

**Technical documentation
for calendar extension
"cz_simple_cal" for developers**

Christian Zenker

Technical documentation for calendar extension "cz_simple_cal" for developers

Christian Zenker

Copyright © 2010 Christian Zenker

Abstract

cz_simple_cal is a simple calendar written on top of extbase.

This documentation aims to help developers who'd like to understand and extend this extension.

Table of Contents

1. Introduction	1
2. Concepts	2
Event Index	2
Fake Actions	2
3. Testing	3
Setting up Unit Tests	3
Setting up Selenium Tests	3
Glossary	5

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`.

Chapter 3. Testing

This section will explain how to set up a testing environment for the extension.

Setting up Unit Tests

You need to install the extension `phpunit` in order to run Unit Tests. You can find the extension `phpunit` on TER [<http://typo3.org/extensions/repository/view/phpunit/current/>].

There is not much to say on the installation of the extension. It is straight forward and as it comes bundled with a PHPUnit library, so you won't even need to install a PEAR package. If you should encounter any difficulties, consult the documentation on the internet [<http://typo3.org/documentation/document-library/extension-manuals/phpunit/3.4.12/view/>] or your local machine.

After successful installation you should see the PHPUnit module in the Admin Section. Click it and select `cz_simple_cal` as extension to run the unit tests. Click `Run all tests` and all the tests should be run. If the bar is not green after running all tests or the word "Success" is not displayed at the bottom of the list, something went wrong.

Setting up Selenium Tests

Tip

You should use a separate TYPO3 installation and database for running selenium tests as you need to create some pages for testing.

You should have the extension `phpunit` installed as described in the last section. All Selenium Tests are run inside PHPUnit, so this is vital.

Note

If the selenium extension is not enabled, all Selenium Tests will be skipped automatically.

Now install Selenium Remote Control (RC) - if you don't have it running already.

Download it at selenium.org [<http://seleniumhq.org/download/>] and unpack. All you need is the `selenium-remote-control-1.x.x/selenium-server-1.x.x` folder - you might as well delete all the others.

Run the `selenium-server.jar` using JAVA, for example by typing **`java -jar selenium-server.jar`** on your console.

Import the data stored in `Tests/Selenium/typo3_testing.sql` to your database.

Use the scheduler to create a `Index all events (cz_simple_cal)-task` and run it to create all the recurring events.

Note

The extension scheduler is shipped with the TYPO3 core but disabled by default.

If you have done that install the extension `selenium` from TER [<http://typo3.org/extensions/repository/view/selenium/current/>]. You can ignore the backend module created by that extension and jump back to the PHPUnit module.

Now if you run the tests again a browser window should pop up and magically call some pages of the frontend.

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>
EventIndex (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> .
ExceptionGroup (Domain Object)	A collection of <code>Exceptions</code> that belong together somehow.
GetDate	<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.
Timespan	A timespan has a start and an end date and covers everything in between. There are no gaps in a timespan.
Timeline	A timeline is a collection of timespans. The contained timespans 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> .