

Calendar extension "cz_simple_cal"

Christian Zenker

Calendar extension "cz_simple_cal"

Christian Zenker

Copyright © 2010 Christian Zenker

Abstract

`cz_simple_cal` is a simple calendar written on top of `extbase`

Table of Contents

1. Introduction	1
2. Concepts	2
Event Index	2
Fake Actions	2
The type date	2
3. HowTo's	4
Add a fake action	4
4. Copyright Notice	5
Glossary	6
A. The type date	7
Date Formats	7
Time Formats	9
Compound Formats	10
Relative Formats	11
Chaining	12

List of Tables

A.1. Used Symbols	7
A.2. Localized Notations	7
A.3. ISO8601 Notations	8
A.4. Used Symbols	9
A.5. 12 Hour Notation	9
A.6. 24 Hour Notation	10
A.7. Used Symbols	10
A.8. Localized Notations	11
A.9. Used Symbols	11
A.10. Day-based Notations	11

List of Examples

2.1. Examples of valid dates and times in YYYY-MM-DD syntax	2
2.2. Examples of valid dates and times in alternate syntax	3
A.1. Chaining of DateTime formats	12

Chapter 1. Introduction

Caution

This extension in alpha state.

It was only tried with TYPO3 4.4 and the corresponding extbase version. Different versions might work, but propably they do not.

Chapter 2. Concepts

This section tries to explain some of the basic concepts behind the calendar.

Event Index

Calendar Base introduced something called `New Recurring Event Model`. This concept was borrowed and applied to all events by default. The index is automatically updated if you modify an event. So depending on how many recurrances and exceptions you've set up, storing might take a while longer.

Note

The extension is smart enough to notify if you changed some values that actually require indexing to run again. So if you only change the title or a description, no indexing is done.

Caution

Until now there is no Indexer to re-index all your events. If you happen to mess up your index somehow (for example by changing an exception you have assigned to an event) you'll have to edit and save every event again. And due to the note above, you'll actually have to change something significant like the start time.

Due to the indexing of events you usually deal with `EventIndices` in your templates. But the objects are smart enough to tunnel unknown methods to the Event they belong to. So you can work with `EventIndices` as if they were `Events`.

Fake Actions

To make the extension as flexible as possible you can add fake actions to the controller in your `TypoScript`.

At the moment the only real actions are `listAction`, `showAction` and `countEventsAction`. `DispatchAction` serves as a fallback and default action.

See `HowTo: Add a fake action` to learn - guess what - how to add a fake action.

The type date

This type is quite heavily used in the extension. It allows for a very flexible and simple calculation of dates and times. It is based on the english language and international date and time formats.

The recommended way of setting a fixed day and time is the `YYYY-MM-DD (HH:MM:SS[T]?)?` syntax. These would be valid dates:

Example 2.1. Examples of valid dates and times in YYYY-MM-DD syntax

- `2009-02-13`
- `2009-02-13 23:31:30`
- `2009-02-13 23:31:30UTC`

But you can easily use different formats like these:

Example 2.2. Examples of valid dates and times in alternate syntax

- `13.2.09 23.31.30+00:00`
- `February 13th, 2009 11 pm`

Note

Note that you can't use localized month names here.

See the appendix for the complete syntax.

Chapter 3. HowTo's

Add a fake action

Adding a fake action is pretty simple and can be done only using TypeScript. Let's say we want to add a view that displays the event that has recently finished. We'll call this action `recent`.

1. Choose a fitting real action

In our example case this would be the `listAction`. `ShowAction` won't fit as you don't know the id of the event to display in advance. Instead, we'll limit the list to display only one event.

2. Extend the TypeScript configuration

Note

All given TypeScript paths are relative to `plugin.tx_czsimplecal`. I am too lazy to add this to the path each time, and so should you. Use the curly brackets: `plugin.tx_czsimplecal{ //... }`

The TypeScript settings already hold some action configurations. So we'll just copy the configuration for the `listAction` in `settings.Event.actions.list` to `settings.Event.actions.recent`.

The configurable options should be pretty obvious. But first we'll add `useAction = list` to the actions configuration. This way the fake `recentAction` knows which real action to call. Now we can change the other configuration like this:

```
recent {
    // ...
    startDate = now -1 month
    endDate   = now

    maxEvents = 1
    orderBy   = end
    order      = DESC
}
```

Guess what each of the configuration values is doing. ;)

3. Add the action to the allowed actions

Just add the name of the action to the `allowedViews` in the root of the extensions settings.

4. Create a view

Well, that's even simpler: Just copy the `list.html` template and rename it to `recent.html`. Do changes on the file if you like.

Chapter 4. Copyright Notice

This project uses some of the Fugue Icons [<http://p.yusukekamiyamane.com/>]. They are published under a Creative Commons [<http://creativecommons.org/licenses/by/3.0>] license:

Copyright (C) 2010 Yusuke Kamiyamane. All rights reserved. The icons are licensed under a Creative Commons Attribution 3.0 license.

Glossary

Date	A date usually means the combination of day and time. See Also Day, Time.
Day	When speaking of a <i>day</i> usually no time is meant. For example 1st January 2010 would be a day. See Also Date, Time.
Time	When speaking of a <i>time</i> usually no day is meant. For example 12:34:56 would be a time. See Also Date, Day.
Event (Domain Object)	The Domain Object <code>Event</code> represents a series of events that share some common information like the name or a description. Events might be recurrent or have exceptions in this recurrences. See Also <code>EventIndex</code> (Domain Object).
Event (Controller)	The most important controller for the Events. Technically it is no controller for the <code>Event</code> but for the <code>EventIndex</code>
<code>EventIndex</code> (Domain Object)	In contrast to the <code>Event</code> an <code>EventIndex</code> is a representation of a concrete occurrence of the event. So an <code>Event</code> that recurs every week will have a <code>EventIndex</code> representation for every week. Even not recurring Events have an <code>EventIndex</code> representation. Queries on several events are almost exclusively done on these domain objects. See Also <code>Event</code> .
Exception (Domain Object)	An <code>Exception</code> is an "Event" that symbolizes that an <code>Event</code> is not taking place when the exception is active. It might be recurring, but <code>Exceptions</code> are not stored as <code>Indices</code> in the database as it is done with <code>Events</code> .
<code>ExceptionGroup</code> (Domain Object)	A collection of <code>Exceptions</code> that belong together somehow.
<code>GetDate</code>	<code>GetDate</code> is a concept taken from the TYPO3 extension <code>cal</code> . <code>GetDate</code> makes some actions configurable using <code>GET</code> -parameters. All relative dates of the action are calculated based on that date.
<code>Timespan</code>	A <code>timespan</code> has a start and an end date and covers everything in between. There are no gaps in a <code>timespan</code> .
<code>Timeline</code>	A <code>timeline</code> is a collection of <code>timespans</code> . The contained <code>timespans</code> might overlap or build gaps. See Also <code>Timespan</code> .
Fake Action	One of the concepts of this calendar is to generate actions dynamically based on <code>TypoScript</code> configuration. Actions that have no method in the corresponding controller are called "fake actions". See Also <code>Real Action</code> .
Real Action	In comparison to fake actions the real actions have a method in the corresponding controller. These are the actions as they are conceptually intended by extbase. See Also <code>Fake Action</code> .

Appendix A. The type date

The first thing to mention is that you can *only* use english phrases and month names. Numeric formats are usually standartized formats.

Note

The type `date` is based on PHP's date and time formats [<http://php.net/manual/en/datetime.formats.php>]. With some exceptions all of the formats there can be used. Some of the relative formats require PHP 5.3, so you should avoid them if possible to keep compatibility.

Date Formats

Table A.1. Used Symbols

Description	Format	Examples
daysuf	"st" "nd" "rd" "th"	
dd	([0-2]?[0-9] "3"[01]) daysuf?	"7th", "22nd", "31"
DD	"0" [0-9] [1-2][0-9] "3" [01]	"07", "31"
m	'january' 'february' 'march' 'april' 'may' 'june' 'july' 'august' 'september' 'october' 'november' 'december' 'jan' 'feb' 'mar' 'apr' 'may' 'jun' 'jul' 'aug' 'sep' 'sept' 'oct' 'nov' 'dec' "I" "II" "III" "IV" "V" "VI" "VII" "VIII" "IX" "X" "XI" "XII"	
M	'jan' 'feb' 'mar' 'apr' 'may' 'jun' 'jul' 'aug' 'sep' 'sept' 'oct' 'nov' 'dec'	
mm	"0"? [0-9] "1"[0-2]	"0", "04", "7", "12"
MM	"0" [0-9] "1"[0-2]	"00", "04", "07", "12"
y	[0-9]{1,4}	"00", "78", "08", "8", "2008"
YY	[0-9]{2}	"00", "08", "78"
YY	[0-9]{4}	"2000", "2008", "1978"

Table A.2. Localized Notations

Description	Format	Examples
American month and day	mm "/" dd	"5/12", "10/27"
American month, day and year	mm "/" dd "/" y	"12/22/78", "1/17/2006", "1/17/6"
Four digit year, month and day with slashes	YY "/" mm "/" dd	"2008/6/30", "1978/12/22"
Four digit year and month (GNU)	YY "-" mm	"2008-6", "2008-06", "1978-12"

Description	Format	Examples
Year, month and day with dashes	y "-" mm "-" dd	"2008-6-30", "78-12-22", "8-6-21"
Day, month and four digit year, with dots, tabs or dashes	dd [.t-] mm [-.] YY	"30-6-2008", "22.12\t1978"
Day, month and two digit year, with dots or tabs	dd [.t] mm "." YY	"30.6.08", "22\t12\t78"
Day, textual month and year	dd ([\t-])* m ([\t-])* y	"30-June 2008", "22DEC78", "14 III 1879"
Textual month and four digit year (Day reset to 1)	m ([\t-])* YY	"June 2008", "DEC1978", "March 1879"
Four digit year and textual month (Day reset to 1)	YY ([\t-])* m	"2008 June", "1978-XII", "1879.MArCH"
Textual month, day and year	m ([\t-])* dd [.,stndrh\t]+ y	"July 1st, 2008", "April 17, 1790", "May.9,78"
Textual month and day	m ([\t-])* dd [.,stndrh\t]*	"July 1st,", "Apr 17", "May.9"
Day and textual month	d ([\t-])* m	"1 July", "17 Apr", "9.May"
Month abbreviation, day and year	M "-" DD "-" y	"May-09-78", "Apr-17-1790"
Year, month abbreviation and day	y "-" M "-" DD	"78-Dec-22", "1814-MAY-17"
Year (and just the year)	YY	"1978", "2008"
Textual month (and just the month)	m	"March", "jun", "DEC"

Table A.3. ISO8601 Notations

Description	Format	Examples
Eight digit year, month and day	YY MM DD	"15810726", "19780417", "18140517"
Four digit year, month and day with slashes	YY "/" MM "/" DD	"2008/06/30", "1978/12/22"
Two digit year, month and day with dashes	yy "-" MM "-" DD	"08-06-30", "78-12-22"
Four digit year with optional sign, month and day	[+-]? YY "-" MM "-" DD	"-0002-07-26", "+1978-04-17", "1814-05-17"

Note

For the y and yy formats, years below 100 are handled in a special way when the y or yy symbol is used. If the year falls in the range 0 (inclusive) to 69 (inclusive), 2000 is added. If the year falls in the range 70 (inclusive) to 99 (inclusive) then 1900 is added. This means that "00-01-01" is interpreted as "2000-01-01".

Note

The "Day, month and two digit year, with dots or tabs" format (dd [.t] mm "." yy) only works for the year values 61 (inclusive) to 99 (inclusive) - outside those years the *time format* "HH [.:] MM [.:] SS" has precedence.

Note

The "Year (and just the year)" format only works if a time string has already been found -- otherwise this format is recognised as HH MM.

Note

It is possible to over- and underflow the dd and DD format. Day 0 means the last day of previous month, whereas overflows count into the next month. This makes "2008-08-00" equivalent to "2008-07-31" and "2008-06-31" equivalent to "2008-07-01" (June only has 30 days).

It is also possible to underflow the mm and MM formats with the value 0. A month value of 0 means December of the previous year. As example "2008-00-22" is equivalent to "2007-12-22".

If you combine the previous two facts and underflow both the day and the month, the following happens: "2008-00-00" first gets converted to "2007-12-00" which then gets converted to "2007-11-30". This also happens with the string "0000-00-00", which gets transformed into "-0001-11-30" (the year -1 in the ISO 8601 calendar, which is 2 BC in the proleptic Gregorian calendar).

Time Formats

Table A.4. Used Symbols

Description	Formats	Examples
frac	. [0-9]+	".21342", ".85"
hh	"0"?[1-9] "1"[0-2]	"04", "7", "12"
HH	[01][0-9] "2"[0-4]	"04", "7", "19"
meridian	[AaPp] .? [Mm] .? [\0\t]	"A.m.", "pM", "am."
MM	[0-5][0-9]	"00", "12", "59"
II	[0-5][0-9]	"00", "12", "59"
space	[\t]	
tz	"(" [A-Za-z]{1,6} ")"? [A-Z][a-z]+([_]/[A-Z][a-z]+)+	"CEST", "Europe/Amsterdam", "America/Indiana/Knox"
tzcorrection	"GMT"? [+ -] hh ":"? MM?	" +0400", "GMT-07:00", "-07:00"

Table A.5. 12 Hour Notation

Description	Format	Examples
Hour only, with meridian	hh space? meridian	"4 am", "5PM"
Hour and minutes, with meridian	hh [:] MM space? meridian	"4:08 am", "7:19P.M."
Hour, minutes and seconds, with meridian	hh [:] MM [:] II space? meridian	"4:08:37 am", "7:19:19P.M."
MS SQL (Hour, minutes, seconds and fraction with meridian), PHP 5.3 and later only	hh ":" MM ":" II [:] [0-9]+ meridian	"4:08:39:12313am"

Table A.6. 24 Hour Notation

Description	Format	Examples
Hour and minutes	't'? HH [.:] MM	"04:08", "19.19", "T23:43"
Hour and minutes, no colon	't'? HH MM	"0408", "t1919", "T2343"
Hour, minutes and seconds	't'? HH [.:] MM [.:] II	"04.08.37", "t19:19:19"
Hour, minutes and seconds, no colon	't'? HH MM II	"040837", "T191919"
Hour, minutes, seconds and timezone	't'? HH [.:] MM [.:] II space? (tzcorrection tz)	"040837CEST", "T191919-0700"
Hour, minutes, seconds and fraction	't'? HH [.:] MM [.:] II frac	"04.08.37.81412", "19:19:19.532453"
Time zone information	tz tzcorrection	"CEST", "Europe/Amsterdam", "+0430", "GMT-06:00"

Compound Formats

Table A.7. Used Symbols

Description	Formats	Examples
DD	"0" [0-9] [1-2][0-9] "3" [01]	"02", "12", "31"
doy	"00"[1-9] "0"[1-9][0-9] [1-2][0-9][0-9] "3"[0-5][0-9] "36"[0-6]	"36"[0-6] "000", "012", "366"
frac	. [0-9]+	".21342", ".85"
hh	"0"?[1-9] "1"[0-2]	"04", "7", "12"
HH	[01][0-9] "2"[0-4]	"04", "7", "19"
meridian	[AaPp] .? [Mm] .? [\0t]	"A.m.", "pM", "am."
ii	[0-5][0-9]	"04", "8", "59"
II	[0-5][0-9]	"04", "08", "59"
M	'jan' 'feb' 'mar' 'apr' 'may' 'jun' 'jul' 'aug' 'sep' 'sept' 'oct' 'nov' 'dec'	
MM	[0-5][0-9]	"00", "12", "59"
space	[\t]	
ss	[0-5][0-9]	"04", "8", "59"
SS	[0-5][0-9]	"04", "08", "59"
W	"0"[1-9] [1-4][0-9] "5"[0-3]	"05", "17", "53"
tzcorrection	"GMT"? [+ -] hh ":"? MM?	" +0400", "GMT-07:00", "-07:00"
YY	[0-9]{4}	"2000", "2008", "1978"

Table A.8. Localized Notations

Description	Format	Examples
Common Log Format	dd "/" M "/" YY : HH ":" II ":" SS space tzcorrection	"10/Oct/2000:13:55:36 -0700"
EXIF	YY ":" MM ":" DD " " HH ":" II ":" SS	"2008:08:07 18:11:31"
ISO year with ISO week	YY "-" "?" "W" W	"2008W27", "2008-W28"
ISO year with ISO week and day	YY "-" "?" "W" W "-" ? [0-7]	"2008W273", "2008-W28-3"
PostgreSQL: Year with day-of-year	YY "." "?" doy	"2008.197", "2008197"
SOAP	YY "-" MM "-" DD "T" HH ":" II ":" SS frac tzcorrection?	"2008-07-01T22:35:17.02", "2008-07-01T22:35:17.03+08:00"
Unix Timestamp	"@" "-" "?" [0-9] +	"@1215282385"
XMLRPC	YY MM DD "T" hh ":" II ":" SS	"20080701T22:38:07", "20080701T9:38:07"
XMLRPC (Compact)	YY MM DD 't' hh II SS	"20080701t223807", "20080701T093807"
WDDX	YY "-" mm "-" dd "T" hh ":" ii ":" ss	"2008-7-1T9:3:37"

Note

The "W" in the "ISO year with ISO week" and "ISO year with ISO week day day" formats is case-sensitive, you can only use the upper case "W".

The "T" in the SOAP, XMLRPC and WDDX formats is case-sensitive, you can only use the upper case "T".

Relative Formats

Table A.9. Used Symbols

Description	Format
dayname	'sunday' 'monday' 'tuesday' 'wednesday' 'thursday' 'friday' 'saturday' 'sun' 'mon' 'tue' 'wed' 'thu' 'fri' 'sat' 'sun'
number	[+-]?[0-9] +
reltext	'next' 'last' 'previous' 'this'
space	[\t] +
unit	(('sec' 'second' 'min' 'minute' 'hour' 'day' 'month' 'year') 's'?) 'weeks' daytext

Table A.10. Day-based Notations

Format	Description	Examples
'yesterday'	Midnight of yesterday	"yesterday 14:00"

Format	Description	Examples
'midnight'	The time is set to 00:00:00	
'today'	The time is set to 00:00:00	
'now'	Now - this is simply ignored	
'noon'	The time is set to 12:00:00	"yesterday noon"
'tomorrow'	Midnight of tomorrow	
'first day' ' of'?	Sets the day of the first of the current month. This phrase is best used together with a month name following it.	"first day of January 2008"
'last day' ' of'?	Sets the day to the last day of the current month. This phrase is best used together with a month name following it.	"last day of next month"
'last' space dayname space 'of'	Calculates the <i>last</i> week day of the current month.	"last sat of July 2008"
number space ? (unit 'week')	Handles relative time items where the value is a number.	"+5 weeks", "12 day", "-7 weekdays"
reltext space 'week'	Handles the special format "weekday + last/this/next week".	"Monday next week"

Caution

Weeks always start with mondays. That's a difference to the original DateTime starts with sundays.

Chaining

All afore mentioned formats can be chained by using the pipe character (|). All rules will be applied one after another.

Example A.1. Chaining of DateTime formats

- `first day this month|monday this week`
- `2009-02-13 00:00:00|sunday next week`