EXT: Document database

Extension Key: doc_db

Language: en

Keywords: forEditors, extJS, fe, search, adding category to page

Copyright 2009, Laurent Cherpit, <laurent@eosgarden.com>

This document is published under the Open Content License
available from http://www.opencontent.org/opl.shtml

The content of this document is related to TYPO3
- a GNU/GPL CMS/Framework available from www.typo3.org

Table of Contents

Introduction

What does it do?

The Document database extension (doc_db) act as a document database. It extends the logic of pages, adding various sorts of descriptors (type of doc, institutional owner, status), including a fully flexible tree of categories (desscriptors).

A search plugin is also featured, that allows users to search for a page. Moreover, the plugin can be configured in the backend, to deliver pre-defined lists of documents (page type, doktype) to the frontend.

That new version (0.3.0) is almost a complete overhaul of version 0.2.3. Indeed, the front-end search form was not very user friendly. Browsing through the tree of descriptors (categories) was not easy and the page was reloaded every time that a branch was opened and there was no way to search for categories without guessing where they could be. So, the work was to improve the way to find what we seek.

The frondend plugin was built with extJS library and use Ext.Direct Remoting technology wich is provided by jw_extjsdirect extension as TYPO3 service.

Screenshots

Here some screenshots of the result in frontend.

The search form.

All data are asynchronously loaded and available items are related to the criteria of selection. Only those related to documents are visible.

Grid result

Credits

The development of this extension was commissioned and funded by the WCC (World Council of Churches).

Keeping the developer happy

If you like this extension, do not hesitate to rate it and if not, too bad. You may also take a step back and reflect about the beauty of sharing.

Installation

Installing the extension

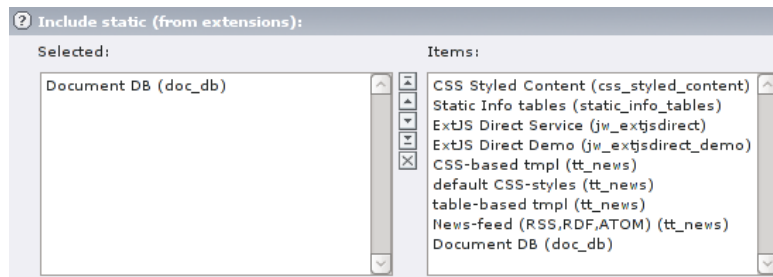The installation is a pretty straightforward procedure:

1. download and install the extension using the Extension Manager. Create db tables.
   Also, install the necessary extensions requested during the installation process. The extension manager will let you know which ones.

   If you have an older version installed, please jump to the next section of the manual: uprading
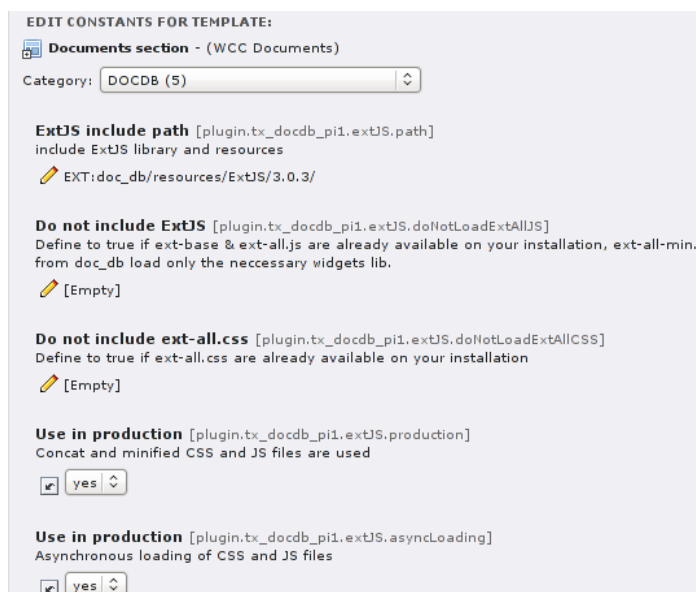
2. Create the sysFolders storage, one for the Descriptors definition (owner,type,status) and one for the Descriptors. Or only one, if you do not have many items.

   [1651] Descriptor definition
   [7265] Descriptors
   [1054] WCC General Secreta
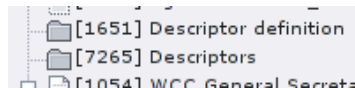
3. Go back to the extension manager and edit the values needed to set.

   1. Document Definition storage page Id

   2. Descriptors(cat) storage page Id

   3. Read the message with ref. By the red number.

4. Include the extension's static TS template (Document DB (doc_db)) into your template.
   You don't need to include static TS from 'ExtJS Direct Service'.

   Include static (from extensions):

   Selected: Items:
   Document DB (doc_db)   CSS Styled Content (css_styled_content)
                                       Static Info tables (static_info_tables)
                                       ExtJS Direct Service (jw_extjsdirect)
                                       ExtJS Direct Demo (jw_extjsdirect_demo)
                                       CSS-based tmpl (tt_news)
                                       default CSS-styles (tt_news)
                                       table-based tmpl (tt_news)
                                       News-feed (RSS,RDF,ATOM) (tt_news)
                                       Document DB (doc_db)

5. You can set the Typoscript contants with the constant editor. The default values are fine for production.
   See the configuration section: configration

   EDIT CONSTANTS FOR TEMPLATE:

   Documents section - (WCC Documents)

   Category: DOCDB (5)

   ExtJS include path [plugin.tx_docdb_pi1.extJS.path]
   include ExtJS library and resources
   🖉 EXT:doc_db/resources/ExtJS/3.0.3/

   Do not include ExtJS [plugin.tx_docdb_pi1.extJS.doNotLoadExtAllJS]
   Define to true if ext-base & ext-all.js are already available on your installation, ext-all-min.js
   from doc_db load only the neccessary widgets lib.
   🖉 [Empty]

   Do not include ext-all.css [plugin.tx_docdb_pi1.extJS.doNotLoadExtAllCSS]
   Define to true if ext-all.css are already available on your installation
   🖉 [Empty]

   Use in production [plugin.tx_docdb_pi1.extJS.production]
   Concat and minified CSS and JS files are used
   ☑ yes

   Use in production [plugin.tx_docdb_pi1.extJS.asyncLoading]
   Asynchronous loading of CSS and JS files
   ☑ yes

6. Add initial data, insert the Document DB plugin as content element on your page and set the parameters provided by flexform. All of that is explained in the section: Users Manual.

Upgrading

There are some minor things to pay attention to when upgrading from older versions

Upgrading from version 0.2.3 and below

Some database fields need to be renamed to work with the new version.

After install, you just need to set the "Document Definition storage page Id" in extension manager and click update at the bottom of the form of settings.

After that, choose the UPDATE menu from the selectbox above the form or click on update link on the right top. After follow instructions on the screen.
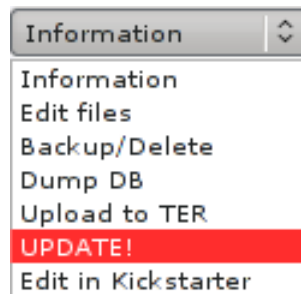
Some database tables will be also renamed. But first they will be copied and renamed with the prefix zzz_deleted followed the old name. (if zzz_ tables already exists, a datetime prefix is added and to not overwrite data) Then existing data is imported into the new tables. It will be handled by the update process.

> note: *if you reinstall an older version after the new version and after that, you re-install this version. you could loose some data. (eventually in the field named status from tx_docdb_status table). but you can find this data in field named zzz_deleted... (with phpmyadmin)*

The updater panel show you if some update are required.

If you have some SQL queries related (ex. in pageTS or tmplTS) to those old tables, the tablenames should be adapted.

Users Manual

Adding initial data

First add "descriptors definition" (owner,type,status)

In the sysFolder that you have created, corresponding to the "Document Definition storage page Id" define in extension manager, add the initial data that you want to have that will serve to define your documents.

And add Descriptors (categories)

In the sysFolder that you have created, corresponding to the "Descriptors(cat) storage page Id" define in extension manager, add the initial descriptors items that you want to have that will serve to define your documents.

Add a descriptor name and select a parent if any.	On the right tab, you can select one or several related desccriptors.

To make the search easier for a descriptor, a wizard is available. It allows a multiple selection before imported into the form.

Click on the magnifying glass icon to open it.

The WCC have about fifteen thousand of these descriptors. Indeed, it can be hard to use hierachical tree to select what you seek.

Once you have a bunch of data, now you are ready to set these data as attributes of a document (type of page:Document)

7

Set page type as Document

Create a new page or choose an existing page. In the page properties, go to the field "Type" select "Document". You should see the next screen or not so far of that:

The properties used in frontend are represented by green bullets.

A.	Type of page: Document
B.	Last update datetime
C.	Document Title
D.	Institutional Owner
E.	Document Type
F.	Document Status
G.	Document Key ref.
H.1	Document Descriptor-s
I.	Related Page-s

1. Descriptor-s: As you can see. Unlike the proposed method of selection for editing and managing descriptors where we must be able to select the parent node, here only the descriptors that are leaves (end of branch) are selectable.

To assist in research, the descriptors search wizard is also available. It display the result that contain only those descriptors leaves.

Add plugin

Add plugin as a content element

Document database (doc_db) plugin can be inserted as a traditional content element on a page, just like any other plug-in. Just choose the right type either from the plug-ins list in the content element form or from the new content wizard.

Setting up the plugin

Two mode ara available:

FE: search form in frontend view. and PI: Pre-defined searches.
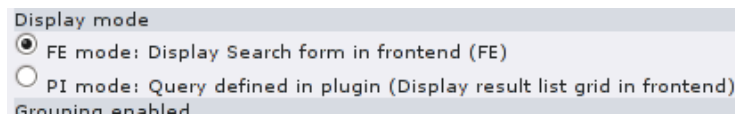
All available parameters can be set using the provided flexform. The flexform is split into two tabs.

1. General : the properties that can be set in each mode are described below.

2. display options: the properties that can be set are described here.

The meaning of the options are easy to understand.

CSS ID: Is the ID name of the div that will be used to get the content that will be diplayed in a extJS popup window. It is used to not loose the context of the list grid of search result. To provide div ID is **requiered** to not display all the page of the target document page. But to select only the desired content.
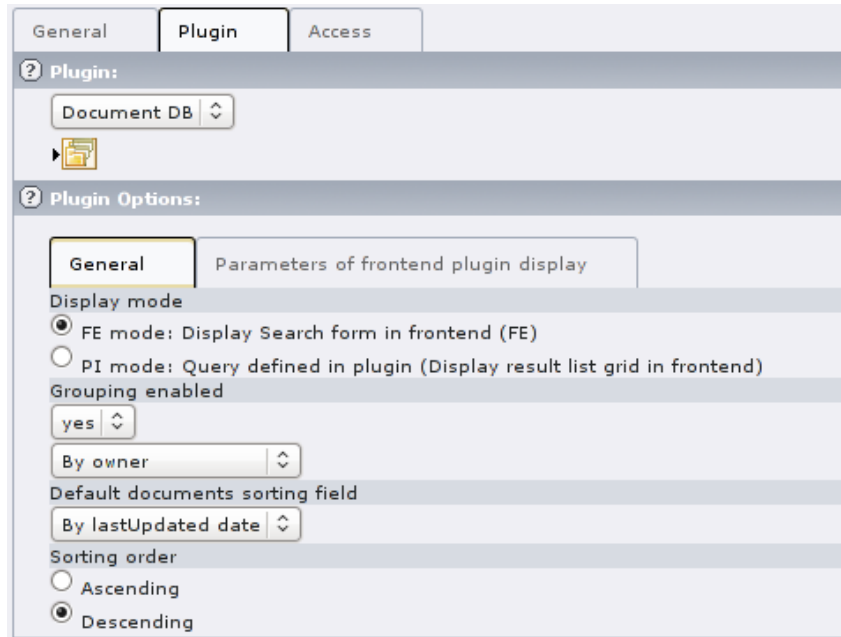
Add in the html template for all pages of type "document".

Otherwise the whole page is displayed in the popup:

<div id="wrap-content-id">

here the content to be displayed in the popup (modal window)

</ div>

Then fill in the name of the id wrapped in FLEXFORM extension. the CSS ID field. with "main-content" value in the current screenshot.

Setting up plugin as search form

There is only one parameter and two sub-parameters to configure the search form:

- Gouping enabled: Displays a select to choose wich field are used by default to show the result view grouped by that field.

- Default documents sorting: Set the default sorting order in list view.

Setting up plugin as pre-defined searches

Predefined searches displays only the list result grid in frontend.

Here, to select a list of descriptors, the wizard allows to search only those they are related to existing document.

Configuration TS

TypoScript Constants

Only few properties are available and can be defined

Property:	Data type:	Description:	Default:
path	string	ExtJS include path: include ExtJS library and resources	EXT:doc_db/resources/ExtJS/3.0.3/
resourcesPath	string	Include ExtJS resources path, Can be used to specify an alternative resources path for css theme and images. Trailing slash required. The resources directory must contains the following structure ./css/ ./images/default/ ./images/themeName/ Show example in EXT:doc_db/resources/ExtJS/3.0.3/resources/	EXT:doc_db/resources/ExtJS/3.0.3/resources/
doNotLoadExtAllJS	boolean	Define to true if ext-base & ext-all.js are already available on your installation, ext-all-min.js from doc_db load only the neccessary widgets lib. But it does not lack many parts.	0
doNotLoadExtAllCSS	boolean	Define to true if ext-all.css are already available on your installation	0
production	Options / boolean	Concat and minified CSS and JS files are used	1
~~asyncLoading~~	~~Options / boolean~~	~~Asynchronous loading of CSS and JS files add defer async to true~~	actually disabled - not available
gridColsWidth	string - csv	Coma separated values in the next order: Title,Date,Owner,Key,Type,Status . (The first column(title) is defined to autoExpand [min:150,max:650])	250,60,130,35,130,60
dateF.row	Options / string	Date string format to display in the Grid result rows	30.12.10 (day.month.year short)
dateFCustom.row	string	Date string format to display in the Grid result rows.(see http://php.net/date for possible strings formatting the date). NOTICE: if specified the above predefined parameter(dateF.row) will be ignored.	
dateF.detail	Options / string	Date string format to display in the detail view in expanded row	30 december 2010 (day month year long)
dateFCustom.detail	string	Date string format to display in the detail view in expanded row.(see http://php.net/date for possible strings formatting the date). NOTICE: if specified the above predefined parameter (dateF.detail) will be ignored.	

[tsref:plugin.tx_docdb_pi1.extJS]

To Do List

- Rewrite a cross browser compatible js asyncLoading process

- Localize Data

- create a second plugin to provided access to data in typoscript to build alternatives views.

- Currently, the tree used by selector of descriptors use xajax to load data . It works, but in the near future, instead of that it would be nice to rewrite this class to use the ajax framework provided by typo3 4.2

- Add the possibility to define additionnals fields/properties

Known problems

None so far. If any, let you know.

Changelog

Version	Changes:
0.3.1	- Rename with locallang_ prefix "extjs.docdb.gridStandalone.xml" and "extjs.docdb.xml". now, they are translatable with the ll-xml be module. - Rewrite class.ext_update.php to avoid errors related to max_execution_time. - At the loading the page, the "grid page size" paramater was not correct. - Remove "By lastUpdated date" value of "Default documents sorting field" in flexform which appeared twice. - Rewrited grid result and remove detail view pane. - Moved doc details in row expanded view. - Added scoll into view for the current expanded row. - Added prev/next button in modal preview window to navigate in the current results of displayed list. - And others minors changes. - Debug ie search filter field, display content in modal window preview. - Add. some constants TS config. and flexform Parameters. - Improve selection of the descriptors search filter.
0.3.0	Allmost complete rewrite and first realease that use extJs in frontend.
0.2.3	Old version. It was the base version to the version 0.3.0