It also can or should be used to reset and re-initialize the the data store.

## Export of all items

To provide a full export of the item database use the same format like for the single item interface. Provide it for the intended timeframe (e.g. last year) as one file and separate it later into smaller chunks to upload it as described in the next chapter.

## Upload an export file

To enable the upload provide a download location that allows the eZ Recommender system to retrieve the XML document containing the item data. In order to get also high amount of data, the export should be separated into smaller files not bigger than 20 MB each. The extension of the filename should reflect the format of the file (e.g. XML).


<http://cms.customer.de/path-to-the-service/download-folder/<filename.extension>>

To initiate the download or respectively the upload the customer invokes the following trigger service where the URL is reflecting the address where the download files can be found. If more than one file is provided for download, this trigger service has to be invoked once for each file with the corresponding file URL.

[https://import.yoochoose.net/\[productid\]/\[clientid\]/item/upload?url=<url>](https://import.yoochoose.net/[productid]/[clientid]/item/upload?url=<url>)

Parameter	Description	Values
productid	This is a reference to the product. It will be provided by YOOCHOOSE together with the client Id.	String
clientid	This is a reference to the account of the customer. It will be provided by YOOCHOOSE.	String
url	The URL from where the export file can be downloaded by YOOCHOOSE service components.	URL-encoded String [2]

After invoking the trigger service, download and further processing will be scheduled asynchronously. Therefore, the usual response code for this service call is 202 (Accepted). The download will take place within the next hours.

 Currently Authentication is not supported to fetch customer downloads. Use cryptic filenames or delete the download after a while ..

## Web Interface

YOOCHOOSE provides a Web Interface in order to access or update customer's data. It can be reached under

<https://admin.yoochoose.net>

You can register or use your already existing account to log in.

## Configuration interface

The configuration services are mainly used to integrate basic information about the eZ Recommender service into a backend system of the customer. The following chapters will give a brief overview about the interfaces. All configuration service methods described in this section are secured with BASIC authentication . Therefore, an authentication header with [technical login data](#) must be present to access the configuration service.

The authentication tokens user-id and password are provided in the [Adminstration Interface \(GUI\)](#). Use the "Customer Id" as user and "License Key" for password.

## License information

Using the following URL will retrieve the license



[https://admin.yoochoose.net/\[productid\]/\[customerid\]/license.json](https://admin.yoochoose.net/[productid]/[customerid]/license.json)

The following table describes the different parameters:

Parameter	Description	Values
productid	This is a reference to the product. It will be provided by YOOCHOOSE together with the client Id.	String
clientid	This is a reference to the account of the customer. It will be provided by YOOCHOOSE.	String

The result of the above call will be JSON formatted and looks like the example below:

```
{
  license: {
    type: "TRIAL"
  }
}
```

## Statistics

With the following methods some basic statistics about the data that is stored in the YOOCHOOSE system can be retrieved.

## Event Summary

In order to gather statistical information about the different types of events for the last seven days (click, buy, recommendations etc.) the following URL can be used. It is a JSON response that can be used to integrate the presentation of statistics within a customer's system.

[https://config.yoochoose.net/\[productid\]/v2/\[customerid\]/revenue/last\\_seven\\_days](https://config.yoochoose.net/[productid]/v2/[customerid]/revenue/last_seven_days)

Parameter	Description	Values
productid	This is a reference to the product. It will be provided by YOOCHOOSE together with the client Id.	String
clientid	This is a reference to the account of the customer. It will be provided by YOOCHOOSE.	String

The result of the above call will be JSON formatted and looks like the example below:

```
{
  revenueResponse: {
    items: [
      {
        currency: null
        revenue: 0
        timespanBegin: "2011-08-28T00:00:00"
        timespanDuration: "P1D"
        clickEvents: 212024
        purchaseEvents: 1861
        deliveredRecommendations: 760
        clickedRecommendations: 13657
        purchasedRecommendations: 0
      },
      {
        currency: null
        revenue: 0
        timespanBegin: "2011-08-29T00:00:00"
        timespanDuration: "P1D"
        clickEvents: 16339
        purchaseEvents: 105
        deliveredRecommendations: 0
        clickedRecommendations: 901
        purchasedRecommendations: 0
      }
    ]
    clientId: "[clientId]"
  }
}
```