

Documentation for KS.Umbraco4.Calendar data type

1. Install the package in Umbraco 4 or Umbraco 6.
If you have an Umbraco installation prior to version 6.1.0 install the package containing Newtonsoft.JSON.
2. Create a new Data Type based on the property editor "Umbraco UserControl Wrapper" and select the usercontrol "ucCalendar.ascx"
3. Add the new datatype to a document type
4. Create some content with the new data type
5. Create a razor-script for listing the events.

```
1 @inherits umbraco.MacroEngines.DynamicNodeContext
2
3 using KS.Umbraco4.Calendar.Core
4 <h1>Kalender</h1>
5 <ul>
6     @foreach (CalendarEvent ce in Calendar.getEvents(DateTime.Now.AddDays(-50), DateTime.Now.AddDays(50), "event"))
7     {
8         <li>
9             <a href="@ce.content.Url">
10                 @ce.startDate.ToString("dd.MM.yy HH:mm") - @ce.content.Name
11             </a>
12         </li>
13     }
14 </ul>
```

There are several method to get your calendar events:

- `getEvents(DateTime startDate, DateTime endDate, string propertyType, bool splitNoneRecurring = true)`
- `getEvents(DateTime startDate, DateTime endDate, string propertyType, string documentType, bool splitNoneRecurring = true)`
- `getEvents(DateTime startDate, DateTime endDate, string propertyType, string documentType, DynamicNode startNode, bool splitNoneRecurring = true)`
- `getEvents(DateTime startDate, DateTime endDate, string propertyType, DynamicNode startNode, bool splitNoneRecurring = true)`
- `getEvents(DateTime startDate, DateTime endDate, string propertyType, int nodeId, bool splitNoneRecurring = true)`
- `getEvents(DateTime startDate, DateTime endDate, string propertyType, string documentType, int nodeId, bool splitNoneRecurring = true)`

The `bool splitNoneRecurring` is an optional parameter that by default is set to `true`. This means that a non-recurring event spanning several days will be split into one event for each day it spans. If you set the parameter `splitNoneRecurring` to `false` you will only get one event for the entire event.

All these will return a `List<CalendarEvent>` for you to use.

The `startDate` is the actual date of the event, `content` is the contentnode as a `DynamicNode`, so you can use `ce.content.Name`, `ce.content.Url` and everything else you are used to.

```

public class CalendarEvent
{
    public int recurrence { get; set; } // 1 - not recurring, 2 - repeat daily, 3 - repeat weekly, 4 - repeat monthly, 5 - repeat yearly
    public int weekInterval { get; set; } //If repeat weekly this tells you if its every week, every 2nd week, every 3rd week etc.
    public int monthYearOption { get; set; } //1 - Use startdate, 2 - Specify weekday etc. for event
    public int? monthOption { get; set; } // 1 - Event month, 2 - Choose months
    public int interval { get; set; } //interval of weekly events
    public int weekDay { get; set; } //weekDay for specified monthly and yearly events
    public int month { get; set; } //month for specified monthly and yearly events
    public int?[] days { get; set; } //array of days 0-6 for daily events
    public int?[] months { get; set; } //array of monts 1-12 for monthly events
    public DateTime startDate { get; set; } //start date of the event
    public DateTime? endDate { get; set; } //end date of the event
    public DynamicNode content { get; set; } //the node containing the event
    public string Debug { get; set; }
}

```

This package is based on the KS.Umbraco7.Calendar found here:

<http://our.umbraco.org/projects/website-utilities/ksumbraco7calendar> and uses much of the same code. There are different language available for the data type, but for now it can only be altered in the JavaScript-file ks.umbraco4.calendar.controller.js located in /scripts/backend

Look at for `$http.get("/scripts/backend/language/en-GB.js")` approximate at line 46. Insert you preferred language here. The language-files is located in the scripts/backend/language-folder.

Source-files available at <http://github.com/ombakke/KS.Umbraco4.Calendar>

Ole Martin Bakke, 07.09.2014

omb@kreativstrek.no