

# DAT User Management by SOAP

Interface for authenticate, login and administer frontend users by the DAT SOAP server.  
It is a base for developing fe\_user authentication by other SOAP servers.



Version: 1.0.0, 2013-01-03

Extension Key: datusers - Tutorials for the extension Browser (extkey: browser)

Language: en

Keywords: feuser, felogin, feuser\_admin, soap, developers

Copyright 2012-2013, 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

DAT User Management by SOAP .....	1	SOAP test .....	11
Introduction .....	3	Reference .....	12
What does it do? .....	3	In General .....	12
Audience .....	3	tx_felogin_pi1.extensions.datusers .....	13
What do you need? .....	3	Workflow .....	14
Dependencies .....	3	Login .....	14
Conflicts .....	3	User Data Administration – My Data .....	14
Quick Installation .....	4	Development .....	15
Groups .....	5	Involved classes .....	15
Groups are required for a login .....	5	Helpfull links .....	15
Add some groups .....	5	Helpful suggestions .....	16
Add the Remove Group .....	5	Known Bugs .....	17
Pages .....	6	To-Do List .....	18
Login .....	6	Further Information .....	19
My Data .....	7	Other extensions published by Die Netzmacher .....	19
Debugging .....	11	Change Log .....	20
DRS – Development Reporting System .....	11	Illustration Index .....	21

# Introduction

## What does it do?

- datusers extends the TYPO3 frontend user administration.
- It enables to authenticate frontend users by the SOAP server of the DAT.
- It enables frontend users to administer their personal data on the SOAP server like the password and the e-mail address.
- datusers extends the TYPO3 system extension felogin. This has two big advantages:
  - You configure the login form and the login work flow on the known way – with the powerful plugin / flexform of the felogin extension.
  - But you can extend the forms of felogin with your own HTML markers. Each marker is a TypoScript content object. You will be very flexible to design your own form.
- datusers hasn't any unwanted effects to the TYPO3 frontend user administration:
  - If a frontend user can't authenticate by a SOAP server, TYPO3 tries to authenticate the frontend user by the common way: TYPO3 tries to find the user in the fe\_users table.
- datusers has the DRS – Development Reporting System – integrated. Developers are able to follow the work flow and are able to investigate the frontend user login and the frontend user administration.

## Audience

- This extension is for developers only!
- It is a base, to develop your own fe\_user authentication with any third party SOAP server.
- You have to adapt the PHP SOAP methods to your needs.
- If you have good ideas, please contact us. See "Helpfull links" on page 15 below. We would like to generalise DAT User Management by SOAP to Frontend User Management by SOAP.

## What do you need?

- A connection to a SOAP server.

## Dependencies

- datusers depends on felogin 1.4.0.

## Conflicts

- datusers isn't able to decrypt passwords.
- If you need a secure login and you like to use datusers, you have enable the login by the https protocol.
- If there will be any conflict with the system extensions "RSA authentication for TYPO3" (rsaauth) or "Salted user password hashes" (saltedpasswords), datusers will prompt a qualified message in the extension manager.

## Quick Installation

- Install this extension: datusers
- Go to the extension manager and call the category "Configuration Check" of this extension.
- If there are warnings or errors, please take care of a proper configuration. You have to configure at least
  - Access data of the SOAP administrator
  - The id of the page / sysfolder with the fe\_user data. The page / sysfolder has to be arranged before.
  - The id of the remove group. The remove group page / sysfolder has to be arranged before.  
See "Groups are required for a login" on page 5 below.
- Add the two pages
  - login
  - my data
- Add fe\_groups
- Add SOAP groups and attribute it to the SOAP groups.

# Groups

## Groups are required for a login

The SOAP group of a user has to match one TYPO3 fe\_group at least, otherwise login will failed.

## Add some groups

Add some groups to your folder with the fe\_user records.

## Add the Remove Group

Add a so called remove group to your folder. A user, which is attributed to this group, will removed after a defined time period like 2 hours automatically by datusers.

Please configure the extension manager of datusers: assign the ID of this group.

See "Quick Installation" on page 4 above.

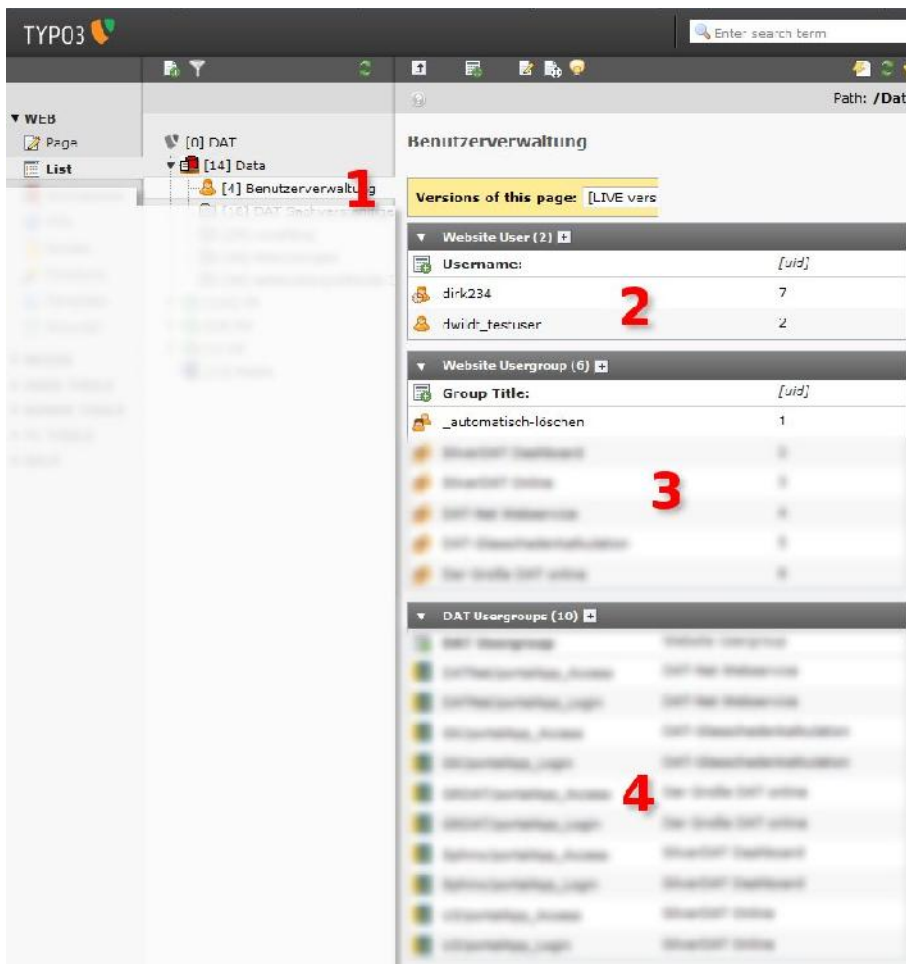


Illustration 1: Sysfolder with fe\_users, fe\_groups and DAT groups

1. Sysfolder for user and group records
2. fe\_user records, most imported from the SOAP server.
3. fe\_groups
4. DAT groups attributed to TYPO3 fe\_groups

## Pages

Please add the two pages

- login
- my data

## Login

- Add the page "Login"
- If you like the extension icon in the page tree, please edit
  - page properties > appearance > contains plugin: "DAT Users: FE Login"
- Set page properties > behaviour > Cache: "disable"

## TypoScript

- Add a TypoScript-Template to the page "Login"
- Call it "+page\_login\_xx", where xx is the Id of the TypoScript-Template
- Include the static template (from extension): "DAT users: felogin (datusers)"

## Login Form

- Add the content element with type "Login" to the page "Login"
- Call it "[Login]" and set the type to "hidden".
- Configure the plugin:
  - Set the user storage page to the sysfolder with the fe\_user data at least

## Illustration

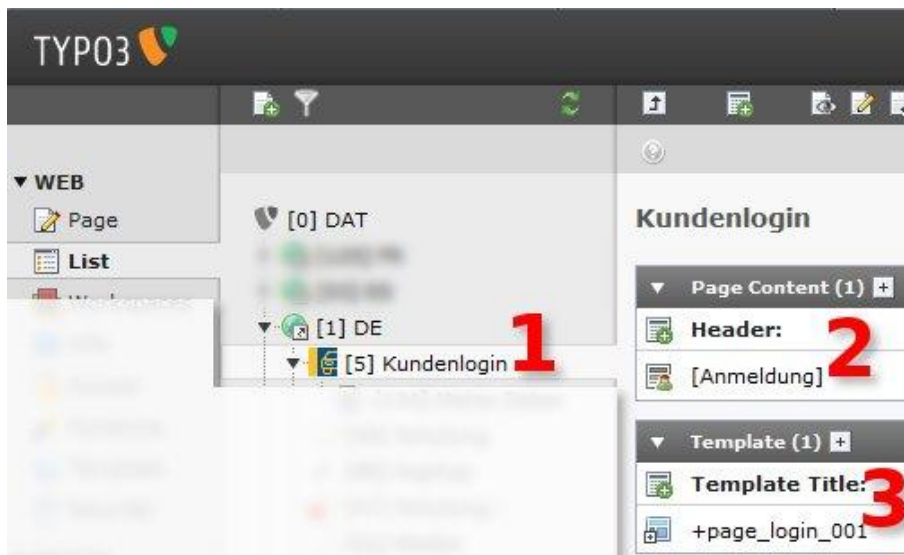


Illustration 2: Page "Login"

1. page "Login"
2. content element "Login"
3. TypoScript template

## My Data

- Add the page "My Data" maybe into the page "Login"
- Set page properties > behaviour > Cache: "disable"
- Set page properties > access > usergroup access rights: "Show at any login"

## Powermail

The form for the user data is configured by Powermail.

### Plugin

- Create a new content element: Powermail
- Call it "[Powermail: My Data]"
- General > type: "hidden"
- Formdata > Form title "My Data"
- Recipients > Subject for recipient's mail: "DAT user has update his data"
- Recipients > Email of recipient: the e-mail of the website administrator  
**A user can't update his data without any proper recipient e-mail!**
- Save the form
- Plugin > Fields > Create new fieldsets (see next section)

### Fieldset

- Plugin > Fields > Create new fieldsets
- Call it "My Data"
- Create new fields within the fieldset (see next section)

### Fields

- uid
  - fieldtype: "Hidden Field"
- last name
  - fieldtype: "Textfield", Read only "enabled"
- first name
  - fieldtype: "Textfield", Mandatory field "enabled"
- e-mail
  - fieldtype: "Textfield", Mandatory field "enabled", Validate this field for "Email-address"
- username
  - fieldtype: "Textfield", Mandatory field "enabled"
- telephone
  - fieldtype: "Textfield"
- fax
  - fieldtype: "Textfield"
- check data
  - fieldtype: "Submit button", Mandatory field "enabled"

## DAT Users

The fields of the Powermail form should be prefilled with values from the fe\_user record. The DAT Users plugin provides all values from this record in the session data. And Powermail will take this data from the session.

### Plugin

- Create a new content element > plugin "DAT users for Powermail"
- Call it "[DAT Users for Powermail]"
- Plugin > Powermail > Uid of the Powermail form: XX, where XX is the ID of the Powermail form (see above)
- Plugin > Powermail > Powermail form has a confirmation page: enabled
- Save and close the plugin.
- **The DAT plugin must be arranged ahead of the Powermail plugin!** If not, please move the plugin ahead of the Powermail element.

## TypoScript

- Add a TypoScript-Template to the page "My Data"
- Call it "+page\_mydata\_xx", where xx is the Id of the TypoScript-Template

### Constants

Copy and paste the snippet below into the field Constants.

```
// Set by TypoScript Constants Editor
plugin.tx_datusers_pi1 {
    fe_users {
        pid = xx
    }
    powermail {
        uid = yy
    }
}
```

- Replace xx with the id of the page / sysfolder, which contains the fe\_user data.
- Replace yy with the id of the Powermail plugin



## Setup

Copy and paste the snippet below into the field Constants.

```
plugin.tx_powermail_pil {
    dbEntry {
        fe_users {
            // Enable the editing of fe_users
            _enable = TEXT
            _enable {
                value = 1
            }
            // Update fe_users record with the given uid
            _ifUnique {
                uid = update
            }
            // Id of the sysfolder with the fe_users records
            pid = TEXT
            pid {
                value = {$plugin.tx_datusers_pil.fe_users.pid}
            }
            // Uid of the current fe_user
            uid = TEXT
            uid {
                data = TSFE:fe_user|sesData|powermail_{$plugin.tx_datusers_pil.powermail.uid}|uid18
            }
            first_name = TEXT
            first_name {
                data = TSFE:fe_user|sesData|powermail_{$plugin.tx_datusers_pil.powermail.uid}|uid20
            }
            email = TEXT
            email {
                data = TSFE:fe_user|sesData|powermail_{$plugin.tx_datusers_pil.powermail.uid}|uid21
            }
            username = TEXT
            username {
                data = TSFE:fe_user|sesData|powermail_{$plugin.tx_datusers_pil.powermail.uid}|uid22
            }
            telephone = TEXT
            telephone {
                data = TSFE:fe_user|sesData|powermail_{$plugin.tx_datusers_pil.powermail.uid}|uid23
            }
            fax = TEXT
            fax {
                data = TSFE:fe_user|sesData|powermail_{$plugin.tx_datusers_pil.powermail.uid}|uid24
            }
        }
    }
    prefill {
        // uid
        uid18 = TEXT
        uid18 {
            data = TSFE:fe_user|sesData|datusers|fe_users.uid
        }
        // last_name
        uid19 = TEXT
        uid19 {
            data = TSFE:fe_user|sesData|datusers|fe_users.last_name
        }
        // first_name
        uid20 = TEXT
        uid20 {
            data = TSFE:fe_user|sesData|datusers|fe_users.first_name
        }
        // email
        uid21 = TEXT
        uid21 {
            data = TSFE:fe_user|sesData|datusers|fe_users.email
        }
        // username
        uid22 = TEXT
        uid22 {
            data = TSFE:fe_user|sesData|datusers|fe_users.username
        }
        // telephone
        uid23 = TEXT
        uid23 {
            data = TSFE:fe_user|sesData|datusers|fe_users.telephone
        }
    }
}
```

```
// fax
uid24 = TEXT
uid24 {
    data = TSFE:fe_user|sesData|datusers|fe_users.fax
}
}
```

- Replace all xx with the ids of the Powermail fields
- Include the static template (from extension): "Powermail (powermail)"
- If you like some CSS effects add "Powermail fancy CSS (powermail)" too.

## Illustration

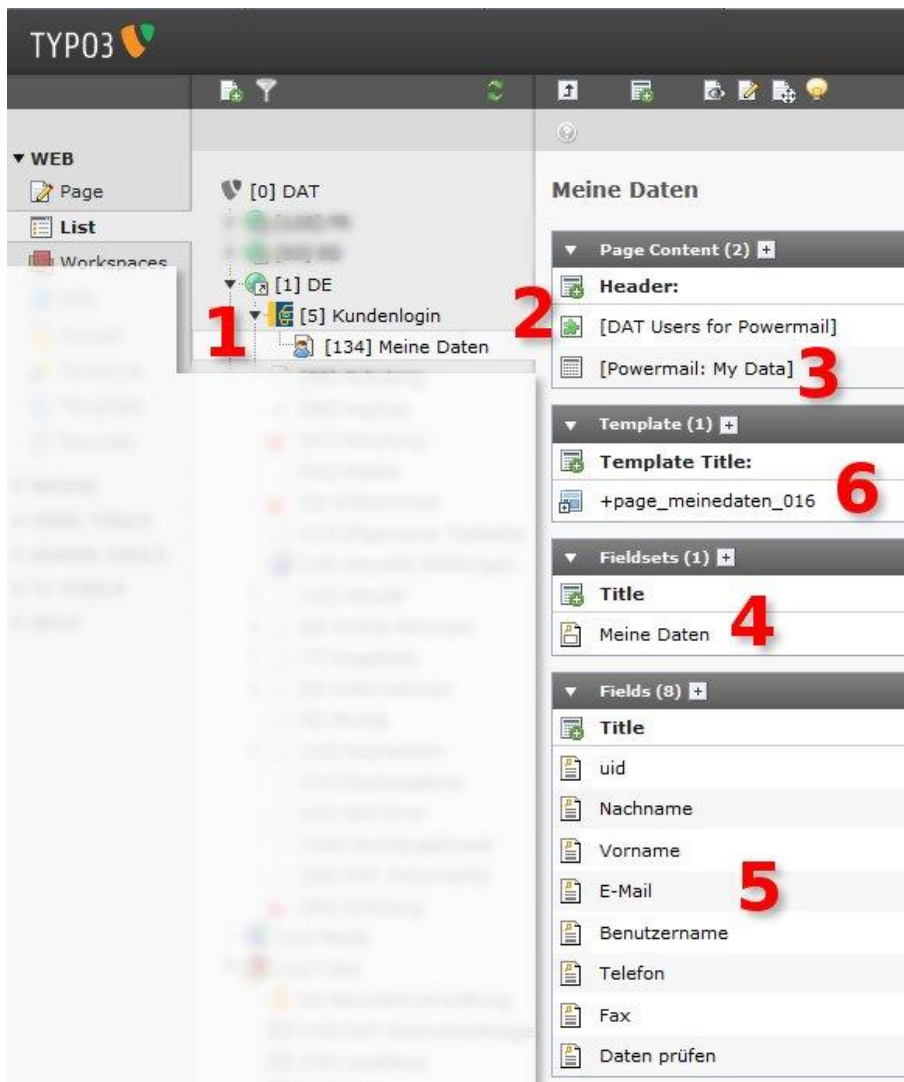


Illustration 3: Page "My Data"

1. Page "My Data"
2. Plugin: DAT Users for Powermail. **It must be ahead of the Powermail plugin!**
3. Plugin: Powermail
4. Powermail fieldset
5. Powermail fields
6. TypoScript Template

# Debugging

You enable the both Debugging Tools in the extension manager of datusers.

## DRS – Development Reporting System

The DRS – Development Reporting System – prompts the whole workflow into the devlog. You can investigate, if a user is authenticated or not and why.

## SOAP test

SOAP test enables to check datusers without any available SOAP server.

If the SOAP test is enabled, you can login with the access data of the test user. The user won't authenticated with a SOAP request but with the access data configured in the extension manager of datusers.

If SOAP test is enabled some relevant information will prompt into the frontend.

If you enable the DRS, you will get a lot of more information in the backend.

## Enable your IP

You have to add your IP to the list of allowed IPs in the extension manager of datusers.

## SOAP test is exclusive for the allowed IPs

The SOAP test is exclusive for clients, which IPs are part of the list of allowed IPs.

This enables you to test datusers with an operating system.

# Reference

## In General

datusers includes itself into the work-flow (by hooks) and into the TypoScript of foreign extensions (here: felogin).

The TypoScript has the same principle independent of the foreign extensions.

datusers will extend foreign extensions with this TypoScript skeleton:

```
extensions {
    datusers {
        ...
    }
}
```

See the snippets below.

## TypoScript

### Principle

```
plugin.foreign_extension {
    extensions {
        datusers {
            ... <- Additional properties by datusers
        }
    }
}
```

### Example

datusers extends felogin with the property, to remove left over markers.

```
plugin.tx_felogin_p11 {
    extensions {
        datusers {
            workflow {
                // [Boolean] Remove left over markers
                removeEmptyMarker = 1
            }
        }
    }
}
```

## tx\_felogin\_pi1.extensions.datusers

Additional properties for the first plugin (login) of the system extension felogin.

Property:	Data type:	Description:	Default:
forms.marker	array	<p>An array with markers for the forms of the felogin extension:</p> <ul style="list-style-type: none"> <li>Each marker is a cObj – a content object like TEXT and COA among others.</li> <li>Each TypoScript marker has to correspondend with an HTML marker like: clientno -&gt; ###CLIENTNO###</li> </ul> <p>Feel free to define any marker.</p> <p><b>Example:</b></p> <pre> plugin.tx_felogin_pi1 {     extensions {         datusers {             forms {                 marker {                     clientno = TEXT                     clientno {                         value = Client no                         lang {                             de = Kundennummer                             en = Client no                         }                     }                     wrap = &lt;label&gt; :&lt;/label&gt;                 }             }         }     } } </pre>	
workflow.removeEmptyMarker	boolean	<p>Left over markers in HTML templates will removed. Disable it, if you have unexpected results like missing marker in HTML templates.</p> <pre> plugin.tx_felogin_pi1 {     extensions {         datusers {             workflow {                 removeEmptyMarker = 1             }         }     } } </pre>	1

# Workflow

## Login

- The page login contains the login form of the extension fe\_login
- datusers replace markers by a hook class
  - The marker customer number is added
- The user adds his data like
  - customer number
  - username and
  - password
- fe\_login is looking for a TYPO3 service.
- datusers is registered as a TYPO3 service.
- fe\_login send the data to datusers.
- datusers sends the data to the SOAP server for authentication.
- If the SOAP server authenticates the user, datusers will return true to fe\_login and creates a session with all needed data.
- If fe\_login receives a true from datusers, it cancel the workflow.
- If fe\_login receives a false from datusers, it tries to authenticate the user supportde by the fe\_user database.
- It will fails in any case, because the fe\_user database doesn't contain customer numbers.

## User Data Administration – My Data

- datusers manage the session data for the user data administration.
- It reads all data from the fe\_user record.
- Powermail is able, to read this session data and pre fill any Powermail form with this data. A TypoScript configuration is needed.
- Powermail takes care of updating the data in the fe\_user record.
- datusers reads the data from the updated fe\_user record, writes it into the session and sends it to the SOAP server.

# Development

## Involved classes

- `t3lib/class.t3lib_svbase.php`
- `t3lib/class.t3lib_userauth.php`
- `typo3/sysex/tx_sv_auth.php`
- `typo3/sysex/tx_sv_authbase.php`

## svbase

## Helpfull links

### TYPO3

- [http://typo3.org/extension-manuals/cc\\_sv\\_auth/1.2.3/view/toc/0/](http://typo3.org/extension-manuals/cc_sv_auth/1.2.3/view/toc/0/)

### User Auth

German

- <http://www.marc-willmann.de/typo3-tutorials/anleitungen/frontend-user-login-gegen-eigene-datenquelle/>
- <http://jimsuperfly.de/blog/typo3-auth-service/>

### SOAP

German

- <http://phpforum.de/forum/showthread.php?t=257851>

## Helpful suggestions

If you have helpful suggestions, feel free to send a question, a bug or a code snippet by e-mail. See <http://wildt.at.die-netzmacher.de>.

E-mails are welcome in English and German.



## Known Bugs

Nothing known.

## To-Do List

Nothing to do.

## Further Information

### Other extensions published by Die Netzmacher

-  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 1 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/>
-  Flip it! Flip it! enables you to run over pages in PDF documents and magazines. Flip it! offers lovely and smooth page flip transitions. The visitor of your website does not need any PDF plugin but a Flash plugin.  
<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 in three hours only.  
<http://gruene-autos.org/>
-  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-job-market.de/>
-  Majordomo: For subscribing to and unsubscribing from a majordomo mailing list.  
<http://typo3.org/extensions/repository/view/majordomo/>
-  Organiser provides a lot of features for handle news, events, staff, headquarters, locations, repertoire, workshops and a calendar. Sell your online tickets! Available from April of 2011  
<http://typo3-organiser.de/>
-  Quick Shop (extkey: quick\_shop): The fastest shop in the history of TYPO3. Install it with one mouse click only! Quick Shop is based on the Browser 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/>
-  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/](http://typo3.org/extensions/repository/view/seo_dynamic_tag/)
-  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. 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/](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.  
<http://typo3-wine.de/>

# Change Log

## 1.0.0 Initial TER Release

### 0.1.1 Manual

- \* #00000: Adding the manual
- Development
- \* #00000: defaultdatuser is removed
  - /lib/extmanager/class.tx\_datusers\_extmanager.php
  - /ext\_conf\_template.txt
  - /lib/locallang.xml

### Bugfixing

- \* #00000: /lib/locallang.xml

### 0.1.0 Development

- \* #00000: extKey is moved from feuserssoap to datusers
- \* #00000: Extension manager checks a lot of properties
- \* #00000: Integration of SOAP data exchange
- \* #00000: HTML template got the marker
  - \* ###DATUSERS\_STATUS\_MESSAGE###
  - \* ###DATUSERS\_SOAPTEST\_MESSAGE###
  - \* #00000: SOAP qualifies error and
    - \* prompts it to the DRS
    - \* and sets session data
    - \* datusers\_status\_message
    - \* datusers\_soaptest\_message
- \* #00000: Session data will reset by the hook class methods at the end of the workflow

### 0.0.4 Development

- \* #00000: extKey is moved from datusers to feuserssoap

### 0.0.3 Development

- \* #00000: Move from plugin to service

### 0.0.2 Development

- \* #00000: Adding the service skeleton

### 0.0.1 Initial release

extKey: datusers  
 Dirk Wildt <http://wildt.at.die-netzmacher.de>

# Illustration Index

Illustration 1: Sysfolder with fe_users, fe_groups and DAT groups.....	5
Illustration 2: Page "Login".....	6
Illustration 3: Page "My Data".....	10