# EXT: RSS feed

The content of this document is related to TYPO3

- a GNU/GPL CMS/Framework available from www.typo3.com

# Table of Contents

# Introduction

## What does it do?

This extension is designed to set up **quickly** and **easily** one or many rss feeds from any tables of the database. Copy / paste a few TypoScript lines in your template setup and that's it! You can follow the latest content of your website – keep an eye on the latest frontend user, etc.

**Features**

– Choose whether ATOM or RSS feed is displayed:
  – http://en.wikipedia.org/wiki/Atom_(standard)
  – http://en.wikipedia.org/wiki/RSS_(file_format)
– Extract value from flexform (requires MySQL 5.2). Some values can be stored in flexform data. This information is not very practical to extract. However with MySQL 5.2, it is now possible to build XPath requests and fetch data within the XML structures.

– Hide / show several different feeds on the same pages. RSS feeds are very practical to follow the life of a website. You can imagine a feed that informs you whether a new page is created, a backend user's profile has been updated or a content element has been updated. Such administrative feeds are not necessarily shown in the frontend but are given only for administrators.

– A lot of possible data combinations and configurations...

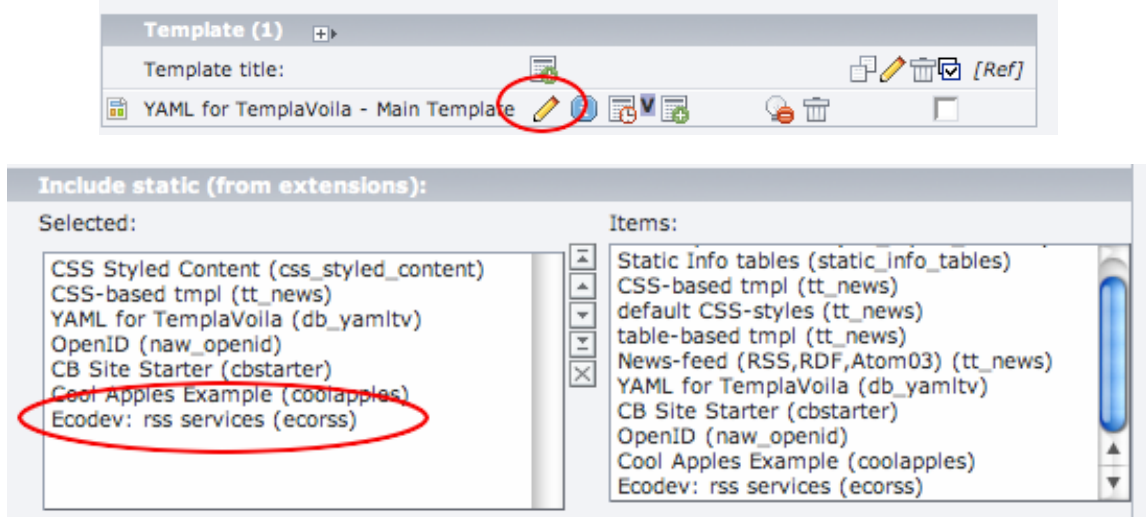**Further documentation of RSS:** http://www.rssboard.org/rss-specification

# No screenshots necessary!

@see rather the TypoScript example in the next section.

# Installation

Download and activate the extension as normal. No database update necessary.

Update your "Static Include" from extension by editing the main template (module Web > List) as follows:





# User guide

A backend user can exclude a content element from the feed. A checkbox is available after the field "hidden".s

In order to generate the RSS on  a page, this ending must be added to the page url: **/?type=103**.

The number must be identical to your setup.

# How to Build a Feed, TypoScript Example

Very Important: This extension won't work until you properly complete the installation section.

## Express Installation

Copy / paste these lines into a template > setup. Beware of the cache period. The feed is regenerating every 3600 seconds. To change this value, check parameter "cache_period".

```
#############################
# GENERATE THE FEED
#############################
feed = PAGE
feed {
  typeNum = 103
  10 >
  10 = USER
  10.userFunc = JambageCom\Ecorss\Controller\FeedController->display
  10.lang = fr-FR
  # CHANGE THIS:
  10.title = title of the feed
  # CHANGE THIS:
  10.subtitle = subtitle of the feed
  config{
    disableAllHeaderCode = 1
    disableCharsetHeader = 1
    additionalHeaders.10.header = Content-Type:application/rss+xml;charset=utf-8
    no_cache = 1
    xhtml_cleaning = 0
    admPanel = 0
    disablePrefixComment = 1
```

```
    linkVars >
  }

}

#############################
# DECLARE FEED IN THE FRONTEND(ATOM)
#############################
page.headerData.12 = USER
page.headerData.12.userFunc = JambageCom\Ecorss\Controller\FeedController->add
page.headerData.12.feed{
  typeNum = 103
  title < feed.10.title
}
```

## More Details: Declaring a RSS or ATOM Feed in the Frontend

### Declaring two different feeds onto the frontend

Copy paste these lines in a template > setup.

```
page = PAGE
page.headerData.12 = USER
page.headerData.12.userFunc = JambageCom\Ecorss\Controller\FeedController->add
page.headerData.12.feed1{
  title = Here comes the title of the feed 1
  typeNum = 103
  feed = atom
}
page.headerData.12.feed2{
  title = Here comes the title of the feed 2
  typeNum = 100
  feed = rss
}
```

## Plugin Global Configuration Constants

| Property: | Data type: | Description: | Default: |
|---|---|---|---|
| feedType | int (mandatory) | the type of the page like index.php?id=1&type=121 | null |
| title | string | the title of the feed | |
| subtitle | string | description | |
| feed | string | 2 possibles values: atom – rss<br>have a look at the following links to know the difference of the formats:<br>● http://en.wikipedia.org/wiki/Atom_(standard)<br>● http://en.wikipedia.org/wiki/RSS_(file_format) | atom |

### Important Remark

In case you have several feeds, you might want define constants to avoid repeating things.

In the constant editor:

```
plugin.tx_ecorss.title = Here comes the *title* for every feeds
plugin.tx_ecorss.subtitle = Here comes the *subtitle* for every feeds
...
```

## Feed Configuration

| Property: | Data type: | Description: | Default: |
|---|---|---|---|
| title | string | the title of the feed | atom |
| subtitle | string | the subtile of the feed | |
| lang | string | the character encoding: fr-FR ; de-DE ; en-US | en-GB |
| host | string | define an other host name t if not the same as t3lib_div::getIndpEnv('TYPO3_SITE_URL'). If protocol (http:// or https://) is not specified, then http:// will be prefixed. | t3lib_div::getIndpEnv('TYPO3_SITE_URL') |
| encoding | string | | utf-8 |
| numberItems | string | number of elements in the feed | 10 |

| Property: | Data type: | Description: | Default: |
|---|---|---|---|
| feed | string | 2 possibles values: atom ; rss<br>have a look at the following links to know the difference of the formats:<br>● http://en.wikipedia.org/wiki/Atom_(standard)<br>● http://en.wikipedia.org/wiki/RSS_(file_format) | atom |
| author.name | string | the author of the feed's (Atom feed).<br>**Remark:** If you want to save this value automatically in the database, have a look at the extension "fhm_author". | Page's author name |
| author.email | string | the email of the author (Atom feed). Beware: default Atom template does not write the email, you have to modify file templates/atom.php)<br>**Remark:** If you want to save this value automatically in the database, have a look at the extension "fhm_author". | Page's author email |
| cache_period | int | How long should the feed be hold in cache. Unit in second | 3600 |
| no_anchor | int | Define if the URL will contain an anchor like this :<br>http://mydomaine.tld/index.php?id=123#c343 | 0 |
| pidRootline | int | which page id define the first page of the Site (update on this page will be included too in the feed). It is useful with multi domain websites or to create feeds related to sections in the website | null |
| sysLanguageUid | int | which language (L parameter in multilingual sites) should be used to filter updated content. If left unconfigured, return all content in all languages | null |
| profileAjaxType | int | which page type should be used to include website content in the feed (useful to change the page type to use a stripped down layout – e.g., w/o menu – when reading the news in a newsreader such as Thunderbird) | 0 |
| hook | user defined | If you create a hook for post-processing feed entries, you may configure it as you wish using this section. See section "Available Hooks" for more information. | null |
| tidy | int | Launch the tidy command | 0 |
| tidy_path | string | Configuration parameters given to tidy | tidy -i -utf8 -xml |
| parseFunc | →> parseFunc | See TYPO3 docu | |

## Elements of the Feed Configuration (select.)

| Property: | Data type: | Description: | Default: |
|---|---|---|---|
| title | string | the field of the database that corresponds to the "title" of an element in the feed. Most of the time, it will be tt_content.header | header |
| titleXPath | string | the field where the flexform is stored. Most of the time, it will be content.pi_flexform | pi_flexform |
| summary | string | the field of the database that corresponds to the "body of an element in the feed. Most of the time, it will be tt_content.bodytext | bodytext |
| summaryXPath | string | the field of the database that corresponds to the "body of an element in the feed. Most of the time, it will be tt_content.bodytext | utf-8 |
| published | string | the field of the database that corresponds to the time stamp when the element is published | tstamp |
| updated | string | the field of the database that corresponds to the time stamp when the element is updated | tstamp |
| uid | string | the uid field name of the database | uid |
| pid | string | the page id's field name of the database | pid |
| filterField | string | define which field will be used to filter elements. Will be transform in a SQL statement like field = value => list_type = some_plugin | null |
| filterInclude | comma separated string | define which value will be used to filter elements. Will be transform in a SQL statement like field = value => list_type = some_plugin<br>E.g.: tx_ecorss_controllers_feed | null |
| filterExclude | comma separated string | define which value will **NOT** be used. Will be transform in a SQL statement like list_type **!=** some_plugin | |
| table | string | which table contains the element of the feed | tt_content |
| debug | true/false | output the request in the RSS feed. Maybe useful to debug the request. | false |
| linkItem | true/false | wrap the title with a link. It is not always the case since in can be a administrative feed telling that a new user has been created for example. | true |
| defaultText | string | should the element contain a default value. Useful for administrative feed. | null |

| Property: | Data type: | Description: | Default: |
|---|---|---|---|
| single_page.pid | int | page id containg the plugin to display a single record as defined in "select" configuration element. | null |
| single_page.linkParamUid | string | parameter to the url to choose the record to be shown. E.g. if you use the extension mininews, you will have to set this to "tx_mininews_pi1[showUid]" as the URL to show a single news is the URL of the page containing the news plugin and the parameter &tx_mininews_pi1[showUid]=<uid> | null |
| orderBy | string | defines the order of the feed. | tstamp DESC |
| where | string | defines additional conditions ( where = AND myField = 'myValue') | null |
| truncate | int | defines how many character the summary should be shorten up. | null |
| author | string | the field of the database that corresponds to the "author". If not defined, retrieves the author from the page. | empty |
| extraFields | string | additionnal fields<br>extraFields = datefrom, dateto | empty |
| additionalSQL | string | example: adding categories management for tt_news<br>`AND tt_news.uid IN (SELECT tt_news_cat_mm.uid_local FROM tt_news_cat_mm WHERE tt_news_cat_mm.uid_foreign = 5)` | empty |
| additionalPids | string | comma separated string. Defines additional pid to include in the request. Useful for adding external sysfolder of the root page. | empty |
| baseUrl | string | Ecorss tries to guess the domain according to the pid of the record but in some cases it can't succeed as the record may be outside the tree (e.g. tt_news). So gives ecorss a little hand by giving the base URL.<br>Example of value:<br>baseUrl = http://replace.tld | empty |

# Feed Configuration Examples

## Generating a RSS or ATOM feed

The minimum configuration to build a feed. Copy paste these lines in a template > setup. It will display the latest updated content from table tt_content with 10 items.

```
feed2 = PAGE
feed2 {
  typeNum = 103
  10 >
  10 = USER
  10.userFunc = JambageCom\Ecorss\Controller\FeedController->display
  10.title = title of the feed
  10.subtitle = subtitle of the feed
  10.lang = fr-FR
  config {
    disableAllHeaderCode = 1
    disableCharsetHeader = 1
    additionalHeaders.10.header = Content-Type:application/rss+xml;charset=utf-8
    no_cache = 1
    xhtml_cleaning = 0
    admPanel = 0
    disablePrefixComment = 1
    linkVars >
  }
}
```

### Generating a RSS or ATOM feed (more configuration)

It will provide exactly the same output as "feed2" but with more configuration

```
feed3 = PAGE
feed3 {
  typeNum = 103
  10 >
  10 = USER
  10.userFunc = JambageCom\Ecorss\Controller\FeedController->display
  10.title = title of the feed
  10.subtitle = subtitle of the feed
  10.lang = fr-FR
```

```
    10.cache_period = 3600
    10.numberItems = 10
    10.feed = atom
    10.select {
        01 {
            table = tt_content
            title = header
            summary = bodytext
            published = tstamp
            updated = tstamp
            debug = 0
        }
    }
    config {
        disableAllHeaderCode = 1
        disableCharsetHeader = 1
        additionalHeaders.10.header = Content-Type:application/rss+xml;charset=utf-8
        no_cache = 1
        xhtml_cleaning = 0
        admPanel = 0
        disablePrefixComment = 1
        linkVars >
    }
}
```

## Filtering Content based on Page Language

Only return entries for the language associated to L=1 (e.g., English).

```
feed4 = PAGE
feed4 {
    typeNum = 103
    10 >
    10 = USER
    10.userFunc = JambageCom\Ecorss\Controller\FeedController->display
    10.title = title of the feed
    10.subtitle = subtitle of the feed
    10.lang = en-GB
    10.sys_language_uid = 1
    config {
        disableAllHeaderCode = 1
        disableCharsetHeader = 1
        additionalHeaders.10.header = Content-Type:application/rss+xml;charset=utf-8
        no_cache = 1
        xhtml_cleaning = 0
        admPanel = 0
        disablePrefixComment = 1
        linkVars >
    }
}
```

## Filtering Records from Table tt_content

```
feed5 = PAGE
feed5 {
    typeNum = 103
    10 >
    10 = USER
    10.userFunc = JambageCom\Ecorss\Controller\FeedController->display
    10.title = title of the feed
    10.subtitle = subtitle of the feed
    10.lang = fr-FR

    10.select {
        01 {
            table = tt_content
            # it will be transformed in SQL for list_type=tx_ecorss_controllers_feed
            filterField = list_type
            # comma separated value
            filterInclude = tx_ecorss_controllers_feed
        }
    }
    config {
        disableAllHeaderCode = 1
        disableCharsetHeader = 1
        additionalHeaders.10.header = Content-Type:application/rss+xml;charset=utf-8
        no_cache = 1
```

```
      xhtml_cleaning = 0
      admPanel = 0
      disablePrefixComment = 1
      linkVars >
  }
}
```

## Including and Excluding Field's Value

Include and exclude field's value in a more verbose configuration.

```
feed6 = PAGE
feed6 {
  typeNum = 103
  10 >
  10 = USER
  10.userFunc = JambageCom\Ecorss\Controller\FeedController->display
  10.title = title of the feed
  10.subtitle = subtitle of the feed
  10.lang = fr-FR
  10.cache_period = 3600
  10.numberItems = 10
  10.feed = atom
  10.select {
    01 {
      table = tt_content
      filterField = list_type
      filterInclude = text, textpic
      filterExclude = list, mailform
      # the value corresponds to a field (title of the enty)
      title = header
      # the value corresponds to a field (body of the enty)
      summary = bodytext
      debug = 0
    }
  }
  config {
    disableAllHeaderCode = 1
    disableCharsetHeader = 1
    additionalHeaders.10.header = Content-Type:application/rss+xml;charset=utf-8
    no_cache = 1
    xhtml_cleaning = 0
    admPanel = 0
    disablePrefixComment = 1
    linkVars >
  }
}
```

## Selecting Content from a Flexform

Requires MySQL >= 5.2.

```
feed7 = PAGE
feed7 {
  typeNum = 103
  10 >
  10 = USER
  10.userFunc = JambageCom\Ecorss\Controller\FeedController->display
  10.title = title of the feed
  10.subtitle = subtitle of the feed
  10.lang = fr-FR
  10.cache_period = 3600
  10.numberItems = 10
  10.feed = rss
  10.select{
    01{
      table = tt_content
      filterField = list_type
      filterInclude =  tx_ecorss_controllers_feed
      # this option is not mandatory
      summary = pi_flexform
      summaryXPath = /T3FlexForms/data/sheet[@index="sSummary"]/language/field/value[text()]
    }
  }
  config{
    disableAllHeaderCode = 1
    disableCharsetHeader = 1
```

```
    additionalHeaders.10.header = Content-Type:application/rss+xml;charset=utf-8
    no_cache = 1
    xhtml_cleaning = 0
    admPanel = 0
    disablePrefixComment = 1
    linkVars >
  }
}
```

## Combining Content from a Flexform and From Another Table

Select content from a Flexform (require MySQL 5.2) AND combine with an other feed from an other table (tx_mininews_news) associated to extension mininews.

```
feed8 = PAGE
feed8 {
  typeNum = 103
  10 >
  10 = USER
  10.userFunc = JambageCom\Ecorss\Controller\FeedController->display
  10.title = title of the feed
  10.subtitle = subtitle of the feed
  10.lang = fr-FR
  10.cache_period = 3600
  10.numberItems = 10
  10.select {
    01 {
      table = tt_content
      filterField = list_type
      filterInclude = tx_ecorss_controllers_feed
      summaryXPath = /T3FlexForms/data/sheet[@index="sSummary"]/language/field/value[text()]
    }
    02 {
      table = tx_mininews_news
      title = title
      summary = teaser
      published = tstamp
      updated = tstamp
      debug = 0

      single_page {
        pid = 87
        linkParamUid = tx_mininews_pi1[showUid]
      }
    }
  }
  config {
    disableAllHeaderCode = 1
    disableCharsetHeader = 1
    additionalHeaders.10.header = Content-Type:application/rss+xml;charset=utf-8
    no_cache = 1
    xhtml_cleaning = 0
    admPanel = 0
    disablePrefixComment = 1
    linkVars >
  }
}
```

## tt_news example

```
feed = PAGE
feed {
  typeNum = 103
  10 = USER
  10 {
  userFunc = JambageCom\Ecorss\Controller\FeedController->display
  lang = fr-FR
  10.title = REPLACE
  10.subtitle = REPLACE
    pidRootline = REPLACE
    tidy = 1
    numberItems = 10
    select {
      01 {
        table = tt_news
        title = title
        summary = short
```

```
        published = crdate
        updated = tstamp
        debug = 0
        additionalSQL = AND tt_news.uid IN (SELECT tt_news_cat_mm.uid_local FROM tt_news_cat_mm WHERE
tt_news_cat_mm.uid_foreign = 5)
        additionalPids = 27

        single_page {
          pid = 483
          linkParamUid = tx_ttnews[tt_news]
        }
      }
    }
  }

  config{
    disableAllHeaderCode = 1
    disableCharsetHeader = 1
    additionalHeaders.10.header = Content-Type:application/rss+xml;charset=utf-8
    no_cache = 1
    xhtml_cleaning = 0
    admPanel = 0
    disablePrefixComment = 1
    linkVars >
  }
}
```

# Available Hooks

One single hook, namely PostProcessingProc, is available and let you post-process each feed entry before displaying / caching it.

## Configuration

Create a PHP class containing your own processing method (see section "Class Example" below) and reference it in localconf.php:

```
include(dirname(__FILE__).'/../fileadmin/class.user_ecorss.inc');
$TYPO3_CONF_VARS['EXTCONF']['ecorss']['PostProcessingProc'][] = 'user_ecorss->proc';
```

**Remark**

This configuration example uses a file class.user_ecorss.inc which is stored in fileadmin. You may change it as you wish to fit your needs!

## Class Example

This example shows how to create a post-processing method which prepends "Hello - " to the feed title.

```
<?php
class user_ecorss {
  public function proc($feed, $cObj) {
    $feed['entry']['title'] = 'HELLO - ' . $feed['entry']['title'];
  }
}
?>
```

## TypoScript Configuration

If you need to parameterize the post processing function, you may extend the feed configuration with any parameter in the hook block:

```
feed1.10.hook {
  myPrefix = Hello World ::
}
```

And then in your class:

```
$feed['entry']['title'] = $feed['config']['myPrefix'] . $feed['entry']['title'];
```

# Administration

## Clear the cache

Add parameter `clear_cache=1` in order to clear the cache and regenerate the content of the feed.

Example URL: http://quart-monde.ch/index.php?type=103&clear_cache=1

# Known problems

–    Please report the problem you might meet to [franz.holzinger@](mailto:franz.holzinger@)ttproducts.de

# To-Do list

–    Manage the <keyword> tag in RSS feed (depending on the demand)

# Acknowledgments

–

# Changelog

Please refer to the Changelog.txt in the extension itself.