Registration for CAL-events

Extension Key: register4cal

Language: en

Keywords: keywords comma-separated

Copyright 2009, Thomas Ernst, <typo3@thernst.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

Table of Contents

Introduction

What does it do?

You use the cal extension to display events for your registered users. The register4cal extension now allows registered users to register as attendees for selected events. The extension register4cal comes with the following features:

- Additional fields for the registration form, customizable via TypoScript

- Users, displaying an event after having registrated for this event, get a message that they have already registered

- Frontend-users can be linked with event organizers. These frontend-users can then see a list of participants for their events.

- Each frontend-users can see a list of events for which he has registered

- When a user registers for an event, he can receive a confirmation email.

- When a user registers for an event, the organizer can receive a notification email

- Registration for single events is possible from the event single view

- Registration for multiple events is possible from the event list view

- Limited number of attendees

- Waitlist feature

Screenshots

Image 1: Event displayed with registration form

Image 2: Registration list for organizer

October 13 - November 13

1 entries found

Register for displayed events

[Testtermin](#)
October 15 07:00 am - 07:30 am

☐ Register for this event

No of participants [1]

I attend the event [Yes ⇕]

Register for displayed events

Image 4: Registration in event list view

Languages

This manual is written in english. A german version is also available. The extension contains localizations for german, english and danish. Currently the danish translation has been completed using an online translation tool. These unchecked translated texts are in brackets.

Users manual

The register4cal extension works closely with the cal extension. Thus, most settings can be found via extended cal objects. If you are looking information on how to create an cal event, please refer to the manual of the cal extension.

Activate the registration feature for an event

To activate the registration feature you need to open the calendar event record. The extension register4cal adds a new tab "Registrations for event", containing all relevant fields.

Image 5: Fields related to registration in the calendar event record

To activate the registration feature, check the field "Activate registration". If you want, you can set a start and an end date for the registration. If you leave these fields empty, the registration period starts immediately and ends at the start day of the event. In addition, you can limit the maximum number of attendees and activate the waitlist feature for this event.

Allow frontend users to see registrations for "their" events

If frontend users should be able to see the registrations for "their" events, there are to things to set up.

First, you need to use an organizer via an "calendar event organizer" record, not an freely typed in organizer.

Image 6: Use an cal organizer in the event record

Second, the extension needs to know, which frontend user(s) are linked with this organizer. This can be set via an additional field in the cal organizer record. At the bottom of the fields for the calendar event organizer is another additional field. You can select the frontend users linked to this organizer here:

Image 7: Link frontend users with the organizer in the organizer record

If you want the organizer to receive emails when someone registers for his event, make sure you have entered an email address for the organizer.

Thats everything needed to activate the registration feature for an event and have the organizer see the participants in the frontend.

Administration

Installation notes

To install the extension, perform the following steps:

1. Make sure the extension "cal" is installed. If it isn't, install it or you can't use the register4cal extension.

2. Import the register4cal extension and install it.

3. Confirm the requested DB modifications. The extension will add new fields to two tables of the cal extension. It also comes with one own table.

4. Insert the static template from the extension into your main template.

5. Adapt the typoscript settings to your needs (See section "Configuration" for details)

You do not need to create any sysfolders as registrations will automatically be stored in the same folder as event.

Displaying registration forms

Registrations for single events

The registration form is automatically displayed, if the following conditions apply:

*   The single event view of cal is being displayed.

*   An event is displayed in this view.

*   A fe-user is logged in.

*   The registration for the displayed event has been activated.

*   The current date is in the registration period (If no registration period is given, the start of the event is the end of the registration period.)

So activate the registration feature for an event, login as fe-user and display this event in your frontend. You are able to register for it.

Registrations for multiple events

The registration for mutliple events requires additional changes in your cal templates.

*   In your list-template (default: EXT:cal/template/list.tmpl) add the marker "###MODULE__tx_register4cal_submit###" (care about upper/lower case and the double underscore after "MODULE"!) in subpart "###LIST_TEMPLATE###" where the submit-button should be displayed if registration fields are being displayed.

```
<!-- ###LIST_TEMPLATE### begin -->

<!-- ###PAGEBROWSER### begin -->
<p class="pagebrowser">###PAGEOF###  ###PAGES###  ###PREVIOUS### ###NEXT###</p>
<!-- ###PAGEBROWSER### end -->

<h1>###HEADING###</h1>
###FOUND###
<!-- ###PRE_LIST_TEMPLATE### begin -->
###MODULE__tx_register4cal_submit###
<dl>
<!-- ###PRE_LIST_TEMPLATE### end -->
        <!-- ###LIST### start -->
        <!-- ###LIST### end -->
<!-- ###POST_LIST_TEMPLATE### begin -->
</dl>
###MODULE__tx_register4cal_submit###
<!-- ###POST_LIST_TEMPLATE### end -->

<!-- ###PAGEBROWSER### begin -->
<p class="pagebrowser">###PAGEOF###  ###PAGES###  ###PREVIOUS### ###NEXT###</p>
<!-- ###PAGEBROWSER### end -->

###CREATE_EVENT_LINK###
<!-- ###LIST_TEMPLATE### end -->
```

Image 8: Example of list template with submit buttons

- In your event template (default: EXT:cal/template/phpicalendar_event.tmpl, from cal 1.3.0: EXT:cal/template/event_model.tmpl) add the marker "###MODULE__tx_register4cal_listreg###" (care about upper/lower case and the double underscore after "MODULE"!) in subparts " ###TEMPLATE_PHPICALENDAR_EVENT_LIST_ODD###" and "###TEMPLATE_PHPICALENDAR_EVENT_LIST_EVEN###" where the registration fields for an event should be displayed if registration for the event is enabled.

```
<!-- ###TEMPLATE_PHPICALENDAR_EVENT_LIST_ODD### start -->
        <dt>
            <!-- ###EVENT_LINK### start-->
                    ###IMAGE### ###TITLE###
            <!-- ###EVENT_LINK### end--><br />
            ###STARTDATE### ###ENDDATE###
            <span class="V9G">###STARTTIME### ###ENDTIME###</span><br />
            ###MODULE__tx_register4cal_listreg###
            <br /><span class="V9G">###CATEGORY_LINK###</span>
        </dt>
        <dd>###DESCRIPTION###</dd>
<!-- ###TEMPLATE_PHPICALENDAR_EVENT_LIST_ODD### end -->
<!-- ###TEMPLATE_PHPICALENDAR_EVENT_LIST_EVEN### start -->
        <dt class="even">
            <!-- ###EVENT_LINK### start-->
                    ###IMAGE### ###TITLE###
            <!-- ###EVENT_LINK### end--><br />
            ###STARTDATE### ###ENDDATE###
            <span class="V9G">###STARTTIME### ###ENDTIME###</span><br />
            ###MODULE__tx_register4cal_listreg###
            <br /><span class="V9G">###CATEGORY_LINK###</span>
        </dt>
        <dd class="even">###DESCRIPTION###</dd>
<!-- ###TEMPLATE_PHPICALENDAR_EVENT_LIST_EVEN### end -->
```

Image 9: Example of event template with registration fields

- Now registration from the list view is possible as well.

Display registrations

To display the registrations either of the frontend user or for the events where the frontend user is the organizer, you need to add plugins, probably on separate pages.

Configure these content elements as folowing:
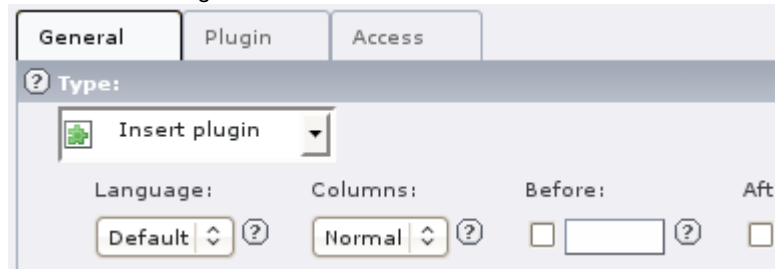
Image 10: Insert a plugin

Go to the "Plugin"-Tab and select the "Registrations for TYPO3-Calendar" plugin. Select the "Registrations of the frontend user" or the "Registrations for events of the frontend user" to be displayed:
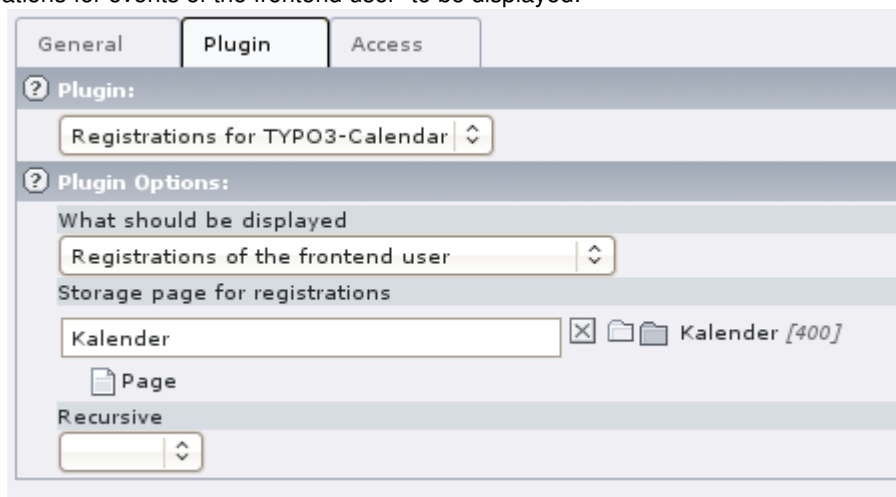

Image 11: Settings for the plugin

This time, you need to set the storage page for registrations. Set it to the sysfolder where the calendar events are stored for which you want to display the registrations.

Configuration

The complete configuration of the extension can be done via TypoScript. This is the basic structure of the TypoScript:

```
#Setup
plugin.tx_register4cal_pi1{
    # -> Some General settings for the extension

    emails {
        # -> General settings for sending emails
    }
    forms {
        default {
            # -> Default settings for all views
        }
        single {
            # -> Settings for forms used for single event view
        }
        list {
            # -> Settings for forms used for event list view
        }
        email {
            # -> Settings for forms used for sending emails
        }
        listOutput {
            # -> Settings for forms used for list plugins
        }
    }
    userfields {
        # -> Settings for userdefined fields in registration forms
    }
    edit {
        # -> Settings for frontend editing
    }
    view {
        # -> General settings for display functions
    }
}
```

The following reference explains the parts

Reference

General Settings

Property:	Data type:	Description:	Default:
template	string	Filename of the template file to use	EXT:register4cal/templates/register.tmpl
dateFormat	string	Format to display dates. Pear Syntax.	%d.%m.%Y
timeFormat	string	Format to display times. Pear Syntax.	%H:%M
disableNeedLoginForm	boolean	Disables the "Need to login"-Form, which informs the user, that he needs to login before he can register.	1
disableWaitlist	boolean	Completely deactivates waitlist	0
disableUnregister	boolean	Deactivates the possibility to unregister from events	0
keepUnregistered	integer	Defines if registration records should be kept after an user unregsters or not. If the records are being kept, they will be displayedin registrationlists with status "unregistered". The following values can be used: <table><tr><td>0</td><td>No registration records will be kept after unregistering</td></tr><tr><td>1</td><td>Registration records will be kept after unregistering</td></tr><tr><td>2</td><td>Registration records will be kept after unregistering but will be deleted once the users registers again for the event</td></tr></table>	1

Property:	Data type:	Description:	Default:
onetimepid	integer	Page id of page containing the onetimeaccount-Plugin	0
onetimereturnparam	string	Name of the url-parameter which is needed to provide the onetimeaccount extension with the url it should return to.	redirect_url
loginpid	integer	Page id of page containing the login form	0
loginreturnparam	string	Name of the url-parameter which is needed to provide to provide the loginform with the url it should return to afterwards	redirect_url

Settings for sending emails when an user has registered

When an user has registered, two emails can be sent. One can confirm the registration to the user, the second can inform the organizer about the registration. The settings for these emails can be found in the "emails" section. These section has the following properties:

Property:	Data type:	Description:	Default:
sendConfirmationMail	boolean	When set to 0, the confirmation email is not being sent.	1
sendNotificationMail	boolean	Wen set to 0, the notification is not being sent.	1
senderName	string	Name of the sender of the email.	Email sender
senderAddress	string	Email address of the sender of the email	Some@mail.address
adminEmail	string	Email address of a registration admin, all notification emails are sent to this address, too. Leave empty if you do not want to use this feature	

Defining userfields:

Userspecific fields can be defined in the "userfields" section. Every section below it defines one userfield. Theoretically the number of userfields is unlimited, while you can an unlimited count of sections below userfields. For each field you have the following properties:

Property:	Data type:	Description:	Default:
name	string	Name of the field	
caption.default	string	Caption for the field in default language	
caption.[language]	string	Caption for the field in another language. [language] is taken from config.language.	
default	string	default value for the field	
options	structure	Definition of options for select-boxes or option lists. See example below for further information on this part.	
layout.show	stdWrap	Definition of the layout to display the value of the field	
layout.edit	stdWrap	Definition of the layout for the entry of the value of the field	

Example definition for userfields:

```
#Setup
plugin.tx_register4cal_pi1.userfields {
        # Example for simple input fields
        0 {
            name=participants
            caption.default=No of participants
            caption.de=Anzahl Teilnehmer
            default = 1
            layout {
             edit = stdWrap
             edit.dataWrap = <tr><td><b>###CAPTION###</b></td><td><input type="text" size="20"
name="###NAME###" value="|" /></td></tr>
             show = stdWrap
             show.dataWrap = <tr><td><b>###CAPTION###</b></td><td>|</td></tr>
            }
        }

        # Example for text area
        1 {
            name=Comments
            caption.default=Comments
            caption.de=Kommentar
            default =
            layout {
             edit = stdWrap
             edit.dataWrap = <tr><td><b>###CAPTION###</b></td><td><textarea rows="10" cols="40"
name="###NAME###">|</textarea></td></tr>
             show = stdWrap
             show.dataWrap = <tr><td><b>###CAPTION###</b></td><td><textarea rows="10" cols="40"
readonly="yes">|</textarea></td></tr>
            }
        }

        # Example for selection
        2 {
            name=Selection
            caption.default=I attend the event
            caption.de=Ich nehme teil
            default =
            options {
             1.default = Yes
             1.de = Ja
             2.default = No
             2.de = Nein
             3.default = Yes, but I'll be late
             3.de = Ja, aber ich komme später
            }
            layout {
             edit = stdWrap
             edit.dataWrap = <tr><td><b>###CAPTION###</b></td><td><SELECT SIZE=1
name="###NAME###">###OPTIONS###</SELECT></td></tr>
             show = stdWrap
             show.dataWrap = <tr><td><b>###CAPTION###</b></td><td>|</td></tr>
            }
        }
}
```

Settings for frontend editing

There are two properties in the edit-section:

Property:	Data type:	Description:	Default:
useDateSelector	boolean	Use the dateselector provided by extension rlmp_dateselectlib. If set to 1 this extension is required.	1
CalConf	structure	Configuration for the dateselector. Refer to the manual of extension rlmp_dateselectlib for further information.	

Settings for the display of registrations

There are two more properties in the view-section:

Property:	Data type:	Description:	Default:
eventViewPid	integer	Pid of the page, containing the single event view. Used for the link from the registration to the event record	0
adminUsers	string	Comma separated list of frontend user id's of frontend users who should be able to see all registrations, not only the ones for their events.	

Frontend Editing

The extension register4cal supports the use of the additional event fields with the frontend editing feature of the cal extension. The register4cal extension includes all necessary TypoScript settings for the cal extension to handle the additional fields.

IMPORTANT: If you have other additional fields defined for the frontend editing, you need to manually combine these settings with the settings of the register4cal extension.

To include the additional fields from the register4cal extension in the frontend editing forms, include the following markers in the cal templates:

###MODULE__tx_register4cal_activate###	Field "Activate registration"
###MODULE__tx_register4cal_regstart###	Field "Start of registration period"
###MODULE__tx_register4cal_regend###	Field "End of registration period"
###MODULE__tx_register4cal_maxattendees###	Field "Maximum number of attendees"
###MODULE__tx_register4cal_waitlist###	Field "Activate waitlist"

When including these markers, please take care of the double underscore after MODULE as well as the upper/lower case!

Using rlmp_dateselectlib

If you wish to include a date selector for the "Start/End of registration period"-fields, set the TypoScript variable plugin.tx_register4cal.edit.useDateSelector to 1. Of course you need to install the rlmp_dateselectlib extension, too. You can then configure the date selector via the plugin.tx_register4cal.edit.calConf settings. These settings are taken from the rlmp_dateselectlib extension, thus please referr to the manual of this extension for further details on these settings.

To add the date selectors to the fields you need to include placeholders in the stdWrap settings for the fields. In plugin.tx_cal_controller.view.create_event.tx_register4cal_regstart_stdWrap.dataWrap and plugin.tx_cal_controller.view.edit_event.tx_register4cal_regstart_stdWrap.dataWrap use the placeholder ###REGSTART_SELECTOR### at the position you wish the selector to be included. In plugin.tx_cal_controller.view.create_event.tx_register4cal_regend_stdWrap.dataWrap and plugin.tx_cal_controller.view.edit_event.tx_register4cal_regstart_stdWrap.dataWrap use the placeholder ###REGEND_SELECTOR###

Telling the users that they need to login

If an user is not logged in, then by default, nothing is displayed below the event display. This behavior can be changed. If you set the TypoScript parameter plugin.tx_register4cal_pi1.disableNeedLoginForm to 0, an information is being displayed. It tells the user, that he needs to login. By default this form then provides two buttons:

• Login using an existing account

• Create an one-time account

Both buttons need some customizing, which should be described here:

Login using an existing account

This option requires the folowing TypoScript parameters to be set

plugin.tx_register4cal_pi1.loginpid	Set this parameter to the page if of the page, containing your login form
plugin.tx_register4cal_pi1.loginreturnparam	Set this parameter to the url-parameter, which tells the login form to which page the user should be forwarded, once he has logged in successfully.

After setting this parameters, the user can click on the "Login using an existing account"-Button, enter his login data and will

be directed back to the registration form afterwards.

To disable the "Login using an existing account"-Button, clear plugin.tx_register4cal_pi1.forms.default.loginbutton.

Create an one-time account

This option requires the extension onetimeaccount to be setup and configured as described in the manual.

In addtion, the folowing TypoScript parameters need to be set

plugin.tx_register4cal_pi1.onetimeaccountpid	Set this parameter to the page if of the page, containing your onetimeaccount-plugin
plugin.tx_register4cal_pi1.onetimeaccountreturnparam	Set this parameter to the url-parameter, which tells the onetimeaccount-plugin to which page the user should be forwarded, once he has created his account in successfully.

After setting this parameters, the user can click on the "Create an one-time account"-Button, enter his data and will be directed back to the registration form afterwards.

To disable the "Create an one-time account"-Button, clear plugin.tx_register4cal_pi1.forms.default.onetimeloginbutton.

Template and markers

General Information

You can place every marker in the template and have a corresponding stdWrap element in TypoScript to define the layout and the content of the marker.

The TypoScript definition for a marker is firstly searched in the subsection of plugin.tx_register4cal_pi1.forms which corresponds to the template section. If no definition is contained there, the definition in the "default" section is being used.

Markers for user, event and registration data

Several special markers can be used to include data from the user, the event, the organizer, the location and the registration record:

Marker	Function
USER_[xxx]	Marker for the field [xxx] from the user record (table fe_users). It's either the user just registering or the user to which the registration record belongs
EVENT_[xxx]	Marker for the field [xxx] from the event record (table tx_cal_events).
ORGANIZER_[xxx]	Marker for the field [xxx] of the organizer record linked to the event (table tx_cal_organizer)
LOCATION_[xxx]	Marker for the field [xxx] of the location record linked to the event (table tx_cal_location)
EVENT_link	Link for the event single view (plugin.tx_register4cal_pi1.view.eventViewPid needs to be set!)
EVENT_formated_start	Startdate and Starttime of the event (formating from plugin.tx_register4cal_pi1.dateFormat and plugin.tx_register4cal_pi1.timeFormat)
EVENT_formated_end	Enddate and Endtime of the event (formating from plugin.tx_register4cal_pi1.dateFormat and plugin.tx_register4cal_pi1.timeFormat)
EVENT_organizer_name	Name of the organizer from the organizer record linked with the event.
EVENT_organizer_email	Email of the organizer from the organizer record linked with the event.
LABEL_[xxx]	Label field, content is taken from the translation file pi1/locallang.xml in the extension directory. IMPORTANT: as [xxx] is not allowed to contain dots, you can use hyphens instead, they will be replaced with dots when looking for the markers.
FIELDS	Inserts the defined userfields.
LINK	Inserts the link, necessary in the registration for to store the registrations.
STATUS	Status of registration
MAXATTENDEES	Maximum number of attendees for the event
NUMATTENDES	Number of currently registered attendees for the event
NUMWAITLIST	Number of entries on the waitlist for the event

Update from previous versions

Update from version less than 0.3.0

With version 0.3.0 the extension has been completely revised and improved. As a consequence of this, templates and configuration need to be checked and adapted.

Despite this, the extension is fully compatible to existing registrations stored in any previous version. The only thing you need to make sure is, that your template contains the " ###SHOW_[fieldtype]### section which was used in previous versions to define the layout of the userfields. The "###INPUT_[fieldtype]###" section is no longer necessary and may be removed.

Update to version 0.5.0

With version 0.5.0 a lot of new features were introduced. To avoid confusion with the templates and TS-settings, these have been renamed and reordered. Thus, check your templates and typoscript settings. You might need to rename your tempates and to rearrange your settings. There should be no do any further changes, unless you want to use the new features, which then need to be configured.

Known problems

– Although I tried to take care for security issues during development, I did not yet perform a security review, yet.

– The danish translation was provided before some of the new features have been implemented. I have adapted the translation using an online translation service. Thus, the translation might not be perfect in this sections. To indicate, which sections have been translated this way, these texts are in brackets.

Bugs, additional features and other support

There is a project page on forge.typo3.org. You can access it via http://forge.typo3.org/projects/show/extension-register4cal Please use this site to report bugs, request additional features or ask for support. This site also includes a subversion repository where the latest changes and corrections can be obtained.

ChangeLog

Please refer to the file changelog.txt in the main directory of the extension.