

# SEO Dynamic Tag 2

Search Engine Optimization for the title tag, the canonical tag and the meta tags author, description and keywords.



Version: 2.0.4, 2013-09-18

Extension Key: seo\_dynamic\_tag

Language: en

Keywords: SEO, search engine optimization, tags, meta tags, meta, title, canonical, author, description, keywords

Copyright 2007 - 2013, Dirk Wildt, Die Netzmacher, <<http://wildt.at.die-netzmacher.de>>

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](http://www.typo3.org)

## Table of Contents

SEO Dynamic Tag 2 .....	1	Allow a parameter .....	32
Screenshots .....	3	Condition .....	33
Backend .....	3	Recommended pages .....	33
Frontend .....	4	Check it! .....	35
Introduction .....	5	Debug Mode .....	36
What does it do? .....	5	Options .....	36
Ready-for-use .....	5	SQL Errors .....	36
Requirements .....	5	Development .....	37
Forum .....	5	Reference .....	38
Installation .....	6	Canonical Tag .....	38
Quick start .....	6	MetaTag Author .....	39
Details .....	6	MetaTag Description .....	39
Tutorials .....	7	MetaTag Keywords .....	39
Calendar Base (cal) .....	7	Title .....	39
News (tt_news) .....	10	Query .....	40
Organiser (org) .....	13	What's new? .....	41
Shop System (tt_products) .....	16	In version 2.x .....	41
Any other database .....	19	Update .....	42
Values from foreign tables .....	26	From smaller than 2.x .....	42
User Interface – TypoScript Constant Editor .....	27	FAQ .....	43
Canonical .....	27	SimulateStaticDocuments and RealUrl .....	43
Condition .....	27	Helpful suggestions .....	44
Database .....	27	Forum .....	44
Debugging .....	28	Support with costs .....	44
Default .....	28	Further Information .....	45
Keywords .....	28	Other extensions published by Die Netzmacher .....	45
Page .....	29	Credits .....	46
Workflow .....	30	State Development Corporation of Thuringia .....	46
In principle .....	30	To-Do list .....	47
Schema .....	30	Changelog .....	48
Canonical Tag .....	31	Illustration Index .....	49
An unique page should have an unique URL .....	31		
The canonical tag of SEO Dynamic Tag .....	31		

# Screenshots

## Backend

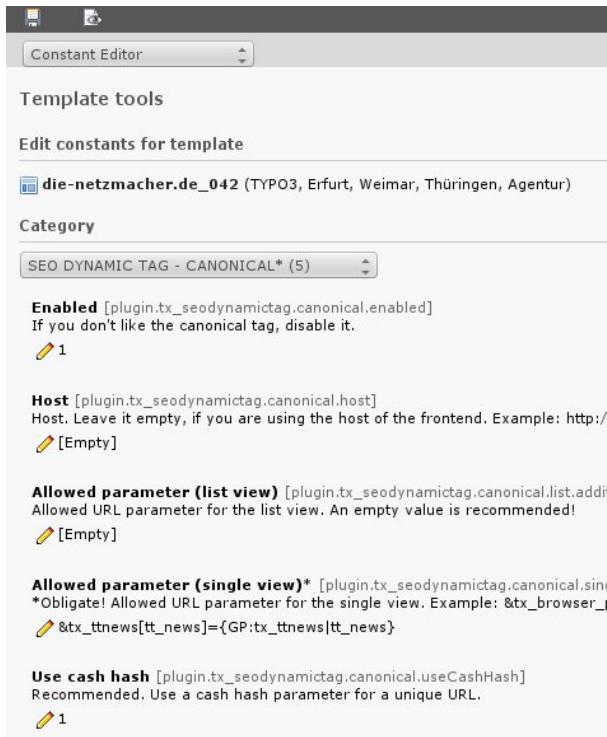


Illustration 1: User interface: Canonical Tag

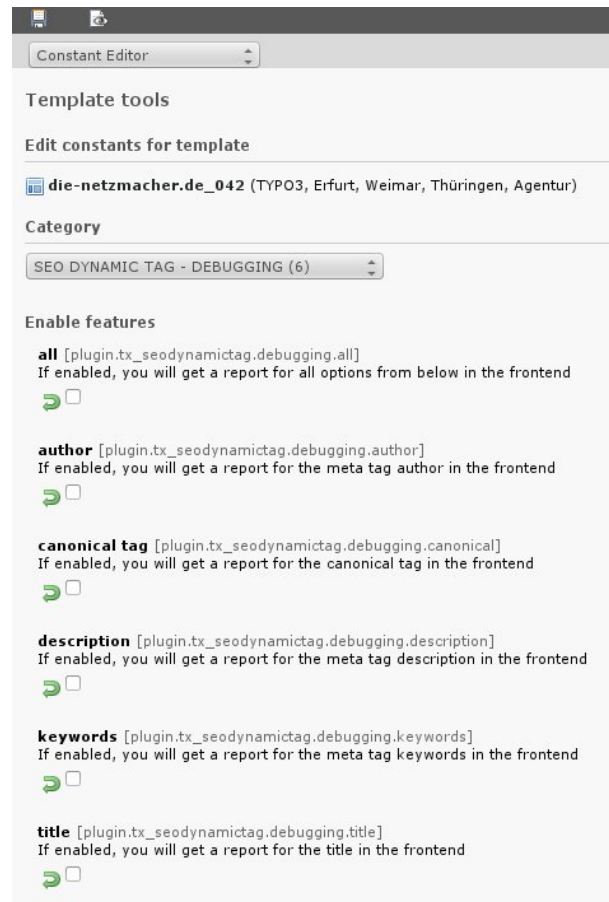


Illustration 2: User interface: Debugging

Two of seven forms of the user interface of SEO Dynamic Tag.

## Frontend

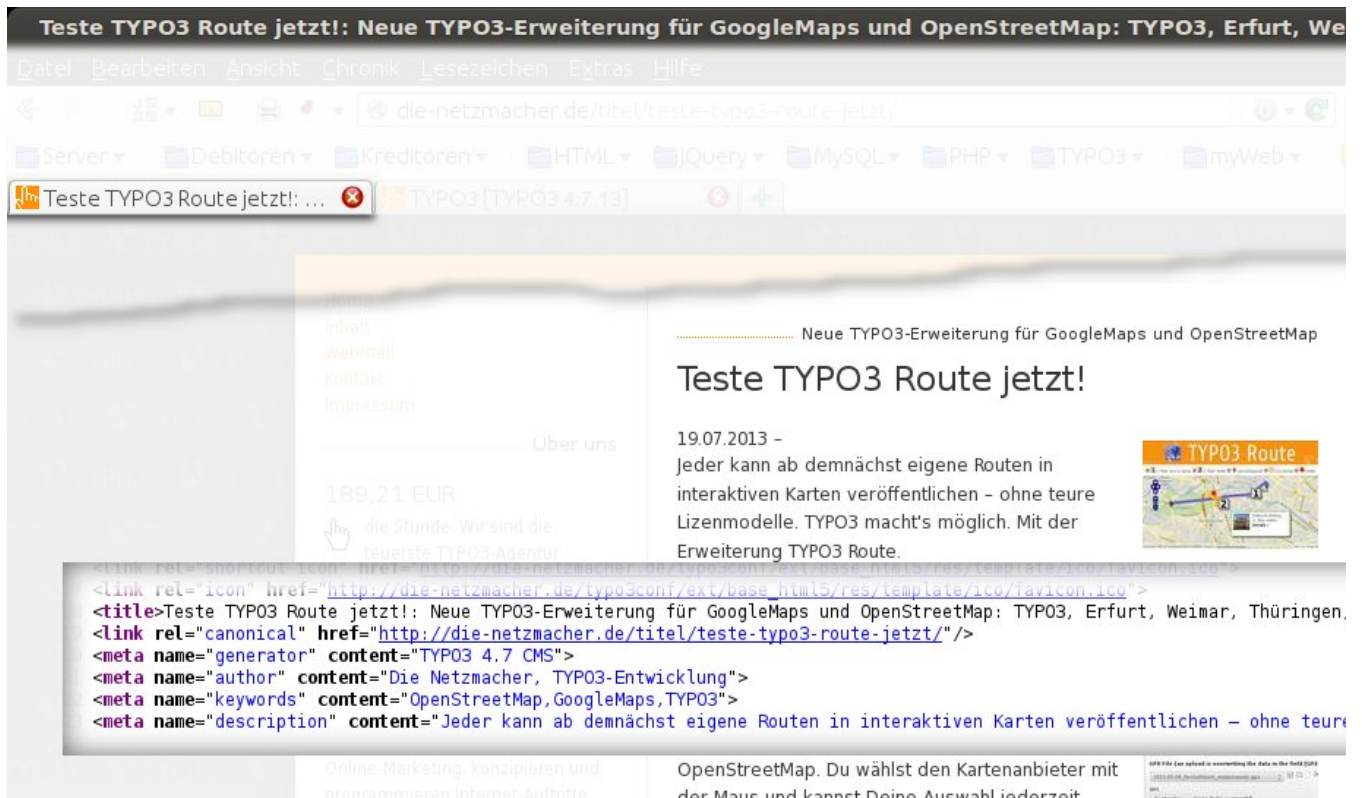


Illustration 3: SEO - Search Engine Optimization - with SEO Dynamic Tag

SEO Dynamic Tag generates the title tag, the canonical tag and the meta tags author, keywords and description.

See details at "Tutorials" on page 7 below.

# Introduction

## What does it do?

- SEO Dynamic Tag supports the Search Engine Optimization of your TYPO3 website.
- You can generate search HTML tags for records of every TYPO3 extension automatically for the
  - <title>-tag
  - <link rel="canonical">-tag
  - <meta name="author">-tag
  - <meta name="description">-tag
  - <meta name="keywords">-tag
- SEO Dynamic Tag has ready-for-use templates for
  - Calendar Base (cal)
  - News (tt\_news)
  - Shop System (tt\_products)

You have to configure the page id for the single view only.

- SEO Dynamic Tag has a user interface for the configuration.
- If you will have any need, you can configure SEO Dynamic Tag by TypoScript directly.

## Ready-for-use

- The Organiser – TYPO3 for the lobby and organisers – (org) has SEO Dynamic Tag integrated<sup>1</sup>. You can install the Organiser ready-for-use with one mouse click. See <http://typo3-organiser.de/>

## Requirements

- Any requirement.
- If you have special needs, experience with TypoScript is recommended.

## Forum

SEO Dynamic Tag 2 has a forum at:

- <http://typo3-browser-forum.de/#cat28>

Posts are welcome in English in German

<sup>1</sup> Upto version 3.3.1. I intend to publish the Organiser 3.3.1 from October, 2013.

# Installation

## Quick start

- Install the extension with the extension manager.
- Include the static template
  - SEO Dynamic Tag (seo\_dynamic\_tag)
- If you like to optimize the following extensions for search engines
  - Calendar Base (cal)
  - News (tt\_news)
  - Shop system (tt\_products)include one of the static templates
  - +SEO Dynamic Tag cal (seo\_dynamic\_tag)
  - +SEO Dynamic Tag tt\_news (seo\_dynamic\_tag)
  - +SEO Dynamic Tag tt\_products (seo\_dynamic\_tag)
- If you like to optimize the following extensions for search engines
  - Organiser (org)you don't need to include any further template.
- Configure SEO Dynamic Tag by the user interface.
  - In case of the ready-for-use templates and in case of the Organiser you need to configure the page id only.

## Details

- Calendar Base (cal) on page 7
- News (tt\_news) on page 10
- Organiser (org) on page 13
- Shop System (tt\_products) on page 16
- Any other database on page 19

# Tutorials

## Calendar Base (cal)

### Initial Situation

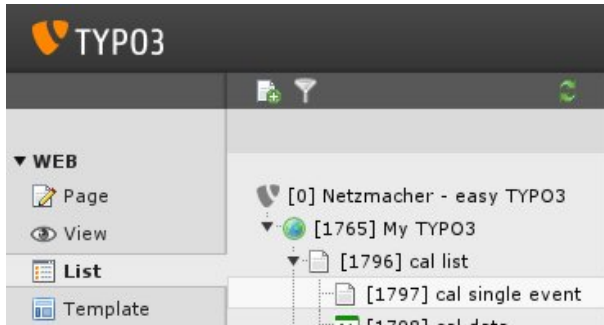


Illustration 4: Calendar Base (cal): initial situation

- I am using Calendar Base (cal) for the single view on page: [1797] cal single event

### Include the static templates

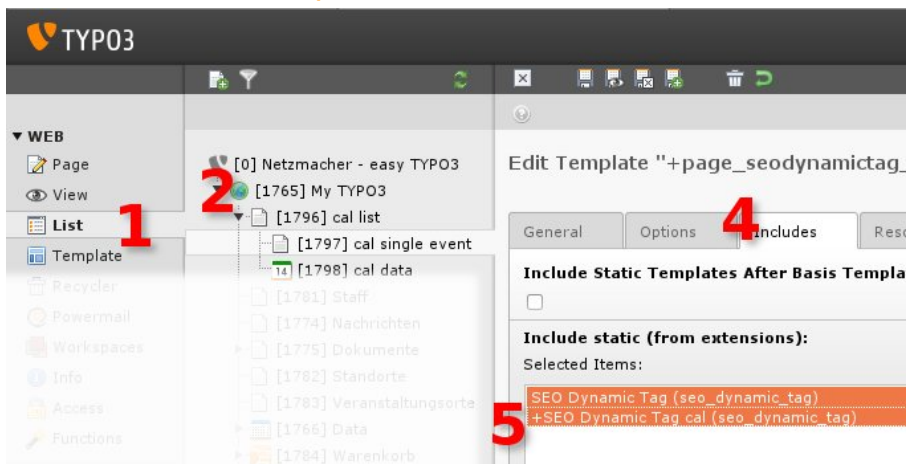


Illustration 5: Calendar Base (cal): include the static templates

Include the static templates:

- Web List
- Page tree Page with the Calendar Base plugin for the single view (here: [1797] cal single event)
- Edit area Select the TypoScript template<sup>1</sup> for editing (not illustrated)
- Tab [Includes]
- Add the static templates:  
SEO Dynamic Tag (seo\_dynamic\_tag)  
+SEO Dynamic Tag cal (seo\_dynamic\_tag)

<sup>1</sup> If you don't have a template on the current page, please add an extension template.

## Configure the page id

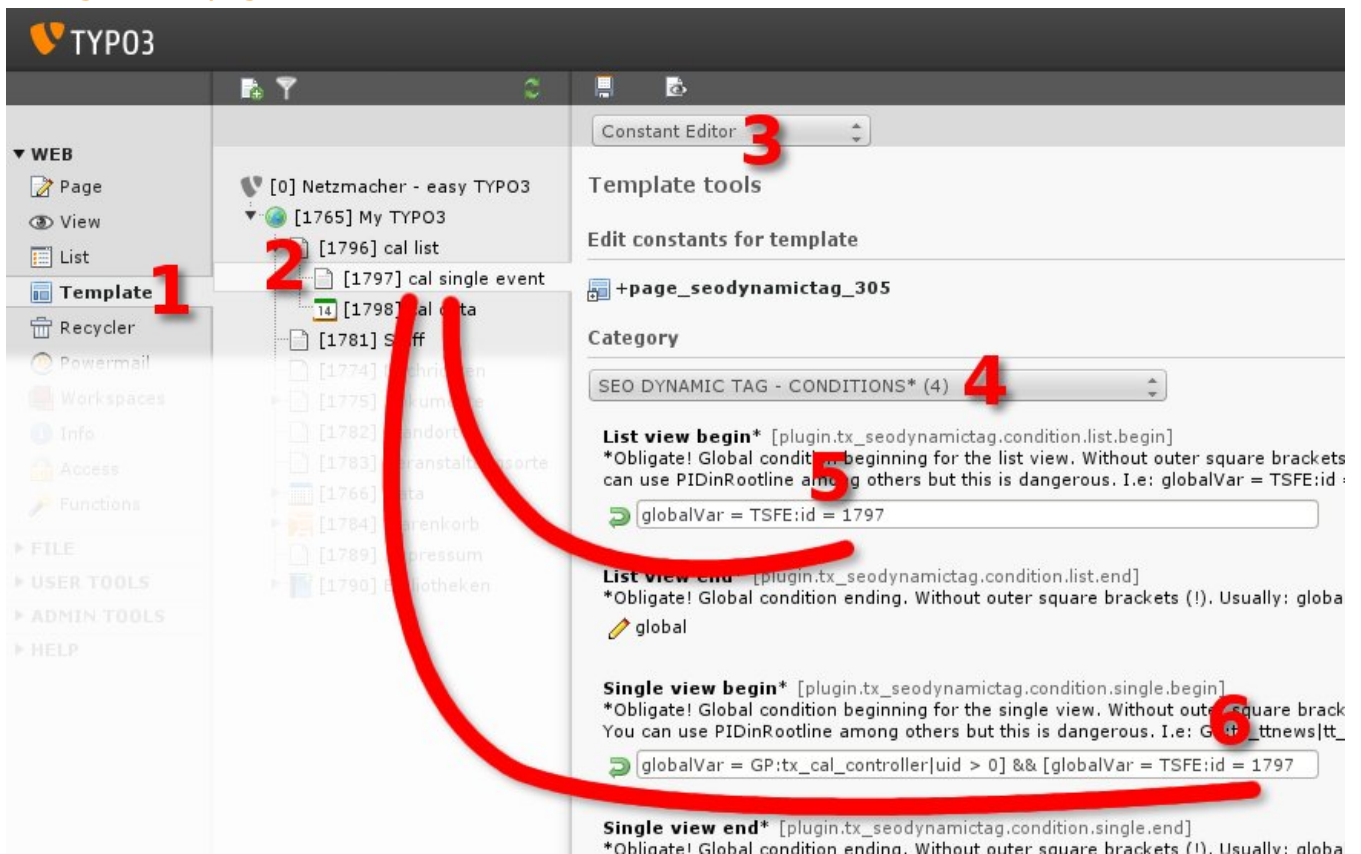


Illustration 6: Calendar Base (cal): Configuration by the Constant Editor

1. Web      Template
2. Page tree      Page with the Calendar Base plugin for the single view (here: [1797] cal single event)
3. Edit area      [Constant editor]
4.      Category [SEO DYNAMIC TAG - CONDITIONS\*]
5.      Replace 'xxx' with the id of your page with the News plugin (here: 1797)
6.      Replace 'xxx' with the id of your page with the News plugin (here: 1797)

If you are interested in details please refer to

- "Workflow" on page 30 below
- "Condition" on page 33 below.



## SEO – Search Engine Optimization – in the TYPO3 frontend

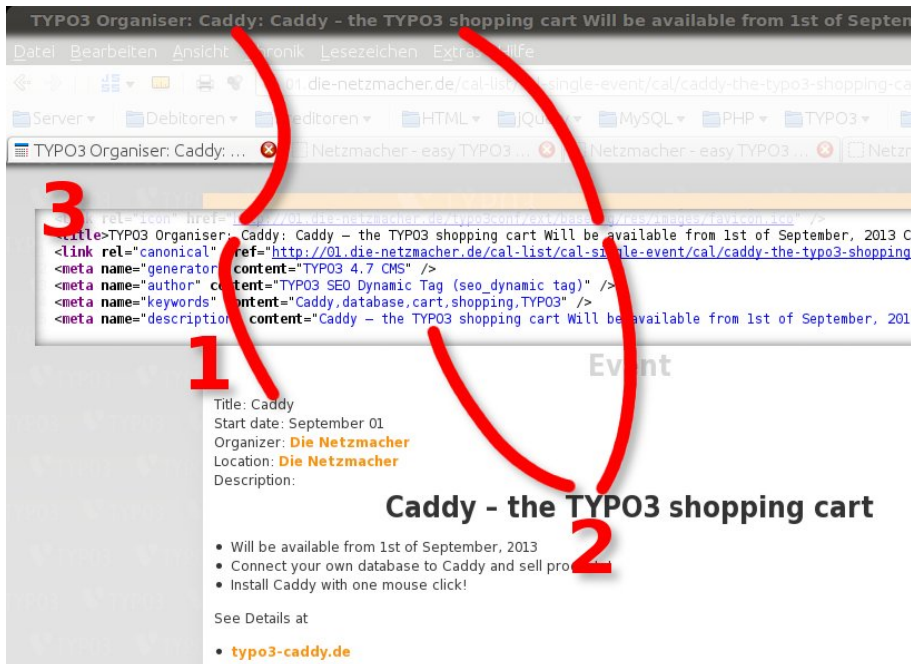


Illustration 7: Calendar Base (cal): SEO – Search Engine Optimization – in the TYPO3 frontend

SEO – Search Engine Optimization – in the TYPO3 frontend:

1. Calendar event title part of the title-tag
2. Calendar event description part of the title tag and content of the description tag
3. HTML code see illustration below

HTML SEO tags

```
<title>TYPO3 Organisier: Caddy: Caddy – the TYPO3 shopping cart Will be available from 1st of September, 2013</title>
<link rel="canonical" href="http://01.die-netzmacher.de/cal-list/cal-single-event/cal/caddy-the-typo3-shopping-cart" />
<meta name="generator" content="TYPO3 4.7 CMS" />
<meta name="author" content="TYPO3 SEO Dynamic Tag (seo_dynamic tag)" />
<meta name="keywords" content="Caddy,database,card,shopping,TYPO3" />
<meta name="description" content="Caddy – the TYPO3 shopping cart Will be available from 1st of September, 2013" />
```

Illustration 8: News (tt\_news): SEO tags

1. <title> field title and field description
2. <link rel="canonical"> URL<sup>1</sup>
3. <meta name="author"> there is no field author, it is taken a default value.
4. <meta name="description"> field description
5. <meta name="keywords"> out of the field description

<sup>1</sup> It depends on realUrl, if you get a path without or with parameters.

## News (tt\_news)

### Initial Situation

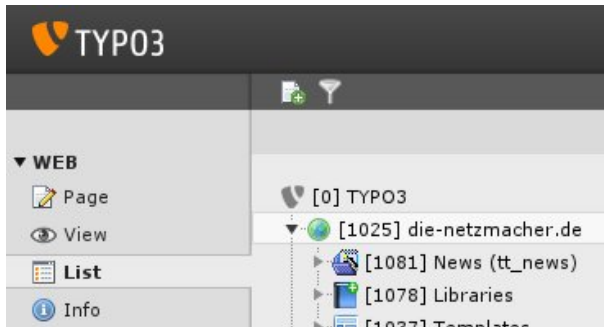


Illustration 9: News (tt\_news): initial situation

- I am using News (tt\_news) for both views – the list and the single view – on my root page: [1025] die-netzmacher.de

### Include the static templates

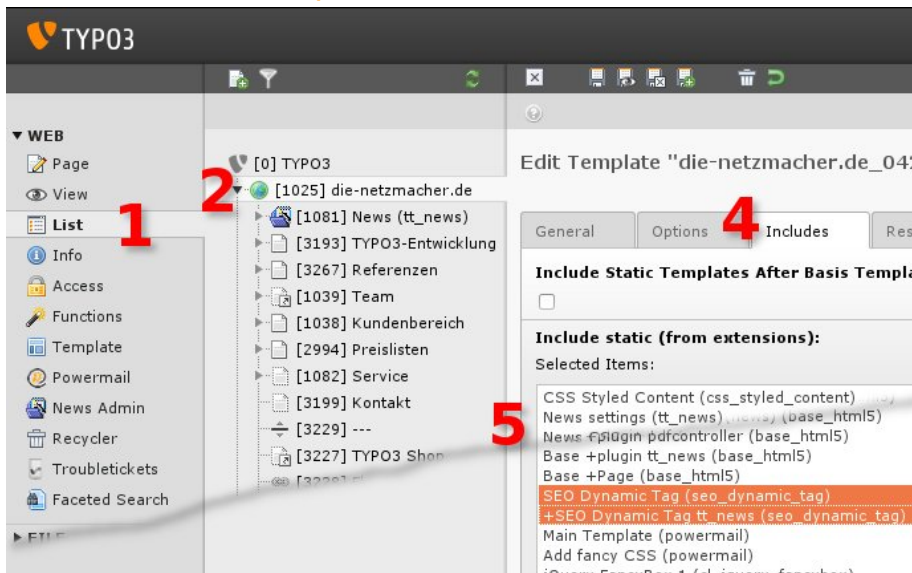


Illustration 10: News (tt\_news): include the static templates

Include the static templates:

- Web List
- Page tree Page with the News plugin for the single view (here: [1025] die-netzmacher.de)
- Edit area Select the TypoScript template<sup>1</sup> for editing (not illustrated)
- Tab [Includes]
- Add the static templates below all templates, which will configure the page object:  
SEO Dynamic Tag (seo\_dynamic\_tag)  
+SEO Dynamic Tag tt\_news (seo\_dynamic\_tag)

<sup>1</sup> If you don't have a template on the current page, please add an extension template.

## Configure the page id

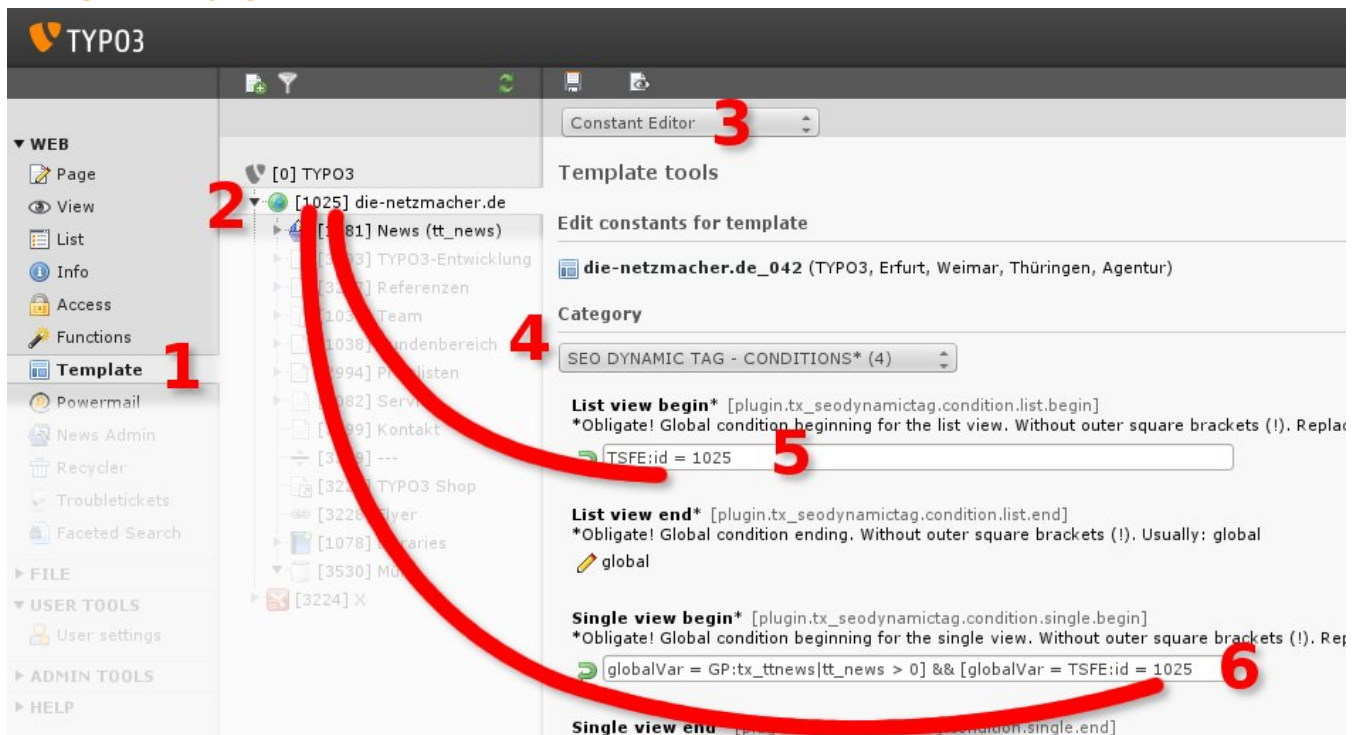


Illustration 11: News (tt\_news): Configuration by the Constant Editor

1. Web                      Template
2. Page tree              Page with the News plugin for the single view (here: [1025] die-netzmacher.de)
3. Edit area              [Constant editor]
4.                          Category [SEO DYNAMIC TAG - CONDITIONS\*]
5.                          Replace 'xxx' with the id of your page with the News plugin (here: 1025)
6.                          Replace 'xxx' with the id of your page with the News plugin (here: 1025)

If you are interested in details please refer to

- "Workflow" on page 30 below
- "Condition" on page 33 below.

## SEO – Search Engine Optimization – in the TYPO3 frontend

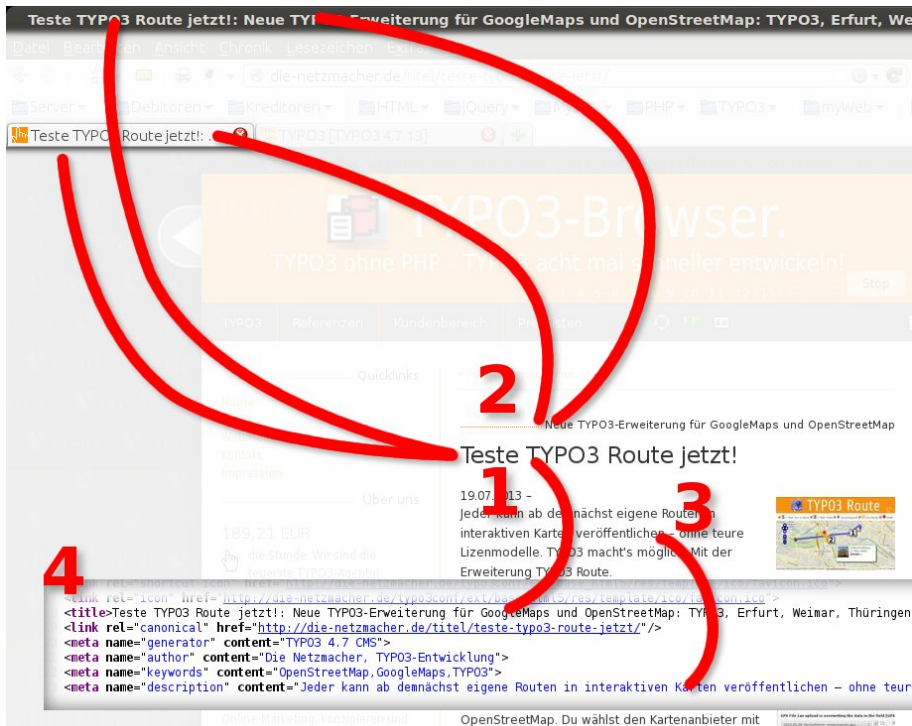


Illustration 12: News (tt\_news): SEO – Search Engine Optimization – in the TYPO3 frontend

See

- <http://die-netzmacher.de/titel/teste-typo3-route-jetzt/>

SEO – Search Engine Optimization – in the TYPO3 frontend:

1. News title part of the title-tag
2. News short part of the title tag
3. News bodytext part of the description tag
4. HTML code see illustration below

HTML SEO tags

```
<title>Teste TYPO3 Route jetzt!: Neue TYPO3-Erweiterung für GoogleMaps und OpenStreetMap: TYPO3, Erfurt, Weimar, Thüringen</title>
<link rel="canonical" href="http://die-netzmacher.de/titel/teste-typo3-route-jetzt/" />
<meta name="generator" content="TYPO3 4.7 CMS">
<meta name="author" content="Die Netzmacher, TYPO3-Entwicklung">
<meta name="keywords" content="OpenStreetMap, GoogleMaps, TYPO3">
<meta name="description" content="Jeder kann ab demnächst eigene Routen in interaktiven Karten veröffentlichen – ohne teure Lizenzen. TYPO3 macht's möglich. Mit der Erweiterung TYPO3 Route." />
```

Illustration 13: News (tt\_news): SEO tags

1. <title> field title and short
2. <link rel="canonical"> URL with the news id only<sup>1</sup>
3. <meta name="author"> field author. If it's empty, a default value will taken.
4. <meta name="description"> field bodytext
5. <meta name="keywords"> field keywords

<sup>1</sup> It depends on realUrl if you get a path without or with parameters.

## Organiser (org)

### SEO Dynamic Tags ready-for-use

If you are installing the Organiser – TYPO3 for lobby and organisers – with the Organiser Installer (org\_installer), there isn't any need for configure anything.

But if you have the Organiser installed already and you update the Organiser from 2.x to 3.x, you have to configure SEO Dynamic Tag. See the instruction below.

### Initial Situation

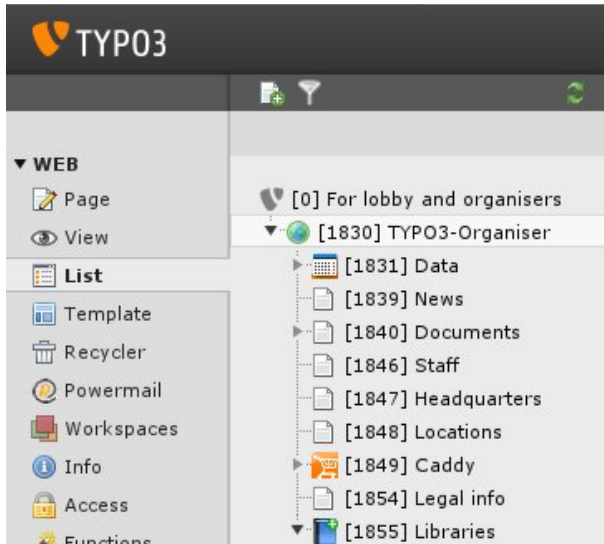


Illustration 14: Organiser (org): initial situation

The illustration above displays the page tree after the one-click-installation.

- You have to set the page id of each page with a single view for SEO Dynamic Tag. This is in the example above:
- [1830] TYPO3-Organiser
  - [1839] News
  - [1840] Documents
  - [1846] Staff
  - [1847] Headquarters
  - [1848] Locations



## Configure the page id

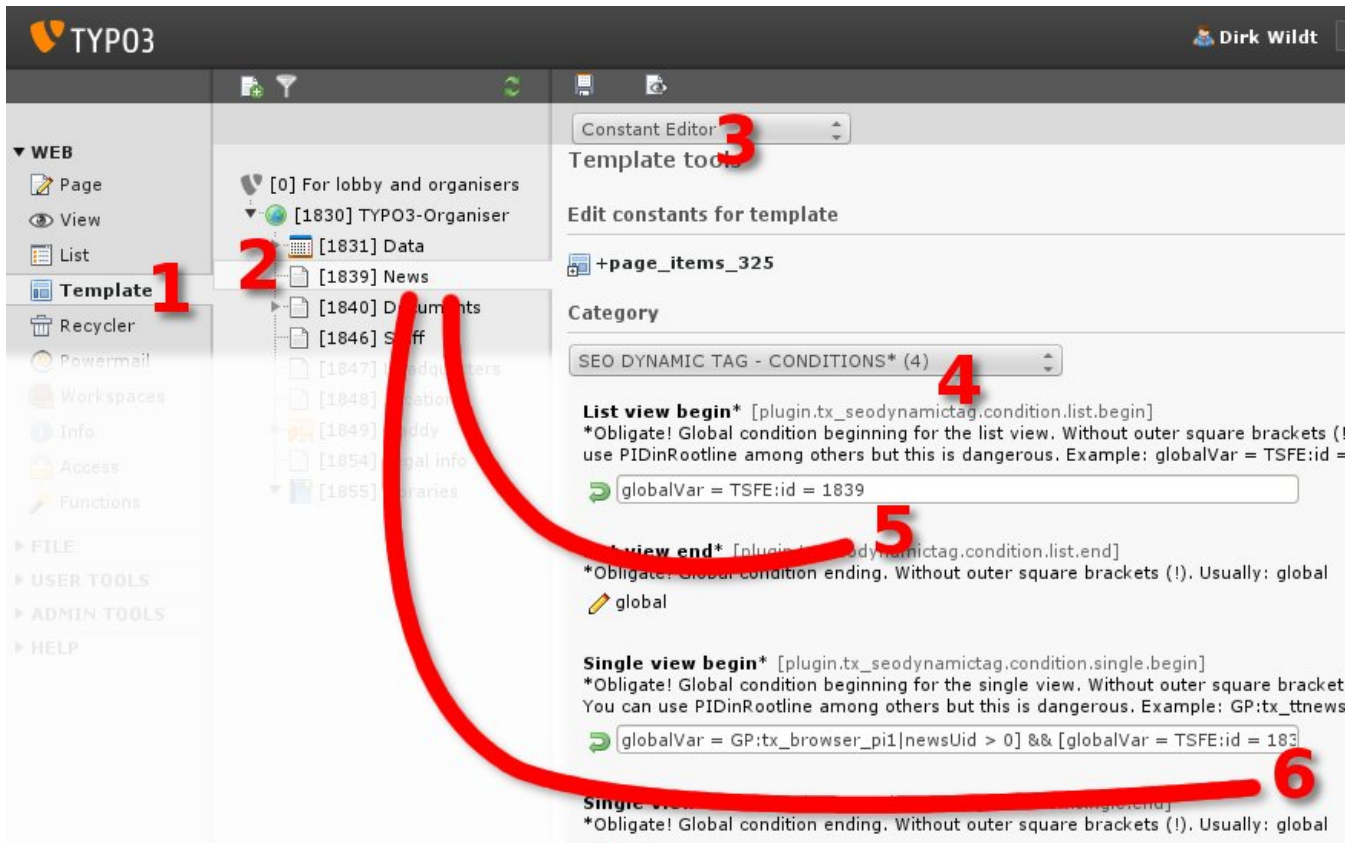


Illustration 15: Organiser (org) News: Configuration by the Constant Editor

1. Web                      Template
2. Page tree              Page with the Organiser News plugin for the single view (here: [1839] News)
3. Edit area                [Constant editor]
4.                            Category [SEO DYNAMIC TAG - CONDITIONS\*]
5.                            Replace 'xxx' with the id of your page with the News plugin (here: 1839)
6.                            Replace 'xxx' with the id of your page with the News plugin (here: 1839)

Repeat the steps 1 to 6 with the other pages too.

If you are interested in details please refer to

- "Workflow" on page 30 below
- "Condition" on page 33 below.

## SEO – Search Engine Optimization – in the TYPO3 frontend

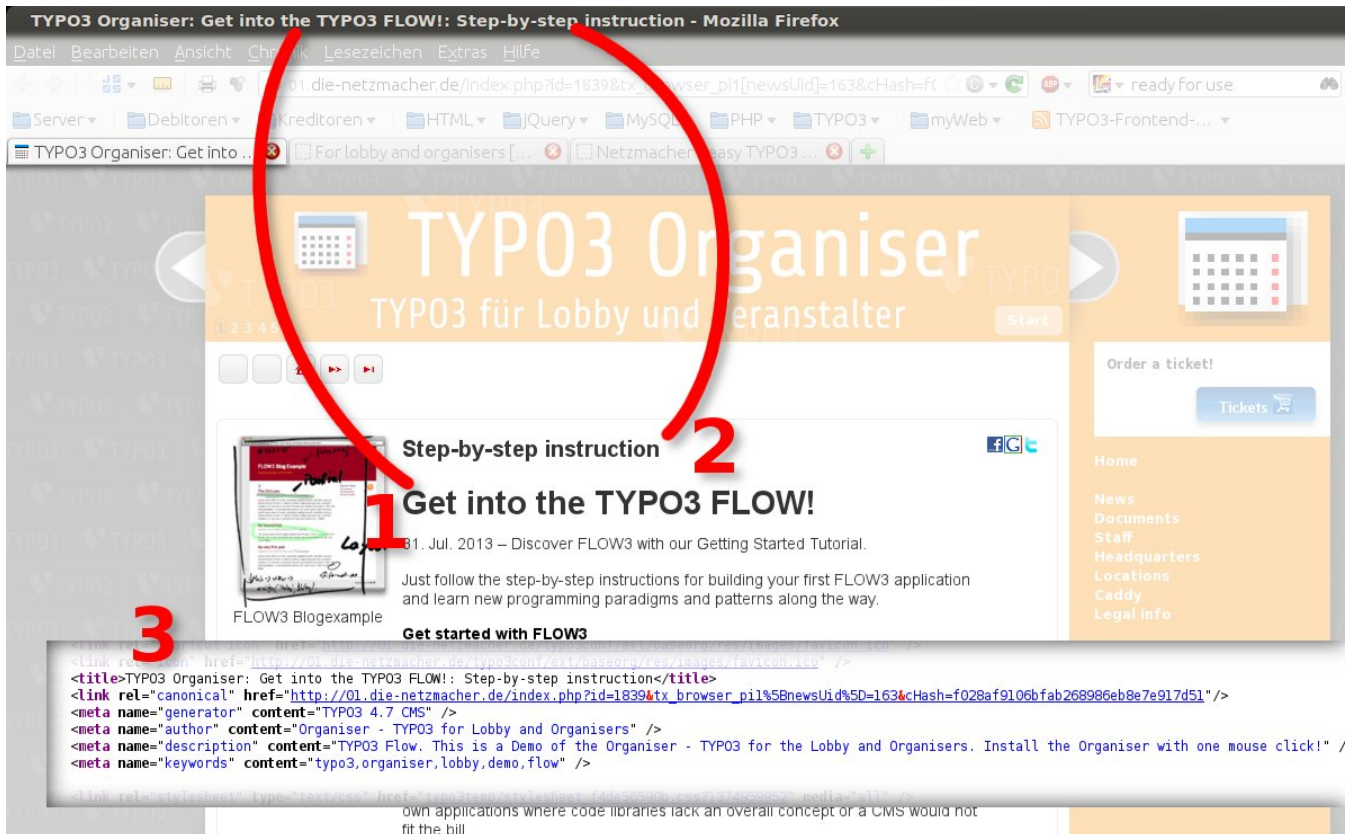


Illustration 16: Organsier (org) News: SEO – Search Engine Optimization – in the TYPO3 frontend

SEO – Search Engine Optimization – in the TYPO3 frontend:

1. News title part of the title-tag
2. News subtitle part of the title tag
3. HTML code see illustration below

HTML SEO tags

```
<title>TYPO3 Organiser: Get into the TYPO3 FLOW!: Step-by-step instruction</title>
<link rel="canonical" href="http://01.die-netzmacher.de/index.php?id=1839&tx_browser_pi1%5BnewsId%5D=163&cHash=f028af9106fab268986eb8e7e917d51"/>
<meta name="generator" content="TYPO3 4.7 CMS" />
<meta name="author" content="Organiser - TYPO3 for Lobby and Organisers" />
<meta name="description" content="TYPO3 Flow. This is a Demo of the Organiser - TYPO3 for the Lobby and Organisers. Install the Organiser with one mouse click!" />
<meta name="keywords" content="typo3,organiser,lobby,demo,flow" />
```

Illustration 17: Organsier (org) News: SEO tags

1. <title> field title and subtitle
2. <link rel="canonical"> URL with the news id only<sup>1</sup>
3. <meta name="author"> field author. If it's empty, a default value will taken.
4. <meta name="description"> field description
5. <meta name="keywords"> field keywords

<sup>1</sup> It depends on realUrl, if you get a path without or with parameters.

## Shop System (tt\_products)

### Initial Situation

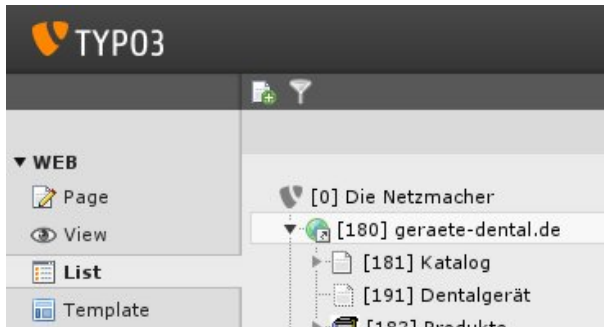


Illustration 18: Shop System (tt\_products): initial situation

- I am using the Shop System (tt\_products) for the single view on page: [191] Dentalgerät

### Include the static templates

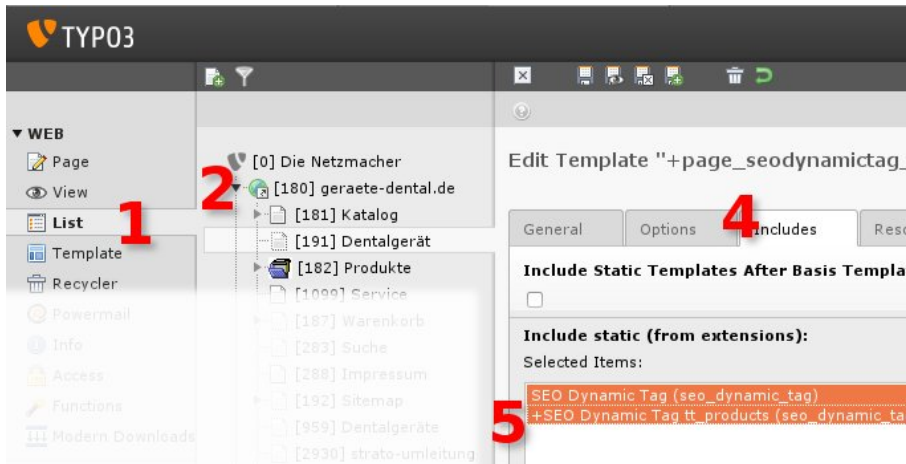


Illustration 19: Shop System (tt\_products): include the static templates

Include the static templates:

- Web List
- Page tree Page with the Shop System plugin for the single view (here: [191] Dentalgerät)
- Edit area Select the TypoScript template<sup>1</sup> for editing (not illustrated)
- Tab [Includes]
- Add the static templates:  
SEO Dynamic Tag (seo\_dynamic\_tag)  
+SEO Dynamic Tag tt\_products (seo\_dynamic\_tag)

<sup>1</sup> If you don't have a template on the current page, please add an extension template.



## Configure the page id

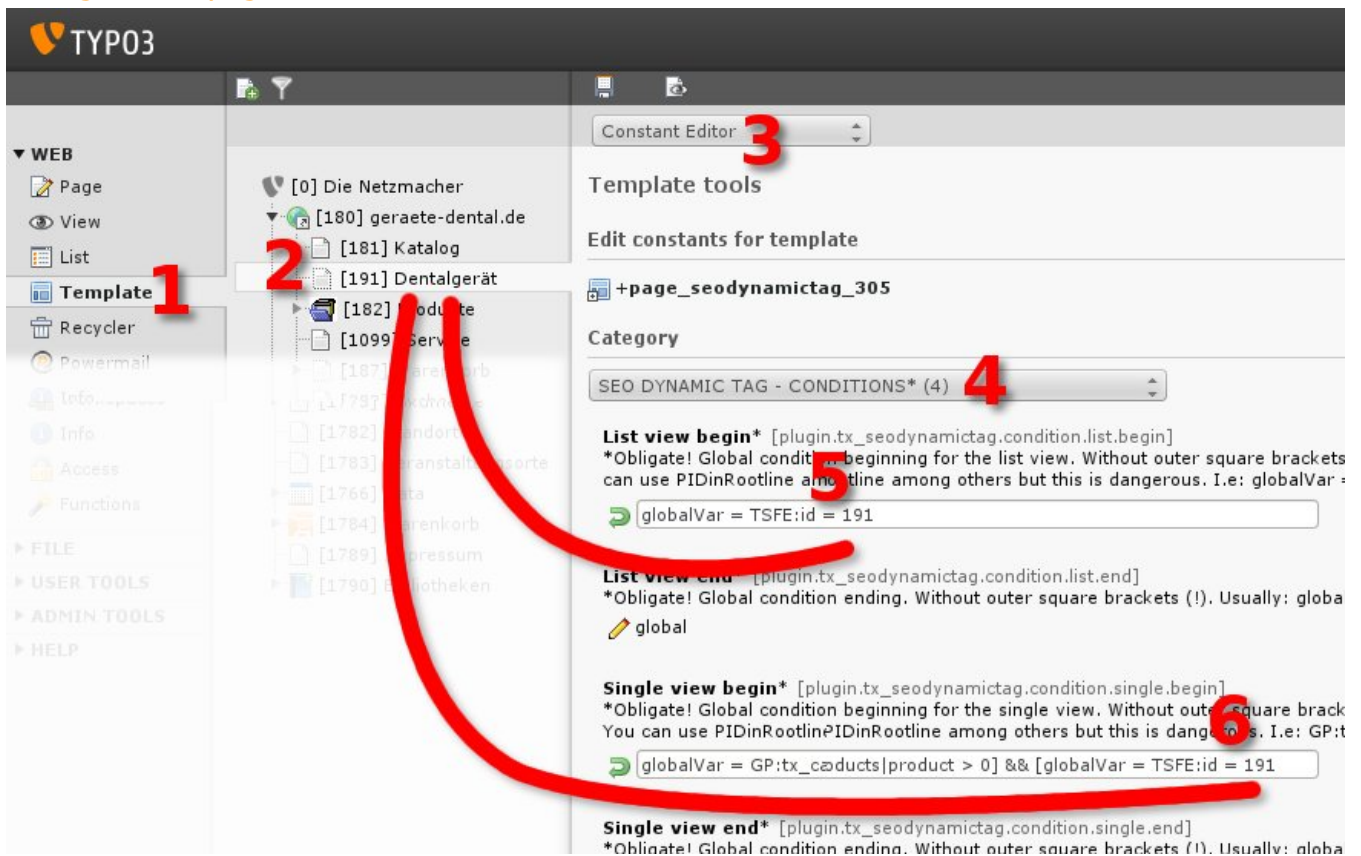


Illustration 20: Shop System (tt\_products): Configuration by the Constant Editor

1. Web Template
2. Page tree Page with the Shop System plugin for the single view (here: [191] Dentalgerät)
3. Edit area [Constant editor]
4. Category [SEO DYNAMIC TAG - CONDITIONS\*]
5. Replace 'xxx' with the id of your page with the Shop System plugin (here: 191)
6. Replace 'xxx' with the id of your page with the Shop System plugin (here: 191)

If you are interested in details please refer to

- "Workflow" on page 30 below
- "Condition" on page 33 below.

## SEO – Search Engine Optimization – in the TYPO3 frontend

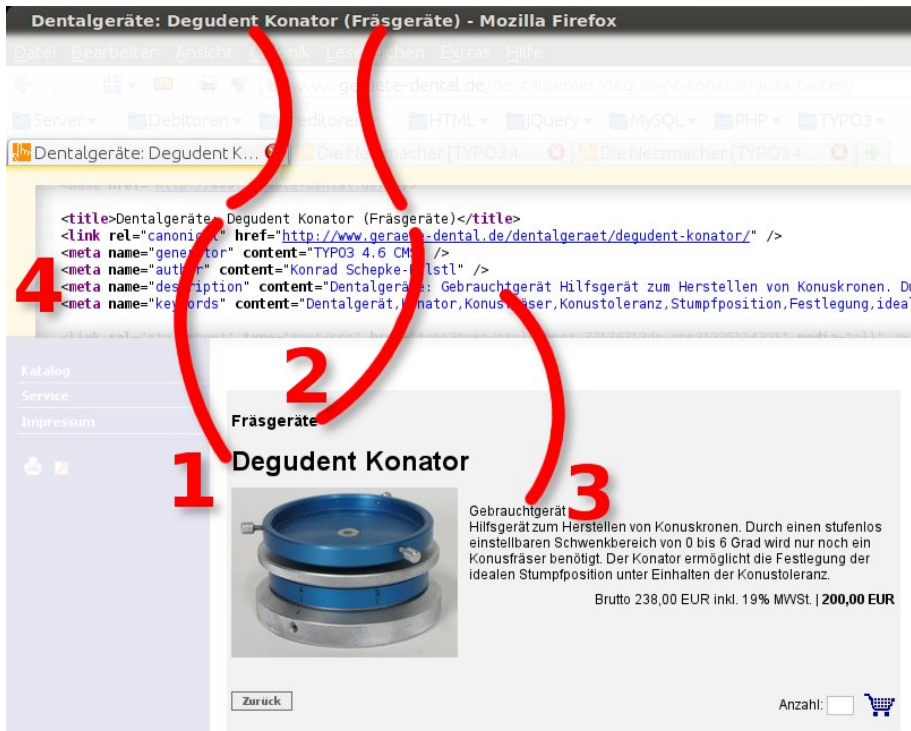


Illustration 21: Shop System (tt\_products): SEO – Search Engine Optimization

SEO – Search Engine Optimization – in the TYPO3 frontend:

- |                     |   |
|---------------------|---|
| 1. Product title    | part of the title-tag and the keywords                  |
| 2. Product category | part of the title-tag and the keywords                  |
| 3. Product note     | content of the description tag and part of the keywords |
| 4. HTML code        | see illustration below                                  |

HTML SEO tags

```
<title>Dentalgeräte: Degudent Konator (Fräsgeräte)</title>
<link rel="canonical" href="http://www.geraete-dental.de/dentalgeraet/degudent-konator/" />
<meta name="generator" content="TYPO3 4.6 CMS" />
<meta name="author" content="Konrad Schepke-Pilstl" />
<meta name="description" content="Dentalgeräte: Gebrauchtgerät Hilfsgerät zum Herstellen von Konuskronen. Durch einen stufenlos einstellbaren Schwenkbereich von 0 bis 6 Grad wird nur noch ein Konusfräser benötigt. Der Konator ermöglicht die Festlegung der idealen Stumpfposition unter Einhalten der Konustoleranz." />
<meta name="keywords" content="Dentalgerät, Konator, Konusfräser, Konustoleranz, Stumpfposition, Festlegung, ideal" />
```

Illustration 22: Shop System (tt\_products): SEO tags

- |                              |   |
|------------------------------|---|
| 1. <title>                   | table.field product.title and category.title                    |
| 2. <link rel="canonical">    | URL <sup>1</sup>  |
| 3. <meta name="author">      | there is no field author, it is taken a default value.          |
| 4. <meta name="description"> | field note  |
| 5. <meta name="keywords">    | out of the product.title, category.title and product.field note |

<sup>1</sup> It depends on realUrl, if you get a path without or with parameters.

## Any other database

### Initial Situation

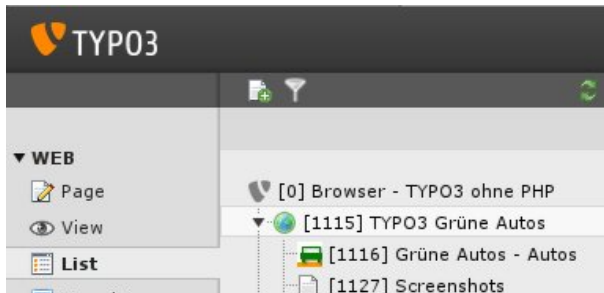


Illustration 23: Any other database: initial situation

Let us suppose

- you have created a database for cars like Grüne Autos (green cars) or you are using this database,
- your page with the plugin for the single view is:  
[1115] TYPO3 Grüne Autos
- the URL parameter for a link to a single view is:  
tx\_browser\_pi1[showUid]<sup>1</sup>

### Include the static template

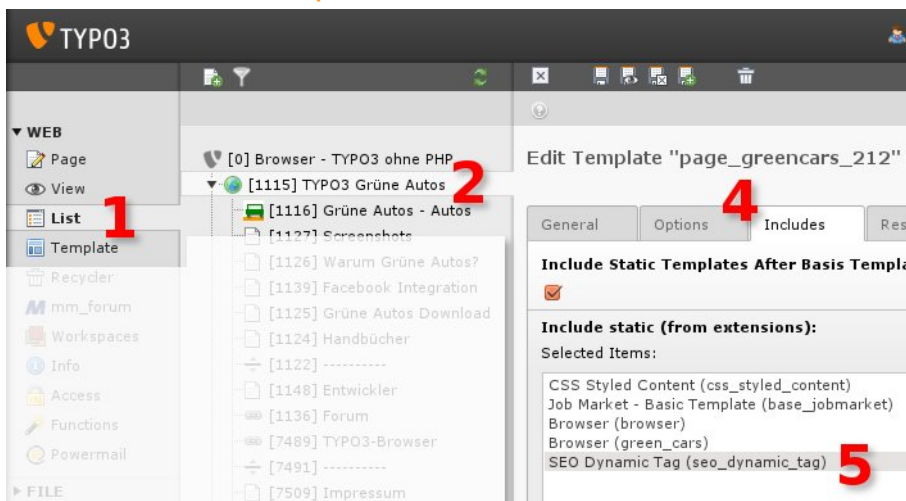


Illustration 24: any other database: include the static template

Include the static template:

1. Web List
2. Page tree Page with the plugin for the single view (here: [1115] TYPO3 Grüne Autos)
3. Edit area Select the TypoScript template<sup>2</sup> for editing (not illustrated)
4. Tab [Includes]
5. Add the static templates:  
SEO Dynamic Tag (seo\_dynamic\_tag)

<sup>1</sup> Green Cars is using the Browser plugin for displaying records. If you are using your own extension, an usual parameter would be: tx\_myextension\_pi1[uid]

<sup>2</sup> If you don't have a template on the current page, please add an extension template.

## Constant Editor

You configure SEO Dynamic Tag supported by the Constant Editor. See the example "Canonical Tag" below.

## Canonical Tag

You have to configure the property "Allowed parameter (single view)" at least.

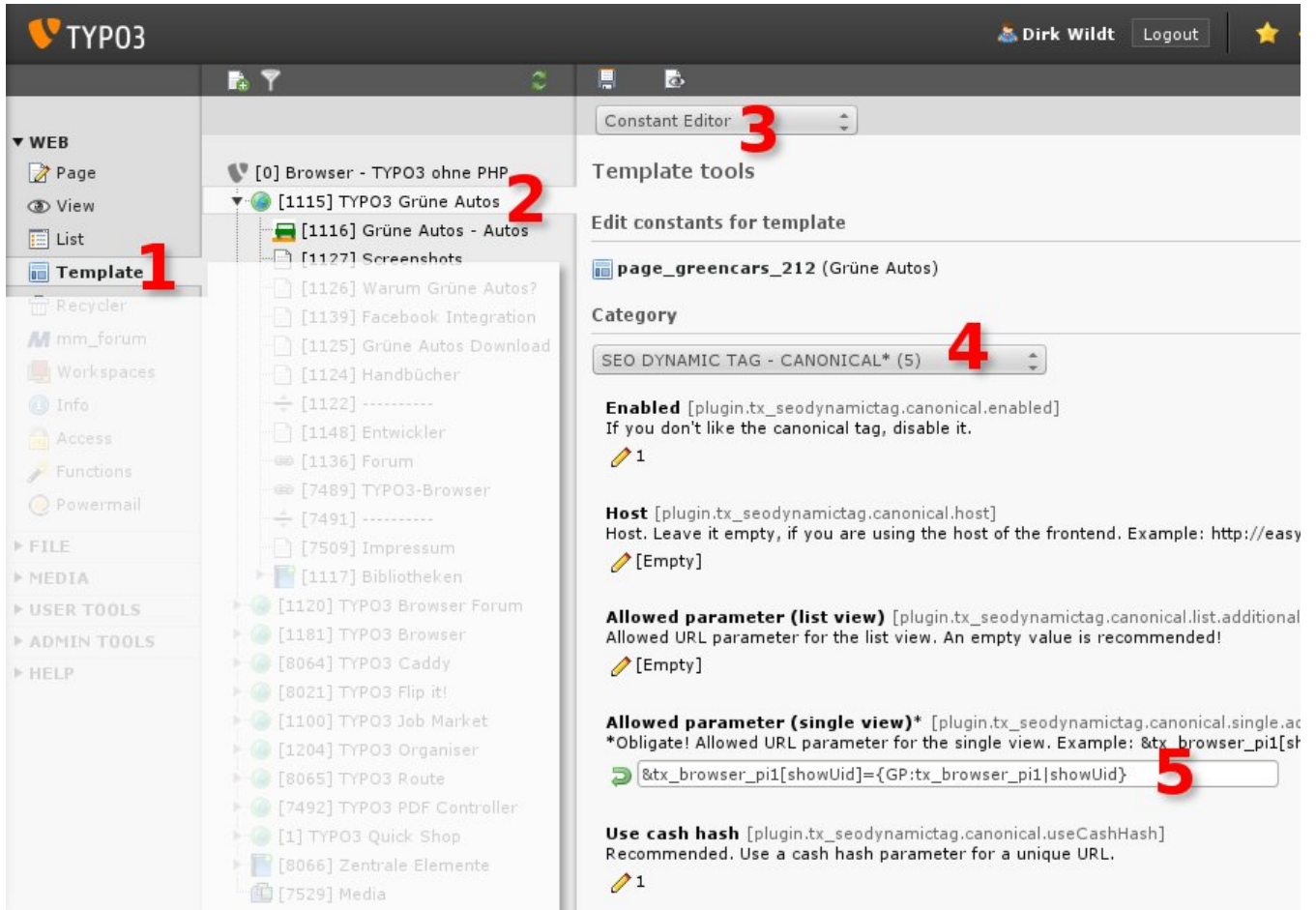


Illustration 25: Any other database: Configuration by the Constant Editor

1. Web                      Template
2. Page tree              Page with the plugin for the single view (here: [1115] TYPO3 Grüne Autos)
3. Edit area                [Constant editor]
4.                            Category [SEO DYNAMIC TAG – CANONICAL TAG\*]
5.                            Field [Allowed parameter (single view)\*]  
                                 Replace  
                                 - 'table' with 'tx\_browser\_pi1' and  
                                 - 'field' with 'showUid'

`&table[field]={GP:table|field}` will become `&tx_browser_pi1[showUid]={GP:tx_browser_pi1|showUid}`

## Details

- "Canonical Tag" on page 31 below
- "Workflow" on page 30 below

## Conditions

You have to configure the properties:

- List view begin
- Single view begin

If you don't know, how to start the Constant Editor, please refer to the section "Canonical Tag" on page 20 above.

1. Web Template
2. Page tree Page with the plugin for the single view (here: [1115] TYPO3 Grüne Autos)
3. Edit area [Constant editor]
4. Category [SEO DYNAMIC TAG – CONDITIONS\*]

### Field [List view begin]

- Replace 'xxx' with the id of the page with the plugin for the single view (here: [1115] TYPO3 Grüne Autos)
- globalVar = TSFE:id = xxx will become globalVar = TSFE:id = 1115

### Field [Single view begin]

- Replace 'table|field' with the URL parameter (here: tx\_browser\_pi1[showUid])
- Replace 'xxx' with the id of the page with the plugin for the single view (here: [1115] TYPO3 Grüne Autos)
- globalVar = GP:table|field > 0] && [globalVar = TSFE:id = xxx will become globalVar = GP: tx\_browser\_pi1[showUid > 0] && [globalVar = TSFE:id = 1115

## Screenshot

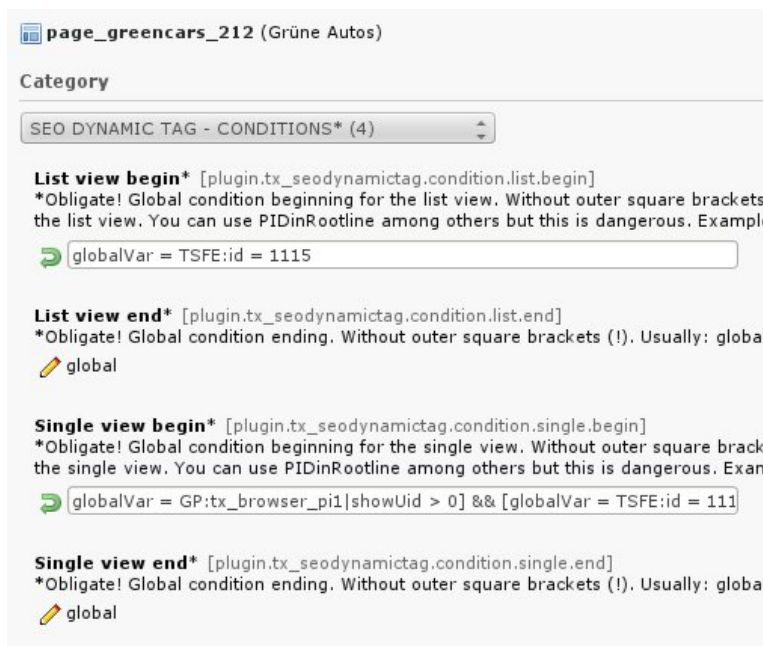


Illustration 26: Any other database: conditions

## Details

- "Condition" on page 27 below
- "Workflow" on page 30 below



## Database

You have to configure the properties:

- table, get parameter for uid, author, description, keywords, title and short

If you don't know, how to start the Constant Editor, please refer to the section "Canonical Tag" on page 20 above.

1. Web Template
2. Page tree Page with the plugin for the single view (here: [1115] TYPO3 Grüne Autos)
3. Edit area [Constant editor]
4. Category [SEO DYNAMIC TAG – DATABASE\*]

## Beforehand

The table of our example – tx\_greencars\_main – isn't a good example for the database configuration:

- There are missing a lot of fields.

Important information are stored in foreign tables.

How to optimise SEO Dynamic Tag for foreign tables is written at the section

- "Values from foreign tables" on page 26 below

## table

Here: tx\_greencars\_main

## get parameter for uid

Here: tx\_browser\_pi1[showUid]

## author

tx\_greencars\_main hasn't any author field. We take a constant. Here: "TYPO3 Green Cars"

## description

tx\_greencars\_main hasn't any description field. We take the title field.

## keywords

tx\_greencars\_main hasn't any keywords field. We take the title field.

## title

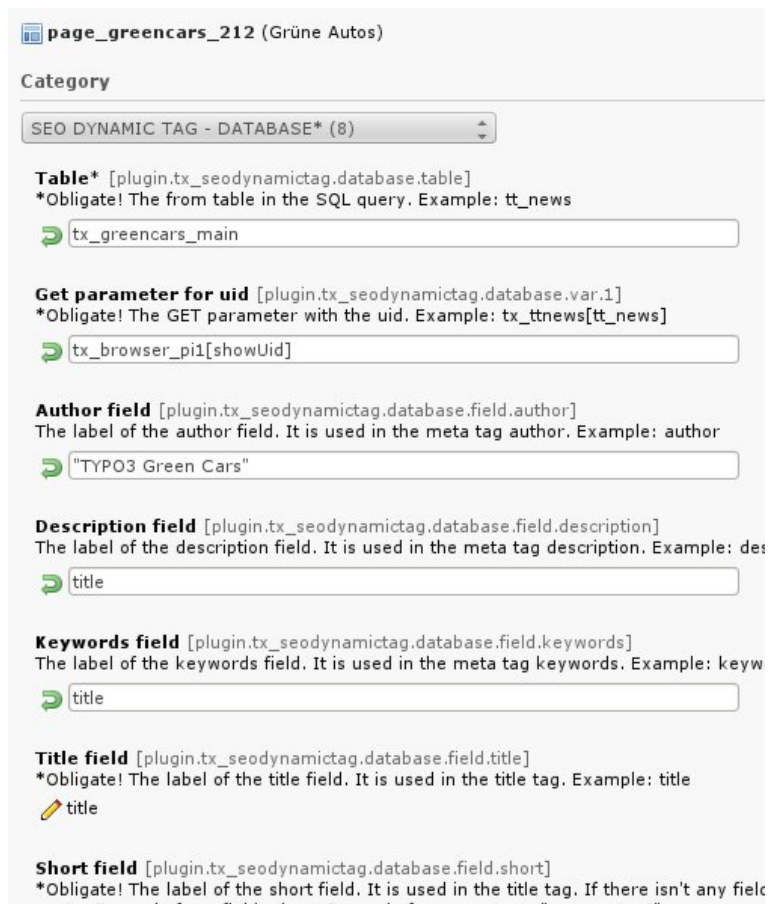
Here: title.

## short

tx\_greencars\_main hasn't any short field. We take the title field.

## Screenshot

See illustration on the right hand.



**Category**

SEO DYNAMIC TAG - DATABASE\* (8)

**Table\*** [plugin.tx\_seodynamictag.database.table]  
\*Obligate! The from table in the SQL query. Example: tt\_news

tx\_greencars\_main

**Get parameter for uid** [plugin.tx\_seodynamictag.database.var.1]  
\*Obligate! The GET parameter with the uid. Example: tx\_ttnews[tt\_news]

tx\_browser\_pi1[showUid]

**Author field** [plugin.tx\_seodynamictag.database.field.author]  
The label of the author field. It is used in the meta tag author. Example: author

"TYPO3 Green Cars"

**Description field** [plugin.tx\_seodynamictag.database.field.description]  
The label of the description field. It is used in the meta tag description. Example: des

title

**Keywords field** [plugin.tx\_seodynamictag.database.field.keywords]  
The label of the keywords field. It is used in the meta tag keywords. Example: keyw

title

**Title field** [plugin.tx\_seodynamictag.database.field.title]  
\*Obligate! The label of the title field. It is used in the title tag. Example: title

title

**Short field** [plugin.tx\_seodynamictag.database.field.short]  
\*Obligate! The label of the short field. It is used in the title tag. If there isn't any field

Illustration 27: Any other database: database

## Details

- "Database" on page 27 below
- "Workflow" on page 30 below

## Use values from foreign tables

- "Values from foreign tables" on page 26 below

## Default values

There isn't any need to configure default values.

We configure the property "Description appendix" in the example below.

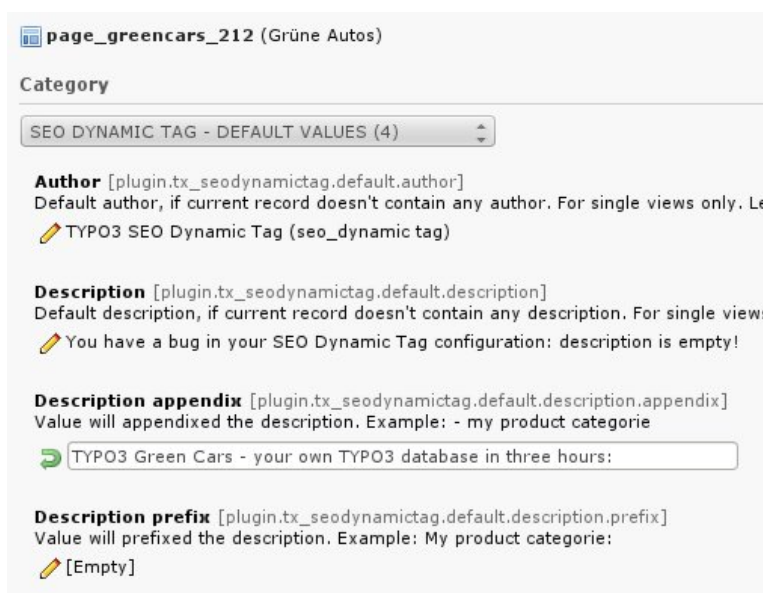
If you don't know, how to start the Constant Editor, please refer to the section "Canonical Tag" on page 20 above.

1. Web                      Template
2. Page tree              Page with the plugin for the single view (here: [1115] TYPO3 Grüne Autos)
3. Edit area              [Constant editor]
4.                          Category [SEO DYNAMIC TAG – DEFAULT VALUES]

## Field [Description appendix]

Here: TYPO3 Green Cars - your own TYPO3 database in three hours:

## Screenshot



**page\_greencars\_212** (Grüne Autos)

**Category**

SEO DYNAMIC TAG - DEFAULT VALUES (4)

**Author** [plugin.tx\_seodynamictag.default.author]  
Default author, if current record doesn't contain any author. For single views only. Le  
✎ TYPO3 SEO Dynamic Tag (seo\_dynamic tag)

**Description** [plugin.tx\_seodynamictag.default.description]  
Default description, if current record doesn't contain any description. For single view:  
✎ You have a bug in your SEO Dynamic Tag configuration: description is empty!

**Description appendix** [plugin.tx\_seodynamictag.default.description.appendix]  
Value will be appended to the description. Example: - my product categorie  
➡ TYPO3 Green Cars - your own TYPO3 database in three hours:

**Description prefix** [plugin.tx\_seodynamictag.default.description.prefix]  
Value will be prefixed to the description. Example: My product categorie:  
✎ [Empty]

Illustration 28: Any other database: default values

## Details

- "Default" on page 28 below
- "Workflow" on page 30 below

## Keywords

There isn't any need to configure keywords.

We configure the property "Forced list" in the example below.

If you don't know, how to start the Constant Editor, please refer to the section "Canonical Tag" on page 20 above.

1. Web                      Template
2. Page tree              Page with the plugin for the single view (here: [1115] TYPO3 Grüne Autos)
3. Edit area              [Constant editor]
4.                          Category [SEO DYNAMIC TAG – KEYWORDS]

## Field [Forced list]

Here: TYPO3, Green Cars, Grüne Autos

## Screenshot



The screenshot shows the 'Category' dropdown set to 'SEO DYNAMIC TAG - KEYWORDS (7)'. Below it, several configuration options are listed, each with a description and a value:

- Default** [plugin.tx\_seodynamictag.keywords.default]: Default keywords, if current record doesn't contain any keyword. For single views or bodytext or description for generating keywords. That isn't the case, if you using a field. Value: **TYPO3, SEO, Dynamic Tag, Bug**
- Amount** [plugin.tx\_seodynamictag.keywords.amount]: Maximum amount of keywords. Recommended: less is more! Value: **10**
- Minimum length** [plugin.tx\_seodynamictag.keywords.minLength]: Minimum length of a single keyword. Value: **6**
- Move to keywords** [plugin.tx\_seodynamictag.keywords.moveToKeywords]: Move source to keywords. Recommended, if your field for keywords doesn't contain bodytext or description for generating keywords. That isn't the case, if you using a field. Value: **1**
- Positive list** [plugin.tx\_seodynamictag.keywords.positiveList]: List with keywords, which will used, if they are part of the keyword list but are shorter amount of keywords. Value: **TYPO3, SEO**
- Negative list** [plugin.tx\_seodynamictag.keywords.negativeList]: List with keywords, which will excluded from the list of found keywords. Value: **[Empty]**
- Forced list** [plugin.tx\_seodynamictag.keywords.forcedList]: List with keywords, which will prepend to the list of keywords. List is independent of the other lists. Value: **[Empty]**

Illustration 29: Any other database: keywords

## Details

- "Keywords" on pag 28 below
- "Workflow" on page 30 below



## SEO – Search Engine Optimization – in the TYPO3 frontend

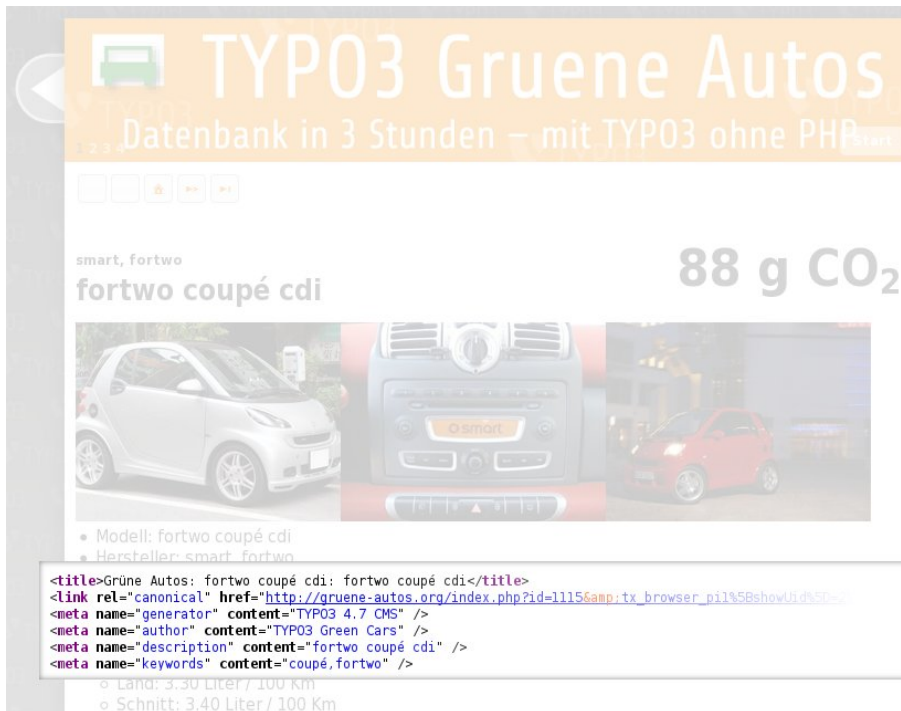


Illustration 30: Any other database: SEO – Search Engine Optimization

SEO – Search Engine Optimization – in the TYPO3 frontend:

1. title part of the title-tag, the description and the keywords
2. HTML code see illustration below

HTML SEO tags

```
<title>Grüne Autos: fortwo coupé cdi: fortwo coupé cdi</title>
<link rel="canonical" href="http://gruene-autos.org/index.php?id=1115&amp;tx_browser_pi1%5BshowUid%5D=0" />
<meta name="generator" content="TYPO3 4.7 CMS" />
<meta name="author" content="TYPO3 Green Cars" />
<meta name="description" content="fortwo coupé cdi" />
<meta name="keywords" content="coupé, fortwo" />
```

Illustration 31: Any other database: SEO tags

1. <title> the title will generated from title and short. Here: title and title
2. <link rel="canonical"> URL<sup>1</sup>
3. <meta name="author"> Here: the given constant
4. <meta name="description"> title
5. <meta name="keywords"> from the title field

### Use values from foreign tables

The table of our example – tx\_greencars\_main – isn't a good example for the database configuration.

How to optimise SEO Dynamic Tag for foreign tables is written at the section

- "Values from foreign tables" on page 26 below

### Debugging

See

- "Check it!" on page 35 below
- "Debug Mode" on page 36 below

<sup>1</sup> It depends on realUrl, if you get a path without or with parameters.

## Values from foreign tables

You can use values from one table only, if you like to configure SEO Dynamic Tag by the Constant Editor.

It is possible, to use values from foreign tables too. You have to configure the SQL statement manually.

### Example with the "Any other database" on page 19 above.

We are using the Green Cars database. Manufacturers are stored in the table tx\_greencars\_manufacturer.

The relation is:

tx\_greencars\_main.manufacturer > tx\_greencars\_manufacturer.uid

We like to have a title-tag like:

- manufacturer: car

## TypoScript

We find the SEO Dynamic Tag configuration of the title tag<sup>1</sup> at

plugin.tx\_seodynamictag\_pil.page.49442.50

Our configuration of the query:

```

1.      plugin.tx_seodynamictag_pil {
2.          page {
3.              49442 {
4.                  50 {
5.                      query {
6.                          select = CONCAT(manufacturer.title, ': ', car.title) as value
7.                          from   = tx_greencars_main as car, tx_greencars_manufacturer as manufacturer
8.                          where  (
9.                              car.uid           = $1
10.                             AND car.manufacturer = manufacturer.uid
11.                             AND car.deleted    = 0
12.                             AND car.hidden     = 0
13.                          )
14.                      var.1 = tx_browser_pil[showUid]
15.                  }
16.              }
17.          }
18.      }
19.  }
20.
21.      [{plugin.tx_seodynamictag.condition.single.begin}]
22.      {plugin.tx_seodynamictag.page.tsProperty} {
23.          {plugin.tx_seodynamictag.page.object} < plugin.tx_seodynamictag_pil.page.49442
24.      }
25.      [{plugin.tx_seodynamictag.condition.single.end}]
26.  
```

From line 6 to 14: individual SQL configuration

From line 21 to 25: You have to init the page object again. Here: single view only.

### Example with Shop System (tt\_products)

SEO Dynamic Tag has a more detailed example.

See the template for Shop System (tt\_products) at

- static/tt\_products/constants.txt
- static/tt\_products/setup.txt

## Debugging

See

- "Check it!" on page 35 below
- "Debug Mode" on page 36 below

<sup>1</sup> See "Reference > Title" on page 39 below

# User Interface – TypoScript Constant Editor

## Canonical

See details at "Canonical Tag" on page 31 below.

plugin.tx\_seodynamictag.canonical

Property:	Data type:	Description:	Default:
enabled	boolean	If you don't like the canonical tag, disable it.	1
host	string	Leave it empty, if you are using the host of the frontend. Example: <code>http://easy-typo3.com/</code>	
list.additionalParams	string	Allowed parameter (list view): Allowed URL parameter for the list view. An empty value is recommended!	
single.additionalParams	string	Allowed parameter (list view)*: *Obligate! Allowed URL parameter for the single view <b>Default: <code>&amp;table[field]={GP:table field}</code></b>	see description
useCashHash	boolean	Recommended. Use a cash hash parameter for a unique URL. Allowed URL parameter for the single view	1

## Condition

See details at "Condition" on page 33 below.

plugin.tx\_seodynamictag.canonical

Property:	Data type:	Description:	Default:
list.begin	string	Obligate! Global condition beginning for the list view. Without outer square brackets (!). Replace 'xxx' with the id of the page with your plugin for the list view. You can use PIDinRootline among others but this is dangerous. Example: <code>globalVar = TSFE:id = 123</code> <b>Default: <code>globalVar = TSFE:id = xxx</code></b>	see description
list.end	string	Obligate! Global condition ending. Without outer square brackets (!). Usually: <code>global</code>	global
single.begin	string	Global condition beginning for the single view. Without outer square brackets (!). Replace 'xxx' with the id of the page with your plugin for the single view. You can use PIDinRootline among others but this is dangerous. Example: <code>GP:tx_ttnews[tt_news &gt; 0] &amp;&amp; [globalVar = TSFE:id = 123</code> <b>Default: <code>GP:table[field] &gt; 0] &amp;&amp; [globalVar = TSFE:id = xxx</code></b>	see description
single.end	string	Global condition ending. Without outer square brackets (!). Usually: <code>global</code>	global

## Database

plugin.tx\_seodynamictag.database

Property:	Data type:	Description:	Default:
table	string	The from table in the SQL query. Example: <code>tt_news</code>	
var.1	string	The GET parameter with the uid. Example: <code>tx_ttnews[tt_news]</code>	
field.author	string	The label of the author field. Example: <code>author</code>	author
field.description	string	The label of the description field. Example: <code>description</code>	description
field.keywords	string	The label of the keywords field. Example: <code>keywords</code>	keywords
field.title	string	The label of the title field. Example: <code>title</code>	title
field.short	string	The label of the short field. Example: <code>short</code> . You can use a constant too. You have to wrap it with double quotes like: <code>"my constant"</code> .	short

## Debugging

See details at "Debug Mode" on page 36 below.

plugin.tx\_seodynamictag.debugging

Property:	Data type:	Description:	Default:
all	boolean	If enabled, you will get a report for all options from below in the frontend	
author	boolean	If enabled, you will get a report for the meta tag author in the frontend	
canonical	boolean	If enabled, you will get a report for the canonical tag in the frontend	
description	boolean	If enabled, you will get a report for the meta tag description in the frontend	
keywords	boolean	If enabled, you will get a report for the meta tag keywords in the frontend	
title	boolean	If enabled, you will get a report for the title in the frontend	

## Default

plugin.tx\_seodynamictag.default

Property:	Data type:	Description:	Default:
author	boolean	Default author, if current record doesn't contain any author. For single views only. Leave it empty, if you don't like a default author. <b>Default: TYPO3 SEO Dynamic Tag (seo_dynamic tag)</b>	see description
description	string	Default description, if current record doesn't contain any description. For single views only. Leave it empty, if you don't like a default description. <b>Default: You have a bug in your SEO Dynamic Tag configuration: description is empty!</b>	see description
description.prefix	string	Value will prefixed the description. Example: My product categorie:	
description.appendix	string	Value will appended the description. Example: - my product categorie	

## Keywords

plugin.tx\_seodynamictag.page

Property:	Data type:	Description:	Default:
default	string	Default keywords, if current record doesn't contain any keyword. For single views only. Leave it empty, if you don't like default keywords. <b>Default: TYPO3, SEO, Dynamic Tag, Bug</b>	see description
amount	int	Maximum amount of keywords. Recommended: less is more!	
minLength	int	Minimum length of a single keyword	
moveToKeywords	boolean	Move source to keywords. Recommended, if your field for keywords doesn't contain a keyword list. This is the case, if you are using a field like bodytext or description for generating keywords. That isn't the case, if you using a field like keywords for keywords.	0
positiveList	string	List with keywords, which will used, if they are part of the keyword list but are shorter than the minimum word length. List is not independent of amount of keywords.	
negativeList	string	List with keywords, which will excluded from the list of found keywords.	
forcedList	string	List with keywords, which will prepend to the list of keywords. List is independent of amount of keywords and minimum length for a keyword.	

## Page

plugin.tx\_seodynamictag.page

Property:	Data type:	Description:	Default:
tsProperty	boolean	The label of your page in your TypoScript configuration. Usually: page	page
object	string	The seo_dynamic_tag element of your page object like page.49442. 49442 is the number of the issue on TYPO3 forge and should be unique.	49442

# Workflow

## In principle

- You configure SEO Dynamic Tag.
- You configure a condition delivered by SEO Dynamic Tag.
- If your condition is met, your SEO-Dynamic-Tag configuration will taken into the page object. And it will override the existing configuration.

## Schema

You have a page object like below.

```
page = PAGE
page {
  * your content
  * some other stuff
  meta {
    * your configuration
  }
}
```

You configure the SEO-Dynamic-Tag condition. Details at "Condition" on page 33 below.

**[Your SEO-Dynamic-Tag Condition]**

If condition doesn't meet, nothing will happen:

**[Your SEO-Dynamic-Tag Condition == false]**

```
page = PAGE
page {
  * your content
  * some other stuff
  meta {
    * your configuration
  }
}
```

If condition meets, the page object will changed:

**[Your SEO-Dynamic-Tag Condition == true]**

```
page = PAGE
page {
  * your content
  49442 {
    * canonical tag
    * metaTag author
    * metaTag description
    * metaTag keywords
    * title author
  }
  * some other stuff
  meta {
    * taken from page.49442.metaTag.*
  }
}
```

# Canonical Tag

## An unique page should have an unique URL

An unique page should have an unique URL – of course!

But there are a lot of reasons, why a unique page hasn't an unique URL.

The both importants are:

- Plugins control the output of their content by parameters.
- Search Engines confuse parameters.

### But plugins control the output of their content by parameters

- Some plugins generates a link to a single view with a back pid. If the single view is accessible by more than one plugin, you will get non unique URL.
- Usually a record in a single view is accessible by a search. Some plugins send the search parameters in the URL.
- Some plugin control the design of the single view by parameters.
- There are a lot of more examples.

### But search engines confuse parameters

Last but not least search engines confuse parameters.

- They double and triple parameters like .../title/my\_news/title/my\_news/title/my\_news
- They append irreproducible parameters like ... /title/my\_news/and/this/is/absolute/nonsense

### Costs of non unique URL

Non unique URL have high costs:

- Search engines don't index a unique page once but a lot of times.
- When search engines update their index, they won't call a unique page once but a lot of times.
- You will get a busy server.
- Search engines could evaluate unique pages with non unique URL as spam. This pages can loose their ranking.

### Solution: the canonical tag

The canonical tag should solve the problem of non unique URL.

The idea is, that a page with a unique URL contains this unique URL in the canonical tag. But this sounds easier, than it is.

The majority of search engines respect the canonical tag. Respect means: they try to check, if the URL in the canonical tag is the unique URL actual – they don't take the URL of the canonical tag without a successful checkup.

## The canonical tag of SEO Dynamic Tag

Before a SEO tool like SEO dynamic tag can render a canonical tag, a determination in principle is needed:

Should the rendering controlled by

- allowed (white list) or
- forbidden (black list)

parameters?

### Allowed parameters (white list)

You have to allow parameters.

If the URL of the single view contains not allowed parameters, they won't respected.

Advantage: you have an unique URL.

Disadvantage: the URL can be wrong.

For example: the extension calendar base (cal) needs three parameters at least for displaying a single view.  
If you miss one of these parameters, you have an unproper URL in the canonical tag.

### Forbidden parameters (black list)

You forbid parameters.

If the URL of the single view contains a forbidden parameter, it won't respected.

Advantage: you have a proper URL usually.

Disadvantage: the URL isn't unique in every case.

Take the examples from above:

- .../title/my\_news/title/my\_news/title/my\_news  
This URL contains only allowed parameters
- ... /title/my\_news/and/this/is/absolute/nonsense  
You can't know, which nonsense parameters are prefixed or appended to an URL. And so you can't forbid nonsense parameters.

### SEO Dynamic Tag respects allowed parameters only by default (white list)

By default SEO Dynamic Tag renders URL for the canonical tag by allowed parameters only (white list).

It is possible too, to use the other principle – to forbid parameter (black list).

But you can't do this with the user interface, you have to configure TypoScript manually.

### Website wide configuration is DANGEROUS!

Because of the default behavior it is not recommended, to configure the canonical tag with SEO Dynamic tag website wide.

It is recommended to configure the canonical tag for pages with list views and single views only!

By default SEO Dynamic Tag doesn't render any canonical tag.

See details at "Condition" on page 33 below.

### Allow a parameter

You are allowing parameters with the Constant Editor:

- See category [SEO DYNAMIC TAG – CANONICAL]
- Look for "Allowed parameter ..."

The illustration on the right hand presents an example for tt\_news in the single view:

- &tx\_ttnews[tt\_news]={GP:tx\_ttnews|tt\_news}

This mean:

The URL will contain only

- the id of the page by default and
- the id of the news.



Illustration 32: Allowed parameter in the single view



## Condition

SEO Dynamic Tag doesn't do anything by default.

You have to allow SEO Dynamic Tag by a condition, to change your page object. See "Workflow" on page 30 above.

You define a condition by the Constant Editor (see illustration on the right hand).

## Recommended pages

It is recommended, to allow SEO Dynamic Tag to change the page object on a page, which contains a plugin for a

- list view and / or
- a single view.

## List views

This is a recommended condition for a list view:

- If page with my list view is called.

This is in TypeScript syntax:

- `[globalVar = TSFE:id = id_of_page_with_list_view]`

You don't need to know this syntax.

SEO Dynamic Tag delivers this condition in the user interface like:

- `[globalVar = TSFE:id = xxx]`

You have to replace xxx with the id of your page with the list view plugin only.

BE AWARE:

The value in the user interface is

- `globalVar = TSFE:id = xxx`

Don't wrap the condition in square brackets! This is a technical need.

## Service in list views

SEO Dynamic Tag adds only the canonical tag in list views.

By default no parameter is respected.

Example:

- Your list view has the URL:  
.../news or  
.../index.php?id=123
- By default the canonical tag will be:  
.../news or  
.../index.php?id=123

If your list view has a page browser, than the following URL are possible:

- Your list view has the URL:  
.../news/pointer/23  
.../index.php?id=123&tx\_myext\_pi1[pointer]=23

But the URL of the canonical tag will be untouched:

- It will be:  
.../news or  
.../index.php?id=123

If you like to allow a parameter for the canonical tag, please refer to "Allow a parameter" on page 32 above.



Illustration 33: Conditions by SEO Dynamic Tag Constant Editor

## Single views

This is a recommended condition for a list view:

- If a page with my single view is called with the id for a record of this view.

This is in TypoScript syntax:

- `[globalVar = TSFE:id = id_of_page_with_lingle_view] && [globalVar = GP:my_table/uid > 0]`

You don't need to know this syntax.

SEO Dynamic Tag delivers this condition in the user interface like:

- `[globalVar = TSFE:id = id_of_page_with_single_view] && [globalVar = GP:my_table/uid > 0]`

You have to replace

- *xxx* with the id of your page with the single view plugin and
- *my\_table/uid* with the parameter of your URL.

BE AWARE:

The value in the user interface is

- `globalVar = TSFE:id = xxx] && [globalVar = GP:my_table/uid > 0`

Don't wrap the condition in square brackets! This is a technical need.

## Service in single views

SEO Dynamic Tag

- adds the canonical tag
- the metaTags
  - author
  - description
  - keywords and
- the title tag.

Add means override!

By default no parameter is respected.

See details at 'List views' on page 33 above.

If you like to allow a parameter for the canonical tag, please refer to "Allow a parameter" on page 32 above.

## Check it!

You can check your configuration with the TypoScript Object Browser.

Please select the page with your configuration in the page tree and then:

1. select the [TypoScript Object Browser]
2. Browse: [Setup]
3. Display constants: [Plain substitution (default)]<sup>1</sup>
4. Enable the SEO-Dynamic-Tag condition.
5. Analyse the content rendered by SEO Dynamic Tag:  
See page.49442.\*
6. Analyse the metaTags rendered by SEO Dynamic Tag:  
See page.meta.\*

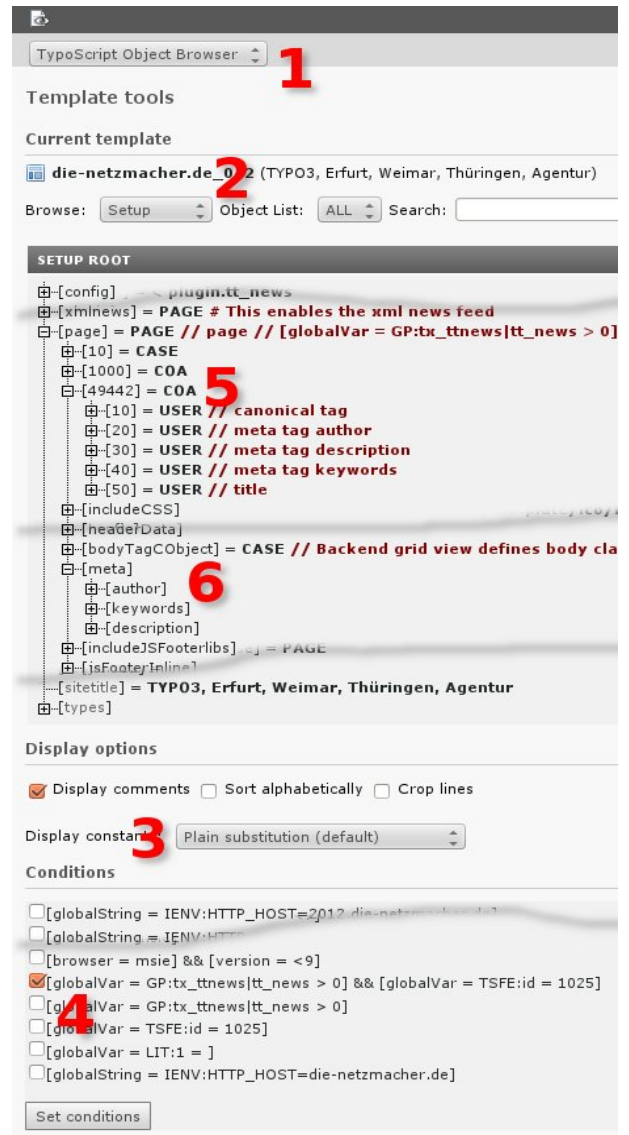


Illustration 34: Page properties by SEO Dynamic Tag

<sup>1</sup> Otherwise you won't see the SEO-Dynamic-Tag condition!

## Debug Mode

You enable the debugging with the user interface.

See Constant Editor category [SEO DYNAMIC TAG – DEBUGGING].

If you have enabled a debug option, you will get a report in the frontend at the bottom of the website.

## Options

All

- You will get a report for all options below.

Author

- Report for the metaTag author.

Canonical Tag

- Report for the canonical tag.

Description

- Report for the metaTag description.

Keywords

- Report for the metaTag keywords.

Title

- Report for the title tag.

## SQL Errors

SQL errors will displayed in the frontend for that tag, that has an enabled debugging option.

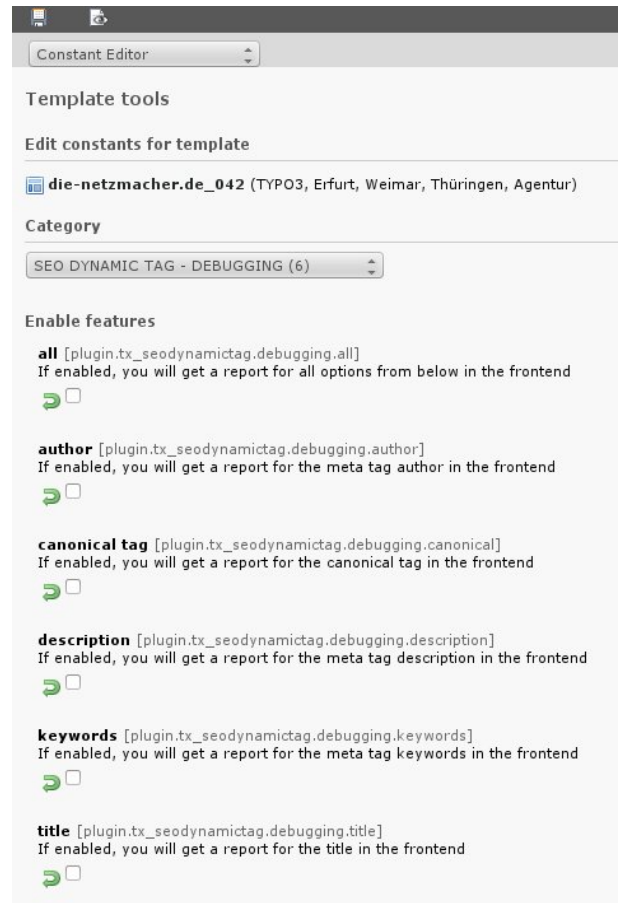


Illustration 35: User interface: Debugging

## Development

If you like to develop SEO Dynamic Tag or to integrate SEO Dynamic Tag into your extension, this is recommended:

Take a look in the ready-for-use templates. See

- static
  - cal
  - tt\_news
  - tt\_products

cal and tt\_news are simple: Only one table has to be handled

tt\_products is complex: Two tables with a relation have to be handled.

Take a look in the Organiser templates, look for "seodynamictag". The configuration is very small. See the extension [Organiser \(org\)](#)

- static
  - calendar
  - department
  - downloads
  - headquarters
  - location
  - news
  - staff

# Reference

## Canonical Tag

plugin.tx\_seodynamictag\_pi1.page.49442.10

BE AWARE: SEO Dynamic Tag copies the setup into the page object. If you like to change any property in the setup, you must configure the copy in the page object at page.49442.\* or page.meta.\* See "Check it!" on page 35 above.

Property:	Data type:	Description:	Default:
special	string	Internal. Don't touch it!	canonical
enabled	boolean	Enable or disable the canonical tag. Please configure the property with the Constant Editor. <b>Example:</b> <pre>plugin.tx_seodynamictag_pi1.page.49442.10 {     enabled = 1 }</pre>	1
host	string	Host for the absolute URL. Leave it empty, if the default host of the frontend should be used. Please configure the property with the Constant Editor. <b>Example:</b> <pre>plugin.tx_seodynamictag_pi1.page.49442.10 {     host = http://easy-typo3.com/ }</pre>	
path	COA	Configuration for rendering the canonical tag. Please configure the property with the Constant Editor. <b>Example:</b> <pre>plugin.tx_seodynamictag_pi1.page.49442.10 {     path = TEXT     path {         typolink {             parameter {                 data = page:uid             }             additionalParams = {\$plugin.tx_seodynamictag.canonical.single.additionalParams}             additionalParams {                 insertData = 1             }             //addQueryString = 1             //addQueryString {             //    method = GET             //    exclude = L,id,cHash,no_cache,tx_ttnews             //5BbackPid%5D             //}             useCacheHash = {\$plugin.tx_seodynamictag.canonical.useCashHash}             returnLast = url         }     } }</pre>	
debug	boolean	Debug the canonical tag. Please configure the property with the Constant Editor. <b>Example:</b> <pre>plugin.tx_seodynamictag_pi1.page.49442.10 {     debug = 1 }</pre>	0

## MetaTag Author

plugin.tx\_seodynamictag\_pi1.page.49442.20

BE AWARE: SEO Dynamic Tag copies the setup into the page object. If you like to change any property in the setup, you must configure the copy in the page object at page.49442.\* or page.meta.\* See "Check it!" on page 35 above.

Property:	Data type:	Description:	Default:
special	string	Internal. Don't touch it!	author
register	string	SEO Dynmaic Tag stores the author value in this register. It calls that value in the page.meta.author property. <b>Default:</b> tx_seodynamictag_author	See description
query	array	See "Query" on page 40 below	
debug	boolean	Debug this tag. Please configure the property with the Constant Editor. <b>Example:</b> <pre>plugin.tx_seodynamictag_pi1.page.49442.20 {     debug = 1 }</pre>	0

## MetaTag Description

plugin.tx\_seodynamictag\_pi1.page.49442.30

See MetaTag Author above.

BE AWARE: SEO Dynamic Tag copies the setup into the page object. If you like to change any property in the setup, you must configure the copy in the page object at page.49442.\* or page.meta.\* See "Check it!" on page 35 above.

## MetaTag Keywords

plugin.tx\_seodynamictag\_pi1.page.49442.40

See MetaTag Author above.

BE AWARE: SEO Dynamic Tag copies the setup into the page object. If you like to change any property in the setup, you must configure the copy in the page object at page.49442.\* or page.meta.\* See "Check it!" on page 35 above.

Property:	Data type:	Description:	Default:
keywords	string/array	Array for controlling the handling of keywords. Please configure the property with the Constant Editor.	
all other properties		See MetaTag Author above.	

## Title

plugin.tx\_seodynamictag\_pi1.page.49442.50

See MetaTag Author above. Title hasn't any register property.

BE AWARE: SEO Dynamic Tag copies the setup into the page object. If you like to change any property in the setup, you must configure the copy in the page object at page.49442.\* or page.meta.\* See "Check it!" on page 35 above.

## Query

plugin.tx\_seodynamictag\_pi1.page.49442.[10 - 50]

BE AWARE: SEO Dynamic Tag copies the setup into the page object. If you like to change any property in the setup, you must configure the copy in the page object at page.49442.\* or page.meta.\* See "Check it!" on page 35 above.

Property:	Data type:	Description:	Default:
select	string	<p>SQL select command.</p> <p>Please configure the property with the Constant Editor.</p> <p><b>Example 1:</b></p> <pre>plugin.tx_seodynamictag_pi1.page.49442.[10-50] {     select = CONCAT(`title`, ': ', `short`)     from   = `tt_news`     where  = `uid` = \$1     var.1  = tx_ttnews[tt_news] }</pre> <p><b>Example 2:</b></p> <pre>plugin.tx_seodynamictag_pi1.page.49442.[10-50] {     select = CONCAT(product.title, ' \     (' , category.title, ')') as value     from   = tt_products as product, \     tt_products_cat as category     where  = product.uid = \$1 \     AND product.category = category.uid \     AND product.deleted = 0 \     AND product.hidden = 0     var.1  = tx_ttproducts_pi1[product] }</pre>	
from	string	SQL table name. See example "select" above.	
where	string	SQL where expression. See example "select" above.	
var	array	<p>Variable in a SQL query.</p> <p>You can use a variable with any name in your SQL query everywhere.</p> <ul style="list-style-type: none"> <li>You have to label the variable with a dollar character (\$).</li> <li>You have to define the array var with an element named with your variable.</li> <li>Then you have to allocate the global get variable or the global post variable.</li> </ul> <p>If you need a post variable see property "method" below.</p> <p>See the example in the tutorial 'Error: Reference source not found' on page Error: Reference source not found.</p> <p><b>Example:</b></p> <pre>plugin.tx_seodynamictag_pi1.page.49442.[10-50] {     ...     where          = `uid` = \$my_var AND \     `pid` = \$my_other_var     var.my_var      = tx_ttnews[tt_news]     var.my_other_var = tx_ttnews[pid] }</pre> <p><i>It's only an example. There is no global var tx_ttnews[pid]</i></p>	
method	string: <i>get, post</i>	<p>You can choose the global array GET or POST for substituting variables in your SQL query.</p> <p><b>Example:</b></p> <pre>plugin.tx_seodynamictag_pi1.page.49442.[10-50] {     method = post }</pre>	get
maxLength	integer	<p>If set, the result will be cut after maxLength chars.</p> <p><b>Example:</b></p> <pre>plugin.tx_seodynamictag_pi1.page.49442.[10-50] {     maxLength = 100 }</pre>	100
dontStripTags	boolean	<p>If set, the result value won't be stripped of it's HTML-tags.</p> <p><b>Example:</b></p> <pre>plugin.tx_seodynamictag_pi1.page.49442.[10-50] {     dontStripTags = 1 }</pre>	0



# What's new?

## In version 2.x

### Canonical tag

[2.0.0]

SEO Dynamic Tag renders the canonical tag.

### Dynamic page object

[2.0.0]

The page object is changed dynamically.  
No manual configuration of TypoScript is needed any longer.

### User interface – the constant editor

[2.0.0]

All properties can set by the user interface – the Constant Editor.  
No manual configuration of TypoScript is needed any longer.

# Update

## From smaller than 2.x

If you update SEO Dynamic Tag from a version smaller than 2.x, please

- remove your configuration
- include the static template of SEO Dynamic Tag
- configure SEO Dynamic Tag with the user interface.

## FAQ

### SimulateStaticDocuments and RealUrl

The extension is compatible. In RealUrl you can't see the global get variables. The extension displays this variables in debug mode (see page 36).

## Helpful suggestions

### Forum

SEO Dynamic Tag 2 has a forum at:

- <http://typo3-browser-forum.de/#cat28>

Posts are welcome in English in German

### Support with costs

If you have a need for a support with costs, feel free to contact me: Dirk Wildt, <http://wildt.at.die-netzmacher.de>

## Further Information

### Other extensions published by Die Netzmacher

-  +AOE Linkhandler Configurator: Configure the AOE linkhandler supported by userinterfaces. Out-of-the-box templates for cal, org, tt\_news and tt\_products. Don't edit page TSconfig any longer.  
<http://typo3.org/extensions/repository/view/linkhandlerconf/>
-  autositemap: A smart site-map optimised for the footer. It groups menus in columns. Great menus will get two columns. Configuration is based on TypoScript HMENU.  
<http://typo3.org/extensions/repository/view/autositemap/>
-  Browser – TYPO3 without PHP. Develop your TYPO3 extension 8 times faster! You need one line TypoScript for a result list with a search form, a record browser and an index browser. Images are wrapped self-acting. SEO, Search Engine Optimization, is integrated. <http://typo3.org/extensions/repository/view/browser/>
-  Caddy – the TYPO3 shopping cart. You can use it for your own needs. You need a database with products or any other items only. You can install Caddy out of the box with one mouse click – see Quick Shop below.<sup>1</sup>  
<http://typo3.org/extensions/repository/view/caddy/>
-  Flip it! offers lovely and smooth page flip transitions. It enables you to run over pages in PDF documents like in a real magazine. It is based on flash. Flipt it! can convert PDF documents to swf files automatically.  
<http://typo3.org/extensions/repository/view/flipit/>
-  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 (see above) in three hours only.  
[http://typo3.org/extensions/repository/view/green\\_cars/](http://typo3.org/extensions/repository/view/green_cars/)
-  Jobmarket is a catalogue with job offers. Views, the a-z-browser, the page-browser, the search, social bookmarks and a lot of other stuff can configured by the Browser plugin with the mouse. Job Market hasn't any PHP code, it should be easy to adapt it to your needs.  
[http://typo3.org/extensions/repository/view/job\\_market/](http://typo3.org/extensions/repository/view/job_market/)
-  Majordomo: For subscribing to and unsubscribing from a majordomo mailing list.  
<http://typo3.org/extensions/repository/view/majordomo/>
-  Organiser – TYPO3 for the lobby and the organisers. Handle news, events, staff, headquarters, locations, workshops and a calendar with one extension. Sell online tickets! Install the Organiser with one mouse click!  
<http://typo3-organiser.de/>
-  Quick Shop – the fastest shop in the history of TYPO3. Install it with one mouse click! Quick Shop is based on the browser (see above) and powermail.  
<http://typo3-quick-shop.de/>
-  PDF Controller: Easy to install. Add to your HTML page the PDF-controller-button. Link from the button to the controller. Adjust the controller by mouseclicks. The PDF Controller supports CSS 3.  
<http://typo3-pdfcontroller.de/>
-  Route – Publish your routes with GoogleMaps or OpenStreetMap. Routes have points of interest (POI). You can categorise and filter both: routes and POI. Address data can geocoded automatically.<sup>2</sup>  
<http://typo3.org/extensions/repository/view/route/>
-  TSconfig Pages and Users by extManager (extkey: tsconf): Configure the the eight most commonly used TSconfig porperties with the mouse - like page tree uids, activated extended view, activated clipboard, ...  
<http://typo3.org/extensions/repository/view/tsconf/>
-  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.  
[http://typo3.org/extensions/repository/view/ttnews\\_selectconf/](http://typo3.org/extensions/repository/view/ttnews_selectconf/)
-  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 (see above).  
<http://typo3.org/extensions/repository/view/wine/>

1 I purpose to publish Caddy in the TYPO3 repository in September of 2013

2 I purpose to publish Route in the TYPO3 repository in September of 2013

## Credits

### State Development Corporation of Thuringia

The German State Development Corporation of Thuringia – German: Landesentwicklungs-Gesellschaft Thüringen – has contracted a part of the development.

- <http://www.leg-thueringen.de/>

## To-Do list

Nothing to do.



# Changelog

- 2.0.4 **Improvement**
  - \* #i0009: Manual
- 2.0.3 **Improvement**
  - \* #i0008: TYPO3 version in ext\_emconf.php
  - \* #i0007: Manual
  - \* #i0006: Labelling Constant Editor
- 2.0.2 **Improvement**
  - \* #i0005: mb\_strtolower
- 2.0.1 **Improvement**
  - \* #i0004: Control workflow of keyword handling by the Constant Editor
- 2.0.0 **Features**
  - \* #49442: Canonical tag
  - \* #i0004: Dynamic page object
  - \* #i0001: User interface: integration of the constant editor**Improvement**
  - \* #i0003: The labels of registers got the prefix tx\_seodynamic\_tag\_
- 1.1.1 **Bugfix**
  - \* #44545: Class 't3lib\_utility\_Debug' not found  
Downgrade for TYPO3 4.4**Improvement**
  - \* #00000: Update the manual
- 1.1.0 **Improvement**
  - \* #42405: t3lib\_div::view\_array is deprecated  
Update for TYPO3 4.7
- 1.0.0 **New Main Version**
  - Corporate Design Die Netzmacher GbR
- 0.0.5 **Improvement**
  - Documentation
- 0.0.3 **Improvement**
  - Keywords
- 0.0.2 **New Feature**
  - Keyword configuration
- 0.0.1 **Initial release**

# Illustration Index

Illustration 1: User interface: Canonical Tag.....	3
Illustration 2: User interface: Debugging.....	3
Illustration 3: SEO - Search Engine Optimization - with SEO Dynamic Tag.....	4
Illustration 4: Calendar Base (cal): initial situation.....	7
Illustration 5: Calendar Base (cal): include the static templates.....	7
Illustration 6: Calendar Base (cal): Configuration by the Constant Editor.....	8
Illustration 7: Calendar Base (cal): SEO - Search Engine Optimization - in the TYPO3 frontend.....	9
Illustration 8: News (tt_news): SEO tags.....	9
Illustration 9: News (tt_news): initial situation.....	10
Illustration 10: News (tt_news): include the static templates.....	10
Illustration 11: News (tt_news): Configuration by the Constant Editor.....	11
Illustration 12: News (tt_news): SEO - Search Engine Optimization - in the TYPO3 frontend.....	12
Illustration 13: News (tt_news): SEO tags.....	12
Illustration 14: Organiser (org): initial situation.....	13
Illustration 15: Organiser (org) News: Configuration by the Constant Editor.....	14
Illustration 16: Organsier (org) News: SEO - Search Engine Optimization - in the TYPO3 frontend.....	15
Illustration 17: Organsier (org) News: SEO tags.....	15
Illustration 18: Shop System (tt_products): initial situation.....	16
Illustration 19: Shop System (tt_products): include the static templates.....	16
Illustration 20: Shop System (tt_products): Configuration by the Constant Editor.....	17
Illustration 21: Shop System (tt_products): SEO - Search Engine Optimization.....	18
Illustration 22: Shop System (tt_products): SEO tags.....	18
Illustration 23: Any other database: initial situation.....	19
Illustration 24: any other database: include the static template.....	19
Illustration 25: Any other database: Configuration by the Constant Editor.....	20
Illustration 26: Any other database: conditions.....	21
Illustration 27: Any other database: database.....	22
Illustration 28: Any other database: default values.....	23
Illustration 29: Any other database: keywords.....	24
Illustration 30: Any other database: SEO - Search Engine Optimization.....	25
Illustration 31: Any other database: SEO tags.....	25
Illustration 32: Allowed parameter in the single view.....	32
Illustration 33: Conditions by SEO Dynamic Tag Constant Editor.....	33
Illustration 34: Page properties by SEO Dynamic Tag.....	35
Illustration 35: User interface: Debugging.....	36