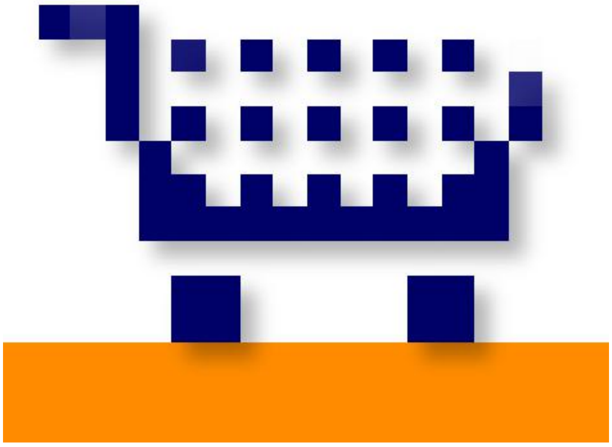
Quick Shop - Tutorial

This is the Tutorial for the Quick Shop. It should minimize lacks of understanding. And it should enables you to extend the Quick Shop with your own fields and tables. After eading you should be able to build your own shop like the Quick Shop. WARNING: This document isn't complete.



Version: 1.0.1 - 2010-07-01

Extension Key: quick_shop

Language: en

Keywords: forDevelopers, forAdvanced, quick shop

Table of Contents

Introduction

What should it do?

- Quick Shop - Tutorial should explain about how to develop your own web shop without any line PHP.
- Quick Shop - Tutorial should play back the developing of Quick Shop from the idea to the product (the extension).

What does it do?

We are sorry. We guess, that only advanced developers are able to follow our instructions.

And we suspect, that there are some parts and snippets missing.

Try it out and give us a feed back!

In case of lacks of understanding

There is another tutorial, which plays back the developing of a complex database with the browser. This tutorial should answer a lot of questions. The tutorial is in German language only.

If you have any lack of understanding, please take a look to

- Green Cars
  http://typo3.org/extensions/repository/view/green_cars/current/
- Green Cars Tutorial (PDF) – in German language
  http://typo3.org/extensions/repository/view/green_cars/current/info/?tx_terfe_pi1%5BdownloadFile%5D=doc%252Ffall_studie.pdf&cHash=8598ab3e74

What do you need?

- The extension Kickstarter (kickstarter).
- You need experience in TypoScript and you should know the TCA – the TYPO3 Configuration Array.

Audience

- TYPO3 developers

Tutorial I/II

You can build your own shop - without any PHP code. This tutorial plays back the developing of Quick Shop from the idea to the product (the extension).

The Idea

First we need the idea: How should the list view and the single view look like?

List and Single View

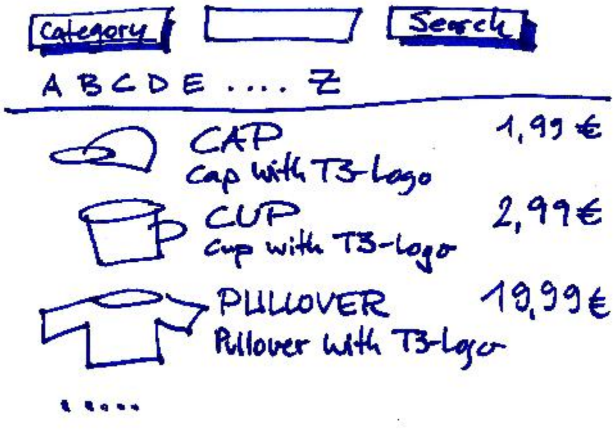


Illustration 1: Our small shop in the list view.

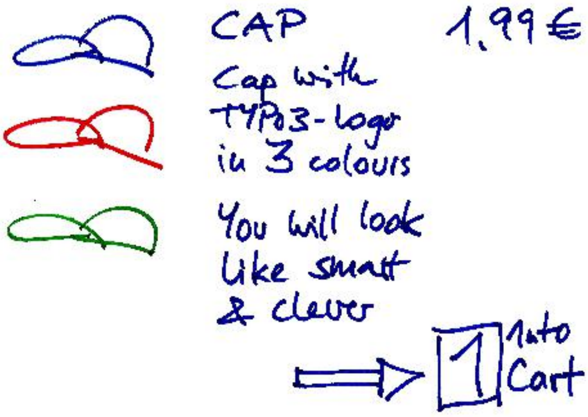

CAP
1,99€
Cap with TYPO3-logo in 3 colours
You will look like smart & clever
1
Into Cart

Illustration 2: Our small shop in the single view.

See our drafts for a small shop for TYPO3 merchandising: list view and single view.

We like to have a filter for categories, a search field and an index-browser or a so called A-Z-browser.

We have only to care about the filters. See the next steps.

The single view offers to order one or more of the selected product: we need a cart.

Database Model

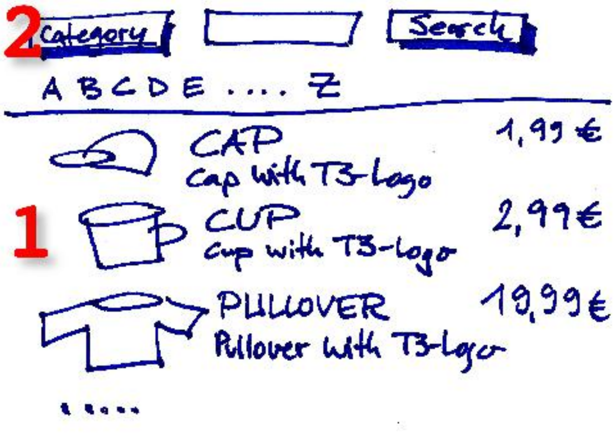


Illustration 3: List view with (1) category filter and (2) products list.

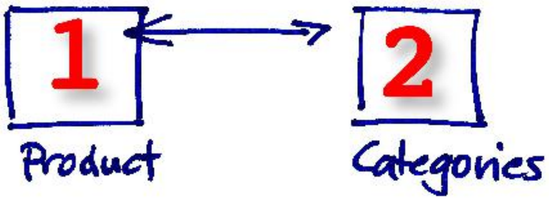
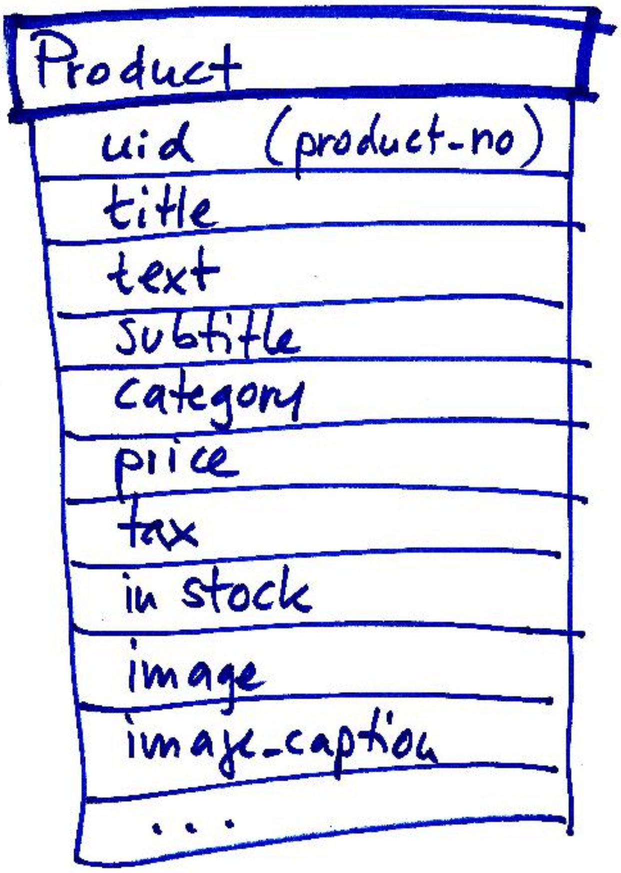


Illustration 4: Database with (2) products and (1) categories

We need two tables.

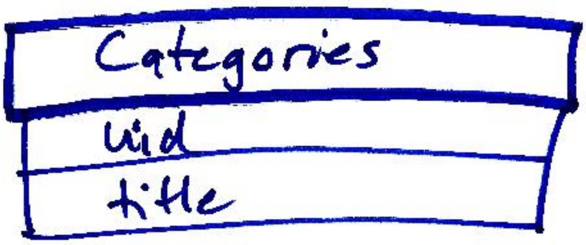
1. We like to display each product. We need a table products . Products will be our local table. It is the table with the main information. Database queries will need the local table in principle but not a foreign table.
2. We like to filter the products by their category. We need a table categories. Categories is our foreign table. If you like to filter products by their colour or size, you would need tables for sizes and colours too.

Tables



Product
uid (product-no)
title
text
subtitle
category
price
tax
in stock
image
image_caption
...

Illustration 5: The table product with its fields.



Categories
uid
title

Illustration 6: The table categories with its fields.

The product tables contains a lot of fields.

The category table is more simple.

TYPO3 will add some more fields to both tables.

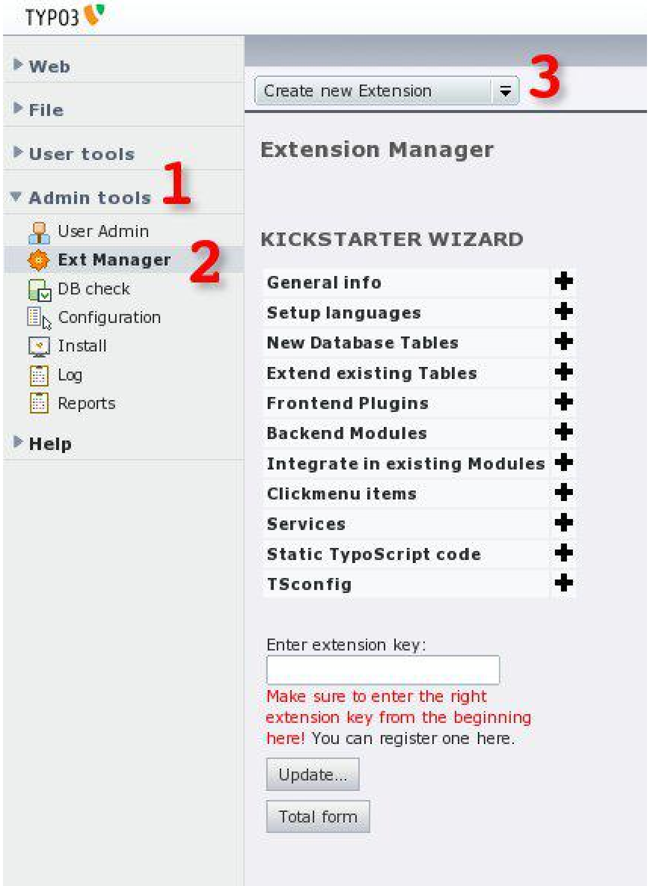
We will create the database with the TYPO3 kickstarter. See the next steps.

New Extension with the Kickstarter

If you don't have installed the kickstarter, please install it first. More about the kickstarter:

- http://typo3.org/extensions/repository/view/kickstarter/current/

Create a new Extension



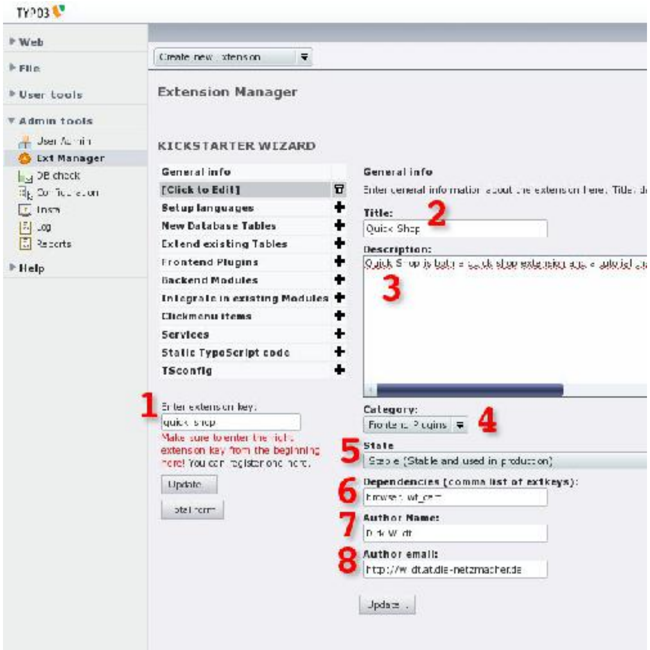
Make sure to enter the right extension key from the beginning here! You can register one here.

Illustration 7: Create a new extension.

1. Modul: Admin tools
2. Ext Manager
3. Create new Extension

General Info

Click on the plus icon of General info.



General info

Enter general information about the extension here: Title, d

Enter extension key: quick_shop

Make sure to enter the right extension key from the beginning here! You can register one here.

Title: Quick Shop

Description: Quick Shop is based on quick shop extension ...

Category: Frontend Plugins

State: Stable (Stable and used in production)

Dependencies (comma list of extkeys): browser, wt_cart

Author Name: Dirk Wildt

Author email: http://wildt.at.die-netzmacher.de

Illustration 8: General Info

1. Your extension key. If you don't want to publish your extension, please take the prefix my_. If you like to publish the extension, please reserve the extension key on typo3.org first.
2. Title of your extension.
3. Description of your extension.
4. Category should be [Frontend Plugin]
5. State: The state is stable because of the missing PHP code. The extension won't have any error or bug.
6. Dependencies: You need the extension browser and the extension wt_cart.
7. Author Name: Your name.
8. Author email: Be aware of spam! If you publish the extension on typo3.org, the code will be published too. Your e-mail will be published in clear text in a lot of files. We recommend to use a link to your website instead of an e-mail address.

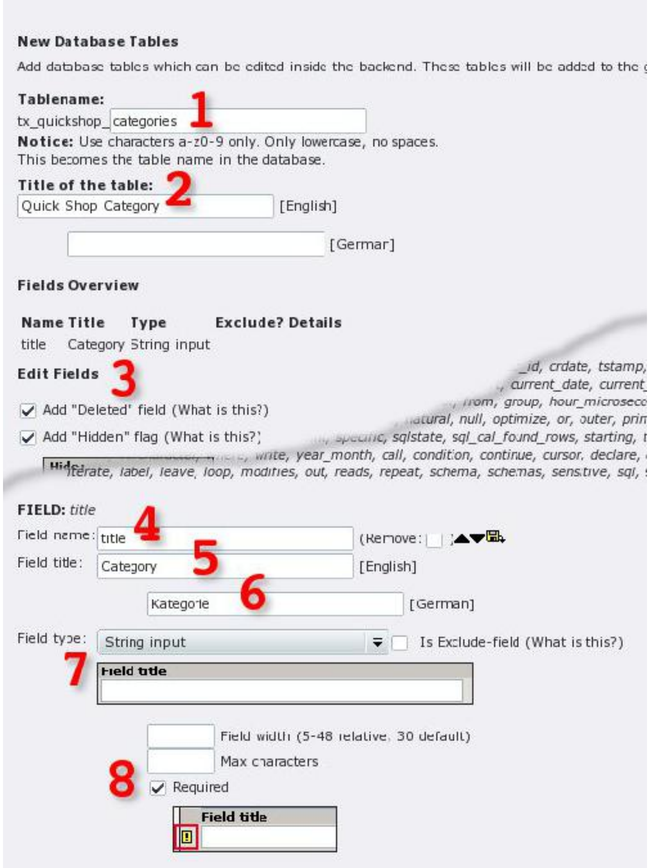
Setup Language

If you like localized backend forms and if your language isn't English, please set up the language. We selected German for the Quick Shop.

New Database Tables

It is recommended to start with all foreign tables first. Otherwise you have to administrate the local table twice.

Categories



New Database Tables

Add database tables which can be edited inside the backend. These tables will be added to the

Tablename:
tx_quickshop_ categories
Notice: Use characters a-z0-9 only. Only lowercase, no spaces.
This becomes the table name in the database.

Title of the table:
Quick Shop Category [English]
[German]

Fields Overview

Name	Title	Type	Exclude?	Details
title	Category	String input		

Edit Fields

☑ Add "Deleted" field (What is this?)
☑ Add "Hidden" flag (What is this?)

_id, crdate, tstamp,
current_date, current_
from, group, hour_microseco
natural, null, optimize, or, outer, prin
specific, sqlstate, sql_cal_found_rows, starting,
write, year_month, call, condition, continue, cursor, declare,
iterate, label, leave, loop, modifies, out, reads, repeat, schema, schemas, sensitive, sql,

FIELD: *title*
Field name: title (Remove: □)
Field title: Category [English]
Kategorie [German]
Field type: String input □ Is Exclude-field (What is this?)

Field width (5-48 relative, 30 default)
Max characters
☑ Required

Illustration 9: General Info

1. Name of the table: categories. The kickstarter prefix the name with tx_ and the extension key.
   Here: tx_quickshop_ ...
2. Title of the table. It is recommended to title the table with the extension name and than the table name.
   Here: Qucik Shop Category
3. Edit Fields. Enable
   [x] Add "Deleted" field
   [X] Add "Hidden" flag
4. Add your first and own field. The title is the field title in the database. If a field is called title, the browser extension will use this field for headers automatically.
   Call it title.
5. The name of the filed, which is displayed in backend forms. The browser use this name in the frontend too.
6. The name for the second language. The browser is localized and will use this name, if the delivered its content in this language (here: German)
7. Field type: Select [String input].

[Update ...] the form

8. The field must be required.

[Update ...] the form

- If "Manual ordering" is not set, order the table by this field: [title: Category]
- Label-field: [title]

Product

- Table name: product (-> tx_quickshop_product)
- Title of the table: Quick Shop Product [English], Quick Shop Produkt [German]
- Edit Fields
  - [X] Add "Deleted" field
  - [X] Add "Hidden" flag

Now we will add fields. After changing the field type you have to update the form. Than you will see the properties of the field type.

- Field name: title[1]
  - Field title: Product [English], Produkt [German]
  - Field type: [String input]
  - [X] Required
- Field name: short
  - Field title: Short [English], Kurzbeschreibung [German]

[1] If a field is called title, the extension browser tries to render it as a header. You can administrate it in the browser plugin > tab [TCA]

    - Field type: [Text area]
- Field name: description
    - Field title: Description [English], Beschreibung [German]
    - Field type: [Text area with RTE]
    - Rich Text Editor Mode: [Typical basic setup (new "Bodytext" field based on CSS stylesheets)]
    - [X] Fullscreen link
    - [X] Storage of images in separate folder (in uploads/[extfolder]/rte/)
- Field name: category
    - Field title: Category [English], Kategorie [German]
    - Field type: [Database relation]
    - Create relation to table: [Extension table: Quick Shop Category (tx_quickshops_categories)]
    - Type: [Selectorbox (from current page)]
    - [10] Max number of relations
    - [10] Size of selector box
    - [X] True M-M relations (otherwise commalist of values)
    - [X] Add "Add record" link
    - [X] Add "List records" link
    - [X] Add "Edit record" link
- Field name: price
    - Field title: Price [English], Preis [German]
    - Field type: [String input, advanced]
    - [X] Required
    - Evaluate value to: [floating point, x.xx]
    - [X] Strip space
    - (X) Not unique (default)
- Field name: tax
    - Field title: Tax included [English], inkl. MwSt. [German]
    - Field type: [String input, advanced]
    - [X] Required
    - Evaluate value to: [integer]
    - [X] Strip space
    - (X) Not unique (default)
- Field name: in_stock
    - Field title: In Stock [English], Auf Lager [German]
    - Field type: [Checkbox, single]
    - [X] Checked by default
- Field name: image[1]
    - Field title: Image [English], Bild [German]
    - Field type: [files]
    - Extensions: [Imagefiles]
    - [10] Max number of files

[1] If a field is called image, the extension browser tries to render it as an image. You can administrate it in the browser plugin > tab [TCA]

    - [1000] Max filesize allowed (kb)
    - [10] Size of selector box
    - [X] Show thumbnails
- Field name: caption[1]
    - Field title: Caption [English], Bildunterschrift [German]
    - Field type: [text area]
- Field name: imagealttext[2]
    - Field title: Alternate Text for Search Engines [English], Bild: alternativer Text für Suchmaschinen [German]
    - Field type: [text area]
- Field name: imagetitletext[3]
    - Field title: Title Text for Search Engines [English], Bild: Titel Text für Suchmaschinen [German]
    - Field type: [text area]

[Update ...] the form

- If "Manual ordering" is not set, order the table by this field: [title: Product]
- Label-field: [title]

Frontend Plugins

- Enter a title for the plugin: Quick Shop [English], Quick Shop [German]
- (X) Add to 'Insert Plugin' list in Content Elements'

[1] If a field is called caption, the extension browser tries to render it as the image caption.
You can administrate it in the browser plugin > tab [TCA]

[2] If a field is called imagealttext, the extension browser tries to render it as the image alt text.
You can administrate it in the browser plugin > tab [TCA]

[3] If a field is called imagetitletext, the extension browser tries to render it as the image title text.
You can administrate it in the browser plugin > tab [TCA]

Static TypoScript code

- Enter a title for the static extension template: quick_shop
- Setup[1]:

```
1.     plugin.tx_browser_pi1 {
2.       views {
3.         list {
4.           1 {
5.             select (
6.               tx_quickshop_products.title,
7.               tx_quickshop_products.short,
8.               tx_quickshop_products.image,
9.               tx_quickshop_products.imagealttext,
10.              tx_quickshop_products.imagetitletext,
11.              tx_quickshop_products.price
12.            )
13.          }
14.        }
15.        single {
16.          1 {
17.            select (
18.              tx_quickshop_products.title,
19.              tx_quickshop_products.description,
20.              tx_quickshop_products.image,
21.              tx_quickshop_products.imagealttext,
22.              tx_quickshop_products.imagetitletext,
23.              tx_quickshop_products.caption,
24.              tx_quickshop_products.tax,
25.              tx_quickshop_products.price,
26.              tx_quickshop_categories.title
27.            )
28.          }
29.        }
30.      }
31.    }
```

Comment

This setup creates two views: a list view (4) and a single view (16).

The list view will display the title of the product (6), the product note (7), the image (8) and the price (11). The image will get an image alt text (9) and an image title text (10).

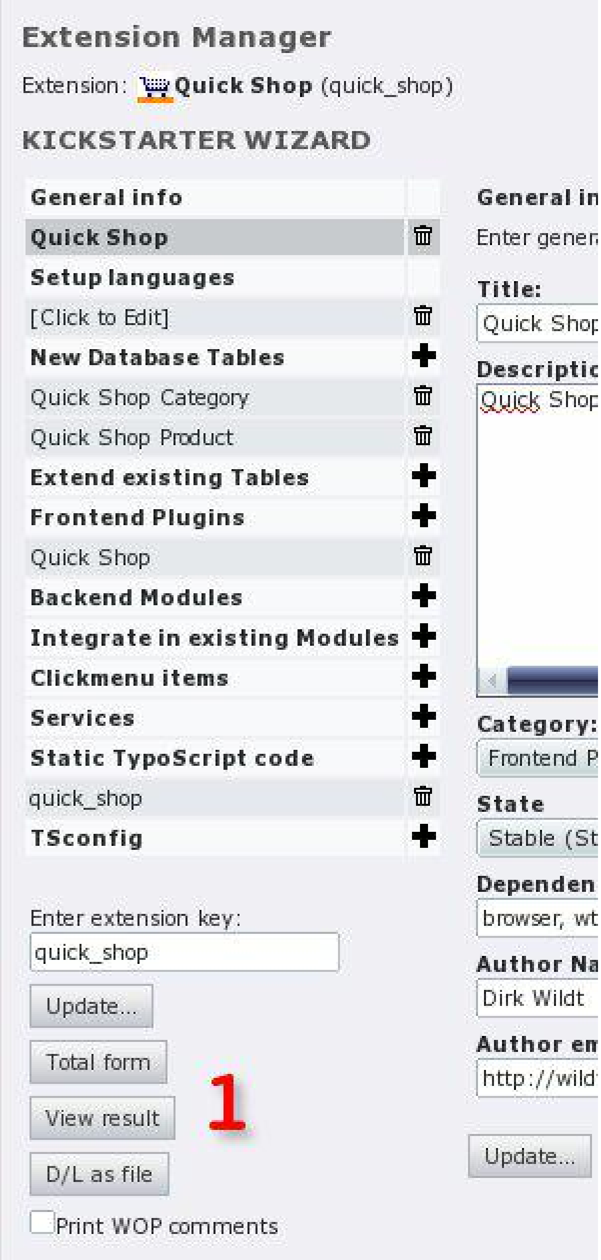
The single view will display some more information: description (19), tax (24) and the category title (26).

You don't have to care about the database relation, because the browser plugin will care about it. You don't have to be care about labelling, because the browser plugin will label all values in the frontend like TYPO3 in the backend forms.

We will enhance the TypoScript in a second step later.

[1] This is the minimum TypoScript code for displaying products in the frontend.

Publish your Extension



Extension Manager

Extension: Quick Shop (quick_shop)

KICKSTARTER WIZARD

Enter extension key: quick_shop

□ Print WOP comments

General in

Enter gener

Title: Quick Shop

Descriptic: Quick Shop

Category: Frontend P

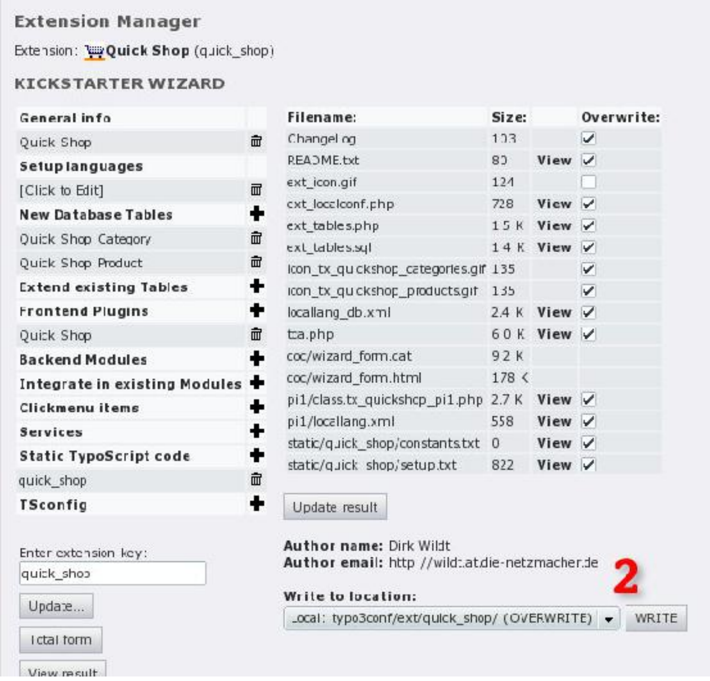
State: Stable (St

Dependen: browser, wt

Author Na: Dirk Wildt

Author em: http://wild

Illustration 10: [View result]



Extension Manager

Extension: Quick Shop (quick_shop)

KICKSTARTER WIZARD

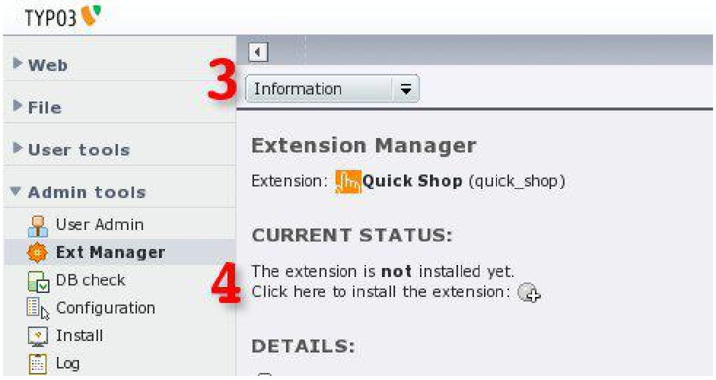
Filename:	Size:		Overwrite:
ChangeLog	133		☑
README.txt	83	View	☑
ext_icon.gif	124		□
ext_localconf.php	728	View	☑
ext_tables.php	1.5 K	View	☑
ext_tables.sql	1.4 K	View	☑
icon_tx_quickshop_categories.gif	135		☑
icon_tx_quickshop_products.gif	135		☑
locallang_db.xml	2.4 K	View	☑
tca.php	6.0 K	View	☑
doc/wizard_form.dat	9.2 K		
doc/wizard_form.html	178 K		
pi1/class.tx_quickshop_pi1.php	2.7 K	View	☑
pi1/locallang.xml	558	View	☑
static/quick_shop/constants.txt	0	View	☑
static/quick_shop/setup.txt	822	View	☑

Enter extension key: quick_shop

Author name: Dirk Wildt
Author email: http://wildt.at.die-netzmacher.de

Write to location: Local: typo3conf/ext/quick_shop/ (OVERWRITE)

Illustration 11: [WRITE]



Extension Manager

Extension: Quick Shop (quick_shop)

CURRENT STATUS:

The extension is **not** installed yet.
Click here to install the extension:

DETAILS:

Illustration 12: Install the extension

1. Click [View result]
2. Click [WRITE]
3. Extension Manager [Information]
4. Click on the phrase [Click here to install the extension]

Insert some records

Everything is ready for inserting some records.

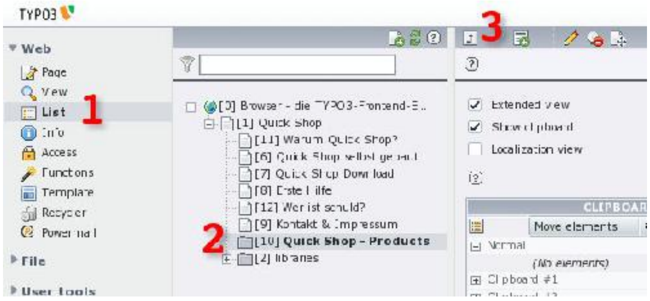


Illustration 13: Sysfolder Quick Shop

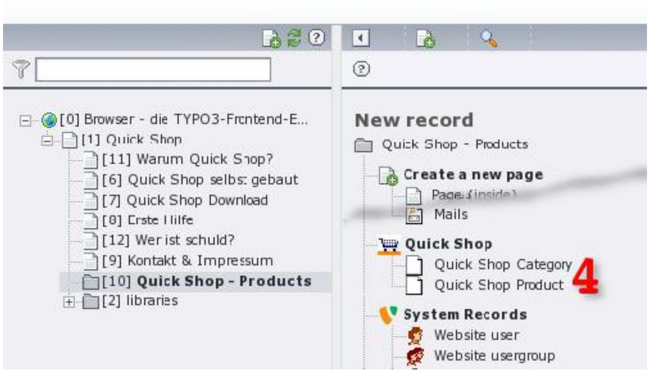


Illustration 14: Add your first record

1. Modul: Web > List
2. Page tree: Insert a sysfolder for storing your products. Call it "Quick Shop -Products". It has the uid 10 in our example above.
3. Edit area:click on the "add record" icon.
4. Click on a Quick Shop Category

Categories

In our example we will add this categories:

- Clothes
- Dishes
- Books

Products

In our example we will add this products:

- 1 cap (green)
- 1 cap (blue)
- 1 cap (red)
- 1 cup
- 1 pullover
- 1 book

Result

1. Table Categories with three records.
2. Table Products with six records.

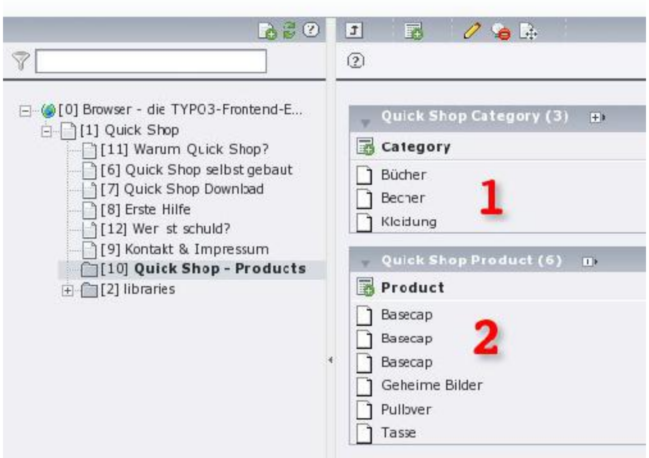


Illustration 15: General Info

Backend

It's time to see the result in the frontend.

You need

- a page "Quick Shop" with
- the browser plugin and
- a TypoScript template

If you don't have installed the extension browser, please install it now. See more about the browser:

- http://typo3.org/extensions/repository/view/browser/current/

Page

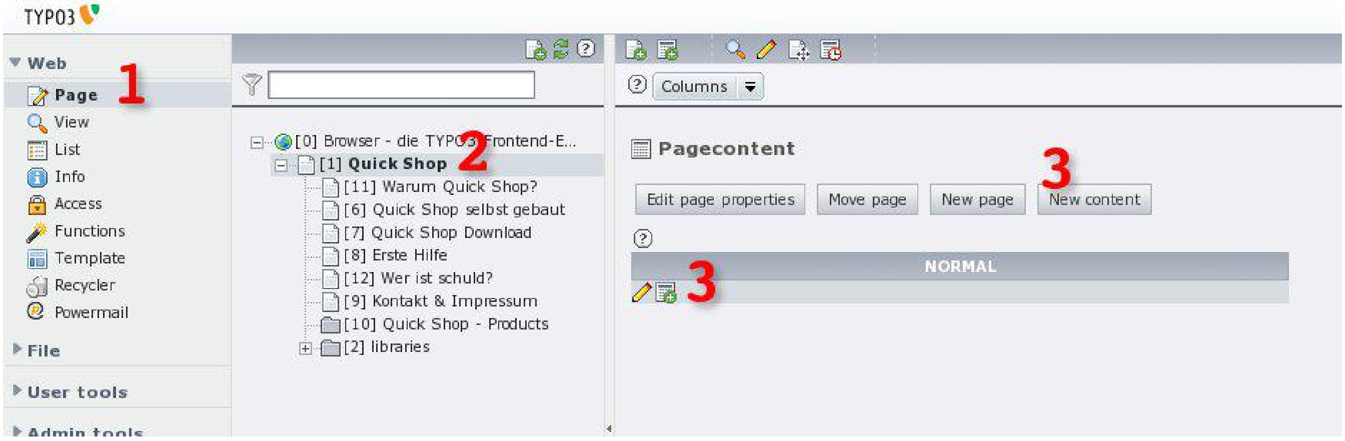


Illustration 16: Page "Quick Shop" for the Browserplugin

1. Modul: Web > Page
2. Page tree: Add a page "Quick Shop" - in our example above we have added the page as the first page of the project (uid 1).
3. Edit area: Add a new content element by clicking the button [New content] or the "add content" icon.

Browser Plugin

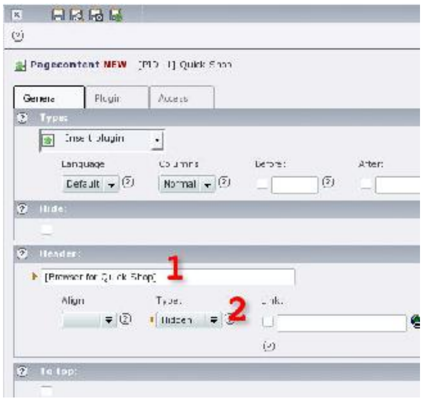


Illustration 17: Browser plugin - title

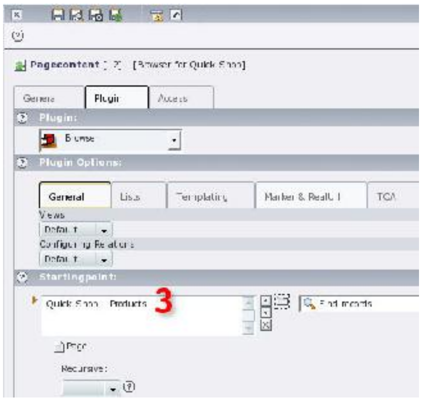


Illustration 18: Plugin - Startingpoint

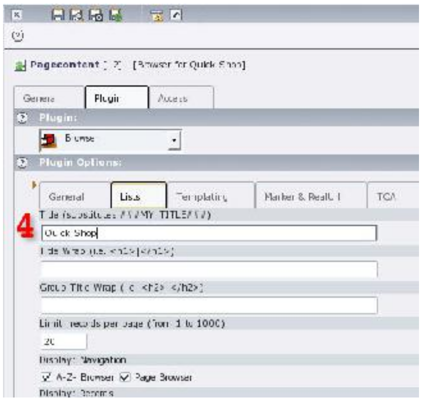


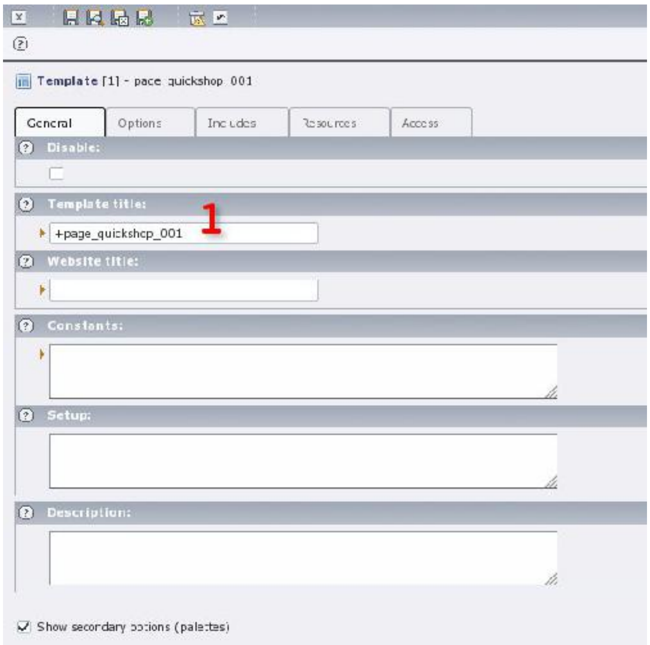
Illustration 19: Plugin - title

1. Tab [General] > field header: Call the plugin [Browser for Quick Shop]
2. Tab [General] > field header type: [hidden]
3. Tab [Plugin] > tab [General] > field startingpoint: sysfolder "Quick Shop - Products"
4. Tab [Plugin] > tab [Lists] > field title: Quick Shop

Template

The page with the Browser plugin needs to templates included for a proper workflow:

- the Browser template (with Browser default configuration)
- the Template from your extension (with your configuration)



Template [1] - page quickshop 001

Disable: □

Template title: +page_quickshop_001

Website title:

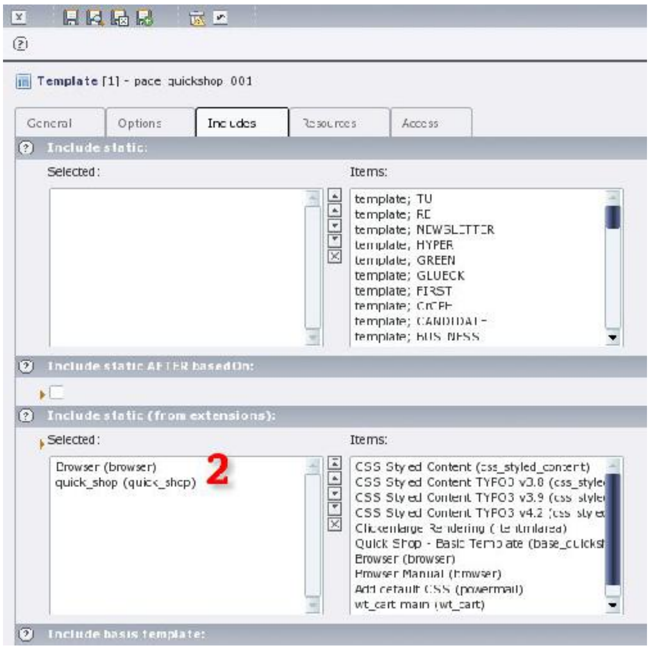
Constants:

Setup:

Description:

☑ Show secondary options (palettes)

Illustration 20: Adding an extension template



Template [1] - page quickshop 001

Include static:

Selected:

Items:
template; TU
template; RE
template; NEWSLETTER
template; HYPER
template; GREEN
template; GLUECK
template; FIRST
template; CrEPE
template; CANDIDATE
template; BUSINESS

Include static AFTER basedOn: □

Include static (from extensions):

Selected:
Browser (browser)
quick_shop (quick_shop)

Items:
CSS Styled Content (css_styled_content)
CSS Styled Content TYPO3 v3.8 (css_style
CSS Styled Content TYPO3 v3.9 (css_style
CSS Styled Content TYPO3 v4.2 (css_styled
Clickenlarge Rendering (clickenlarge)
Quick Shop - Basic Template (base_quicksh
Browser (browser)
Browser Manual (browser)
Add default CSS (powermail)
wt_cart main (wt_cart)

Include basis template:

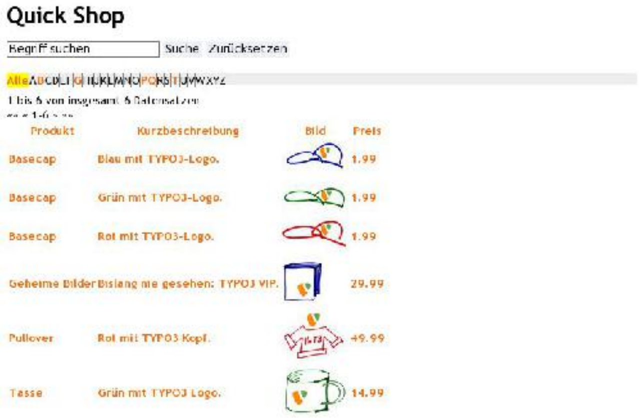
Illustration 21: Include static templates from extensions

1. If you don't have any template on the page with the Browser plugin, please add a new extension template. Call it in the format type + name of the current page + id of the typoscript template. Here: +page_quickshop_003
2. Include static templates in the existing template of your page or in the new template, which you have added in (1). Field Include static (from extensions):
   - Browser (browser)
   - quick_shop (quick_shop) <- the template of your extension

Please clear all TYPO3 cache.

Frontend

List View and Single View



Quick Shop

Begriff suchen | Suche | Zurücksetzen

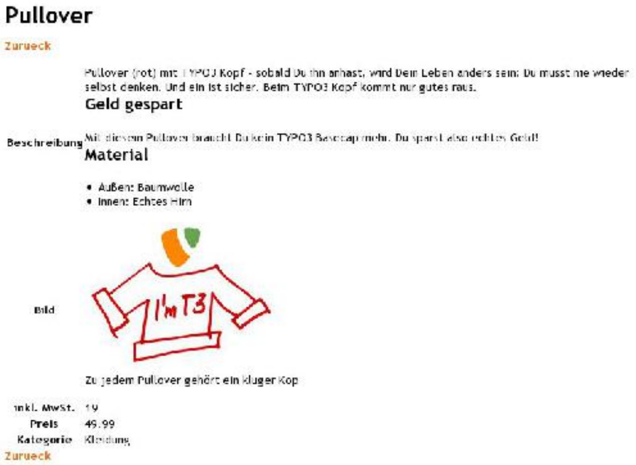
Alle A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

1 bis 6 von insgesamt 6 Datensätzen

<< < 1-6 > >>

Produkt	Kurzbeschreibung	Bild	Preis
Basecap	Blau mit TYPO3-Logo.		1.99
Basecap	Grün mit TYPO3-Logo.		1.99
Basecap	Rot mit TYPO3-Logo.		1.99
Geheime Bilder	Bislang nie gesehen: TYPO3 VIP.		29.99
Pullover	Rot mit TYPO3 Kopf.		49.99
Tasse	Grün mit TYPO3 Logo.		14.99

Illustration 22: List view



Pullover

Zurueck

Beschreibung: Pullover (rot) mit TYPO3 Kopf - sobald Du ihn anhast, wird Dein Leben anders sein: Du musst nie wieder selbst denken. Und ein ist sicher. Beim TYPO3 Kopf kommt nur gutes raus.

Geld gespart

Mit diesem Pullover brauchst Du kein TYPO3 Basecap mehr. Du sparst also echtes Geld!

Material

- Außen: Baumwolle
- Innen: Echtes Hirn

Bild: Zu jedem Pullover gehört ein kluger Kop

inkl. MwSt. 19

Preis 49.99

Kategorie Kleidung

Zurueck

Illustration 23: Single view

The result depends on the records , you have inserted, and of your CSS style sheets.

Be aware, what you have doing for this result:

- You have registered your database in TYPO3 - supported by the kickstarter.
- You have wrote two lines TypoScript

TypoScript

This is the code from above. Select property is in one line (5) and (10).

```
1.     plugin.tx_browser_pi1 {
2.       views {
3.         list {
4.           1 {
5.             select = tx_quickshop_products.title, tx_quickshop_products.short, ...
6.           }
7.         }
8.         single {
9.           1 {
10.            select = tx_quickshop_products.title, tx_quickshop_products.description, ...
11.          }
12.        }
13.      }
14.    }
```

Tutorial II/II

Now we like to have our own HTML template and a filter for categories.

HTML Templating

In this tutorial we explain all you need, but not all, which is possible.

If you want to know more about HTML Templating with the Browser plugin, please read the Tutorial

- Browser Tutorial Basics
  http://typo3.org/extensions/repository/view/browser_tut_basics_en/current/

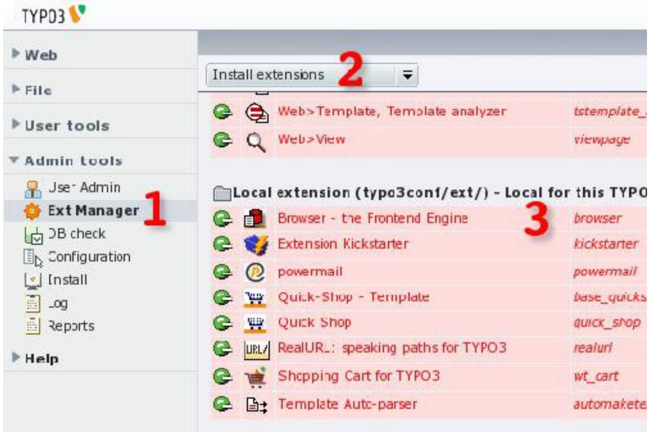
There is a section "Tutorial XII: HTML Templating".

Edit the default Template

Five steps:

1. Download the sample_ttnews template from the Browser extension
2. Edit the list view, add the marker for the filter
3. Edit the single view
4. Upload the template supported by the Browser plugin.

Download the default template

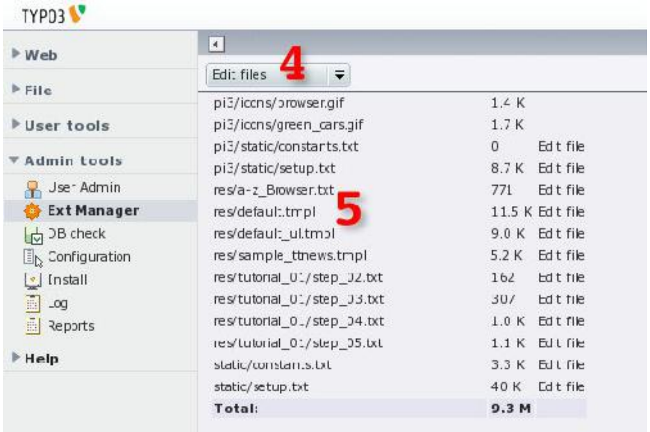


Web>Template, Template analyzer	tstemplate_
Web>View	viewpage

Local extension (typo3conf/ext/) - Local for this TYPO

Browser - the Frontend Engine	browser
Extension Kickstarter	kickstarter
powermail	powermail
Quick-Shop - Template	base_quicks
Quick Shop	quick_shop
RealURL: speaking paths for TYPO3	realurl
Shopping Cart for TYPO3	wt_cart
Template Auto-parser	automakete

Illustration 24: Extension Manager with Extension Browser



pi3/icons/browser.gif	1.4 K	
pi3/icons/green_cars.gif	1.7 K	
pi3/static/constants.txt	0	Edit file
pi3/static/setup.txt	8.7 K	Edit file
res/a-z_Browser.txt	771	Edit file
res/default.tmpl	11.5 K	Edit file
res/default_ul.tmpl	9.0 K	Edit file
res/sample_ttnews.tmpl	5.2 K	Edit file
res/tutorial_01/step_02.txt	162	Edit file
res/tutorial_01/step_03.txt	307	Edit file
res/tutorial_01/step_04.txt	1.0 K	Edit file
res/tutorial_01/step_05.txt	1.1 K	Edit file
static/constants.txt	3.3 K	Edit file
static/setup.txt	40 K	Edit file
Total:	**9.3 M**	

Illustration 25: The Browser files

1. Modul: Admin tools > Ext Manager
2. Select: [Install extensions]
3. Select: Browser extension
4. Select: [Edit files]
5. Click on "res/sample_ttnews.tmpl" or on "Edit file" in the same line.

Edit the list view, add the marker for the filter

The list view is beginning and ending with the subpart marker ###TEMPLATE_LIST###.

```
<!-- ###TEMPLATE_LIST### begin -->
        ###MY_TITLE###
<!-- ###SEARCHFORM### begin -->
        <div class="searchbox searchbox-###VIEW###-###MODE###">
          <form action="###ACTION###" method="post" >
            <input type="hidden" name="no_cache" value="1" />
            <input class="sword" type="text" name="tx_browser_pi1[sword]" ...
            <input class="button" type="submit" value="###BUTTON###" />
            <input class="button" type="button" onclick="location='###MY_URL###'" ...
            <!-- Add filter with the ###TABLE.FIELD### syntax and configure your ...
</form>
```

The form contains the comment "<!-- Add filter with the ###TABLE.FIELD### ...". See line (10) above.

You have to insert a table.field marker[1] for your filter. We like to get a filter for the table categories and the field title. We have to add ###TX_QUICKSHOP_CATEGORIES.TITLE###

```
1.    <!-- ###TEMPLATE_LIST### begin -->
2.        ###MY_TITLE###
3.    <!-- ###SEARCHFORM### begin -->
4.        <div class="searchbox searchbox-###VIEW###-###MODE###">
5.          <form action="###ACTION###" method="post" >
6.            ###TX_QUICKSHOP_CATEGORIES.TITLE###
7.            <input type="hidden" name="no_cache" value="1" />
8.            <input class="sword" type="text" name="tx_browser_pi1[sword]" ...
9.            <input class="button" type="submit" value="###BUTTON###" />
10.           <input class="button" type="button" onclick="location='###MY_URL###'" ...
11.           <!-- Add filter with the ###TABLE.FIELD### syntax and configure your ...
12.   </form>
```

We add the marker in line (6). See the red line above.

Now we replace the tt_news marker with our markers in the body part. See section ###LISTBODY###

Code before editing

```
1.        <!-- ###LISTBODY### begin --><!-- ###LISTBODYITEM### begin -->
2.          <div>
3.            <div style="float:left;">
4.              ###TT_NEWS.IMAGE###
5.            </div>
6.            <div style="padding-left:9em;">
7.              <h2>###TT_NEWS.TITLE###</h2>
8.              <ul>
9.                <li>
10.                 ###TT_NEWS_CAT.TITLE###
11.               </li>
12.               <li>
13.                 ###TT_NEWS.SHORT###
14.               </li>
15.             </ul>
16.           </div>
17.           <div style="clear:both"></div>
18.         </div><!-- ###LISTBODYITEM### end --><!-- ###LISTBODY### end -->
```

Code after editing

```
1.    <!-- ###LISTBODY### begin --><!-- ###LISTBODYITEM### begin -->
2.          <div>
3.            <div style="float:left;">
4.              ###TX_QUICKSHOP_PRODUCTS.IMAGE###
5.            </div>
6.            <div style="padding-left:9em;">
7.              <h2>###TX_QUICKSHOP_PRODUCTS.TITLE###</h2>
8.              <ul>
9.                <li>
10.                 ###TX_QUICKSHOP_PRODUCTS.SHORT###
11.               </li>
12.               <li>
13.                 ###TX_QUICKSHOP_PRODUCTS.PRICE###
14.               </li>
15.             </ul>
16.           </div>
17.           <div style="clear:both"></div>
18.         </div><!-- ###LISTBODYITEM### end --><!-- ###LISTBODY### end -->
```

[1] The extension Browser can handle table.field markers. In HTML template you have to write it in upper case and wrapped with ###. I.e ###TT_NEWS.TITLE###. Every table.field marker can have a corresponding marker in TypoScript in lowercase like tt_news.title. See more in the manual of the extension Browser.

Edit the single view

The single view is wrapped by the subpart marker ###TEMPLATE_SINGLE###.

We replace the tt_news marker with our markers in the body part. See section ###LISTBODY###

Code before editing

```
<!-- ###SINGLEBODY### begin --><!-- ###SINGLEBODYROW### begin -->
<div>
  <div style="float:left;">
    ###TT_NEWS.IMAGE###
  </div>
  <div style="padding-left:9em;">
    <h2>###TT_NEWS.TITLE###</h2>
    <ul>
      <li>
        ###TT_NEWS_CAT.TITLE###
      </li>
      <li>
        ###TT_NEWS.SHORT###
      </li>
    </ul>
  </div>
  <div style="clear:both"></div>
</div><!-- ###SINGLEBODYROW### end --><!-- ###SINGLEBODY### end -->
```

Code after editing

```
          <!-- ###SINGLEBODY### begin --><!-- ###SINGLEBODYROW### begin -->
          <div>
            <div style="float:left;">
              ###TX_QUICKSHOP_PRODUCTS.IMAGE###
            </div>
            <div style="padding-left:9em;">
              <h2>###TX_QUICKSHOP_PRODUCTS.TITLE###</h2>
              <ul>
                <li>
                  ###TX_QUICKSHOP_CATEGORIES.TITLE###
                </li>
                <li>
                  ###TX_QUICKSHOP_PRODUCTS.DESCRIPTION###
                </li>
              </ul>
            </div>
            <div style="clear:both"></div>
          </div>
<!-- ###SINGLEBODYROW### end --><!-- ###SINGLEBODY### end -->
```

Connection to wt_cart and powermail

What does wt_cart do?

- wt_cart is a small but powerful extension which provides the shopping cart for quick shop. wt_cart itself neither allows you to create products, articles or categories nor includes a checkout functionality. The extension depends on powermail which helps you to integrate the checkout process. Furthermore powermail is utilized for sending the order confirmation.
- wt_cart comes with a very powerful dynamic marker functionality. With the help of TypoScript you can create individual markers. As soon as you've added a new marker it's available in your HTML template. Regarding quick_shop wt_cart and powermail are configured for an optimal usage with a proper set of markers / information.
- Technically wt_cart is an extension which utilizes the TYPO3 session. Build your own add-to-cart form and send information like color, size and quantity as post parameters. These post parameters will be added to the TYPO3 session. wt_cart handles the session data and – based on your setup – looks up a specific TYPO3 database table to present the product title and any other additional information. In the case of quick_shop wt_cart uses the quick_shop database tables. Right now there are no fields for color and size. But such fields will be added soon.

Administration

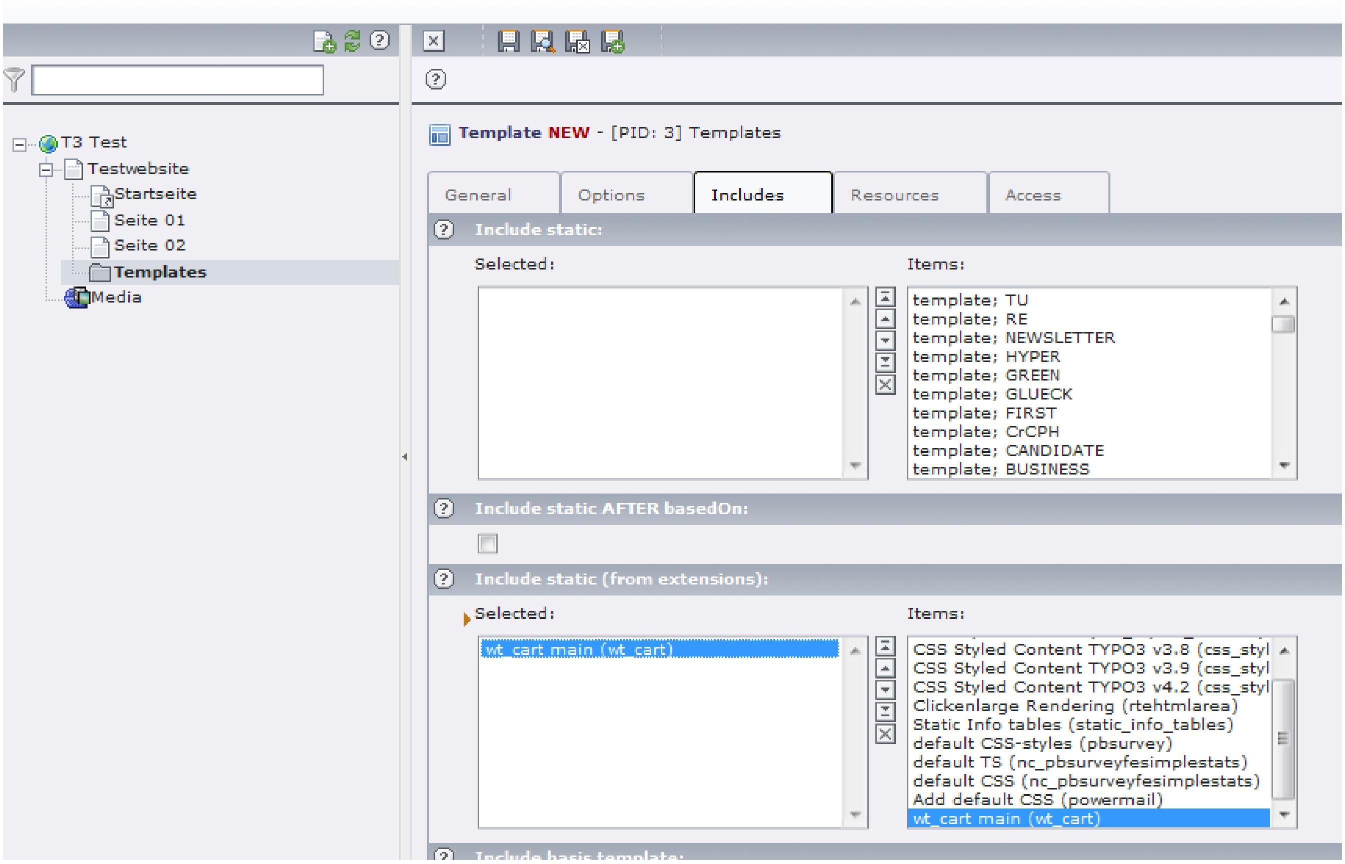
Install wt_cart with Extension Manager

Open the extension manager, download the extension "wt_cart" and install it. If you haven't already installed powermail you'll be asked to do it during the installation process of wt_cart.

There are no additional settings you can change within the extension manager regarding wt_cart.

Include Static Template

Create a new extension template (TypoScript) or edit an existing one and include the static template of wt_cart.



Template NEW - [PID: 3] Templates

Include static:

Selected:

Items:
template; TU
template; RE
template; NEWSLETTER
template; HYPER
template; GREEN
template; GLUECK
template; FIRST
template; CrCPH
template; CANDIDATE
template; BUSINESS

Include static AFTER basedOn:

□

Include static (from extensions):

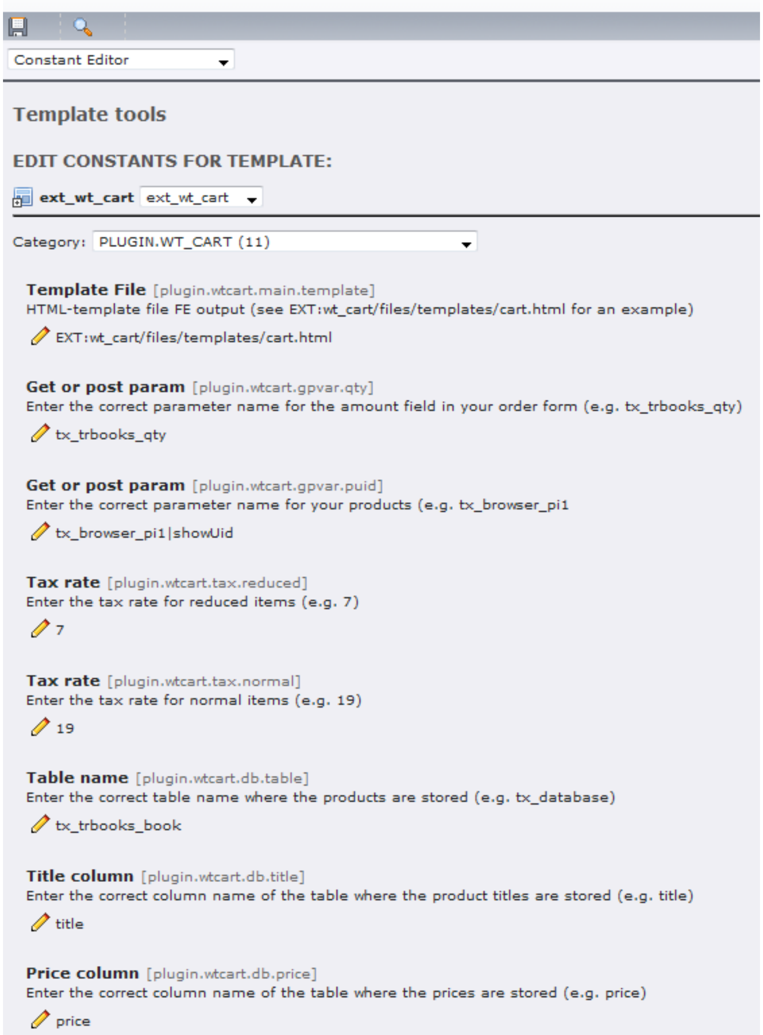
Selected:
wt_cart main (wt_cart)

Items:
CSS Styled Content TYPO3 v3.8 (css_styl
CSS Styled Content TYPO3 v3.9 (css_styl
CSS Styled Content TYPO3 v4.2 (css_styl
Clickenlarge Rendering (rtehtmlarea)
Static Info tables (static_info_tables)
default CSS-styles (pbsurvey)
default TS (nc_pbsurveyfesimplestats)
default CSS (nc_pbsurveyfesimplestats)
Add default CSS (powermail)
wt_cart main (wt_cart)

Include basis template:

Illustration 26: Include static template of wt_cart

TypoScript Constants



Template tools

EDIT CONSTANTS FOR TEMPLATE:

ext_wt_cart ext_wt_cart

Category: PLUGIN.WT_CART (11)

Template File [plugin.wtcart.main.template]
HTML-template file FE output (see EXT:wt_cart/files/templates/cart.html for an example)
EXT:wt_cart/files/templates/cart.html

Get or post param [plugin.wtcart.gpvar.qty]
Enter the correct parameter name for the amount field in your order form (e.g. tx_trbooks_qty)
tx_trbooks_qty

Get or post param [plugin.wtcart.gpvar.puid]
Enter the correct parameter name for your products (e.g. tx_browser_pi1
tx_browser_pi1|showUid

Tax rate [plugin.wtcart.tax.reduced]
Enter the tax rate for reduced items (e.g. 7)
7

Tax rate [plugin.wtcart.tax.normal]
Enter the tax rate for normal items (e.g. 19)
19

Table name [plugin.wtcart.db.table]
Enter the correct table name where the products are stored (e.g. tx_database)
tx_trbooks_book

Title column [plugin.wtcart.db.title]
Enter the correct column name of the table where the product titles are stored (e.g. title)
title

Price column [plugin.wtcart.db.price]
Enter the correct column name of the table where the prices are stored (e.g. price)
price

Illustration 27: Illustration 28: Editing TypoScript Constants with Constant Editor

In order to show products within wt_cart you have to configure some constants with the help of the Constants Editor. Please check the next section "Configuration" for more details about the specific settings.

Reference

TypoScript Constants

Possible subsections: Reference (TypoScript)

Property:	Data type:	Description:	Default:
main.template	File	Template File: HTML-template file FE output (see EXT:wt_cart/files/templates/cart.html for an example) You can specify the path to your own HTML template.	EXT:wt_cart/files/templates/cart.html
gpvar.qty	Text	Get or post param: Enter the correct parameter name for the amount field in your order form (e.g. tx_trbooks_qty)	tx_trbooks_qty
gpvar.puid	Text	Get or post param: Enter the correct parameter name for your products (e.g. tx_browser_pi1|showUid)	tx_browser_pi1|showUid
tax.reduced	Text	Tax rate: Enter the tax rate for reduced items (e.g. 7)	7
tax.normal	Text	Tax rate: Enter the tax rate for normal items (e.g. 19)	19
db.table	Text	Table name: Enter the correct table name where the products are stored (e.g. tx_database)	tx_trbooks_book
db.title	Text	Title column: Enter the correct column name of the table where the product titles are stored (e.g. title)	title
db.price	Text	Price column: Enter the correct column name of the table where the prices are stored (e.g. price)	price
db.tax	Text	Tax column: Enter the correct column name of the table where the tax categories are stored (e.g. tax)	tax
powermailContent.uid	Int+	Powermail content uid: Enter the uid of the powermail content element to hide this element if the cart is empty AND to clear cart if this form was submitted.	
debug	Boolean	Debug Mode: Enable or disable some debug output	0

Configuration

Create page with wt_cart plugin

Create a new page called "Cart" or "Order" or whatever you want and add a new content element. In order to add the wt_cart plugin choose as type "Insert plugin". On the tab "Plugin" select as plugin "Cart to powermail". That's it. No addiotional settings are available. Nice ;)

On the same page add after the wt_cart plugin a new powermail plugin and configure it. Create all the required fields and logic you need.

Configure powermail to work with wt_cart

Edit your powermail plugin and edit the email which is sent to the sender or recipient respectively. Add the following marker:

###POWERMAIL_TYPOSCRIPT_CART###

This marker includes the whole shopping cart information provided by wt_cart. After including the marker your powermail emails will contain the shopping cart data.

Helpful suggestions

If you have helpful suggestions, feel free to contact me: Dirk Wildt http://wildt.at.die-netzmacher.de.

To-Do List

- Complete the tutorial for developers, which aren't advanced.

Further Information

About the plugin icon

Illustration 29: The logo

The concept of the icon:

- The cylinder is the common symbol for a database.
- The sheet is a symbol for the frontend.
- The colour of the stripe, orange, is the one of the TYPO3 colours.

About the Authors

Björn Jacob

Illustration 30: Björn Jacob

Björn Jacob (* 1981 Ilmenau/Germany)

Founder of TRITUM (Jena/Gräfenroda)

Member of TYPO3 Usergroup Thuringia

Member of TYPO3 Usergroup Leipzig

since 2007: TRITUM is born and growing rapidly

2001 - 2007: University of applied sciences Jena – Business Administration

since 2000: Internet junky

Alexander Kellner

Illustration 31: Alexander Kellner

Dirk Wildt

Illustration 32: The Author - a little bit blurred

Dirk Wildt (* 1963 Hamburg/Germany)

Founder of Die Netzmacher GbR (Erfurt/Weimar)

Founder of TYPO3 Usergroup Thuringia

Member of TYPO3 Usergroup Leipzig

2003: Founding think visually! It is Die Netzmacher since 2009.

2000 - 2002: Financial Controller, G.E.B.B

1999 - 2000: Financial Controller, G|M|Z - Tagesspiegel-Gruppe

1997 - 1999: Spokesman, Senatsverwaltung für Finanzen, Berlin, Berlin

1989 - 1997: Editor and Financial Controller, taz, die tageszeitung

1982 - 1989: Freelancing Press Photographer

Tutorial in PDF

You find this tutorial in PDF format at

- doc/tutorial.pdf
  http://typo3.org/extensions/repository/view/quick_shop/current/info/?tx_terfe_pi1%5BdownloadFile%5D=doc%252Fturorial.pdf

Other extensions published by Die Netzmacher

- Browser for BZD Staff Directory. List view and single view for the extension BZD Staff Directory. It is out of the box. You can edit the views by TypoScript. You can add fields and whole views.
  http://typo3.org/extensions/repository/view/browser_bzdstaffdirectory/current/
- Browser for Virtual Civil Services - List views and single views for the extension Virtual Civil Services (civserv) . It supersedes smarty and enables SEO - search engine optimisation -, Real URL and the TYPO3 cache management. It is out of the box. It You can edit the views by TypoScript. You can add fields and whole views.
  http://typo3.org/extensions/repository/view/browser_civserv/current/
- Green Cars (Grüne Autos) - Database optimized for ecological cars. It is a case study and demonstrates, how to get a complex database with the Browser in three hours only.
  http://typo3.org/extensions/repository/view/green_cars/current/
- Juridat - Database for Juridical Data. Juridat provides a juridical data base with backend functionality. Juridat is out of the box. It is a teamwork with the Browser.
  http://typo3.org/extensions/repository/view/juridat/current/
- logical_form: A frontend plugin for evaluating forms. I.e you can evaluate mail addresses.
  http://typo3.org/extensions/repository/view/logical_form/current/
- Majordomo: For subcribing to and unsubscribing from a majordomo mailing list.
  http://typo3.org/extensions/repository/view/majordomo/current/
- rm_staticfile: Improve your performance! This extension enables the delete of static files, which where generated by nc_staticfilecache. It is a team play with the TYPO3 cache management, ttnews, ttnewscache, sg_newsplus and sg_zfelib.
  http://typo3.org/extensions/repository/view/delete_staticfile_by_3party/current/
- seo_dynamic_tag: Search Engine Optimisation. You can generate values dynamically with this extension especially for the <title>-tag, for the <meta>-tag description and the <meta>-tag keywords.
  http://typo3.org/extensions/repository/view/seo_dynamic_tag/current/
- tt_news Marker (extkey: ttnews_marker) enables to add any marker to a tt_news HTML template including links. Any marker has the std_wrap property.
  http://typo3.org/extensions/repository/view/ttnews_marker/current/
- tt_news select configuration (extkey: ttnews_selectconf) enables to select tt_news by any SQL clause. The extension adds an andWhere clause to the SQL query of the tt_news plugin. I. e. it is possible to display tt_news items in dependence on the ownership of a fe_user.
  http://typo3.org/extensions/repository/view/ttnews_selectconf/current/
- Wine Catalogue provides a data base for wine with regions, wineries, styles, variety and ageing among others. It is localized. English, German and Spanish ist of the box. Wine based on the extension Browser.
  http://typo3.org/extensions/repository/view/wine/current/

Change Log

1.0.1	**Initial release**

Illustration Index