

# Allgemeines

## Architektur der Anwendung

Grundsätzlich führen viele Wege nach Rom bzw. es gibt nicht den einen Weg, wie die gestellte Aufgabe mit Magnolia gelöst werden muss. Es wäre problemlos möglich, eine Anwendung zu erstellen, die ähnlich wie z. B. mit Rails die einzelnen Event-Seiten dynamisch während der Laufzeit "zusammenbaut". Dadurch würde man aber einen der wesentlichen Vorteile bei der Verwendung eines CMS verlieren: Das Editieren von Webseiten auch durch nicht-technische Anwender (dies ist zudem ein Bereich, wo Magnolia meiner Meinung nach besonders stark ist).

Um Magnolia's Stärke bei der einfachen Integration von externen Daten zu demonstrieren, werden die Daten in der vorliegenden Lösung über einen asynchronen Hintergrund-Job importiert und nicht jedes Mal online neu abgefragt. Dies verringert auch die Last auf dem System bei vielen gleichzeitig zugreifenden Benutzern.

Nachfolgend habe ich einige weitere grundlegende Informationen zu Magnolia allgemein und zur Problemstellung speziell zusammengestellt, um ein besseres Verständnis der Funktionsweise von Magnolia CMS zu ermöglichen.

Bei weiterem Interesse stehen unsere ausführlichen Dokumentationen jederzeit online zur Verfügung: <http://documentation.magnolia-cms.com>.

## Server starten

Die Magnolia Anwendungen werden standardmäßig mit dem Apache Tomcat Server ausgeführt. Den Server kann man über die im "bin" - Verzeichnis enthaltenen Bash-Skripte starten:

- *startup.sh* / *shutdown.sh* für einen Start als Hintergrund-Task
- *catalina.sh run* für einen Start in der aktuellen Konsole

Folgende URLs stehen nach dem Start zur Verfügung:

<a href="http://localhost:8080">http://localhost:8080</a>	„Root“ mit Magnolia Menü
<a href="http://localhost:8080/magnoliaAuthor/magnolia/admincentral">http://localhost:8080/magnoliaAuthor/magnolia/admincentral</a>	AdminCentral Author Server
<a href="http://localhost:8080/magnoliaPublic/magnolia/admincentral">http://localhost:8080/magnoliaPublic/magnolia/admincentral</a>	AdminCentral Public Server
<a href="http://localhost:8080/magnoliaPublic/">http://localhost:8080/magnoliaPublic/</a>	Startseite der eigentlichen Website

## Anmeldung

Nach dem Start des Tomcat Servers mit Magnolia kann man sich mit den Standard-Credentials

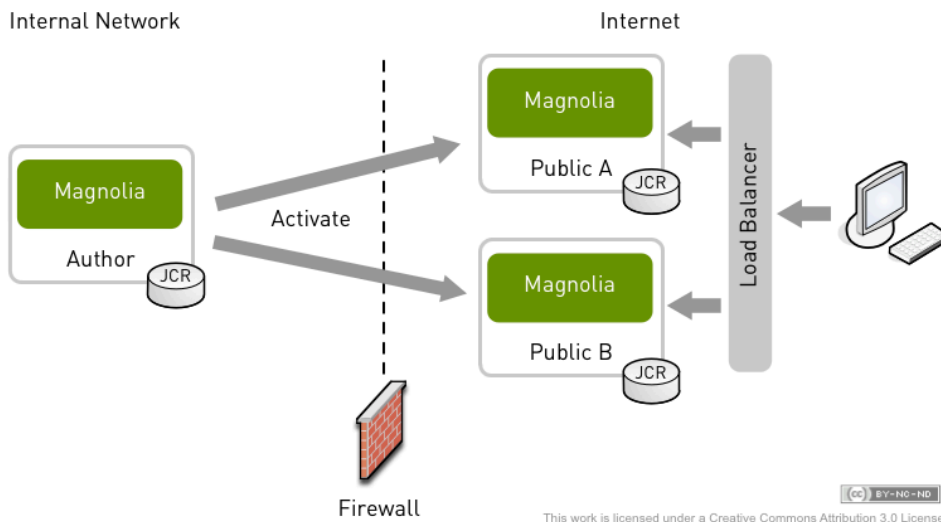
User name	superuser
Password	superuser

am System anmelden. Die Rolle "superuser" besitzt die höchsten Rechte und kann alle Aktionen innerhalb Magnolia durchführen.

## Author- und Public-Instanzen

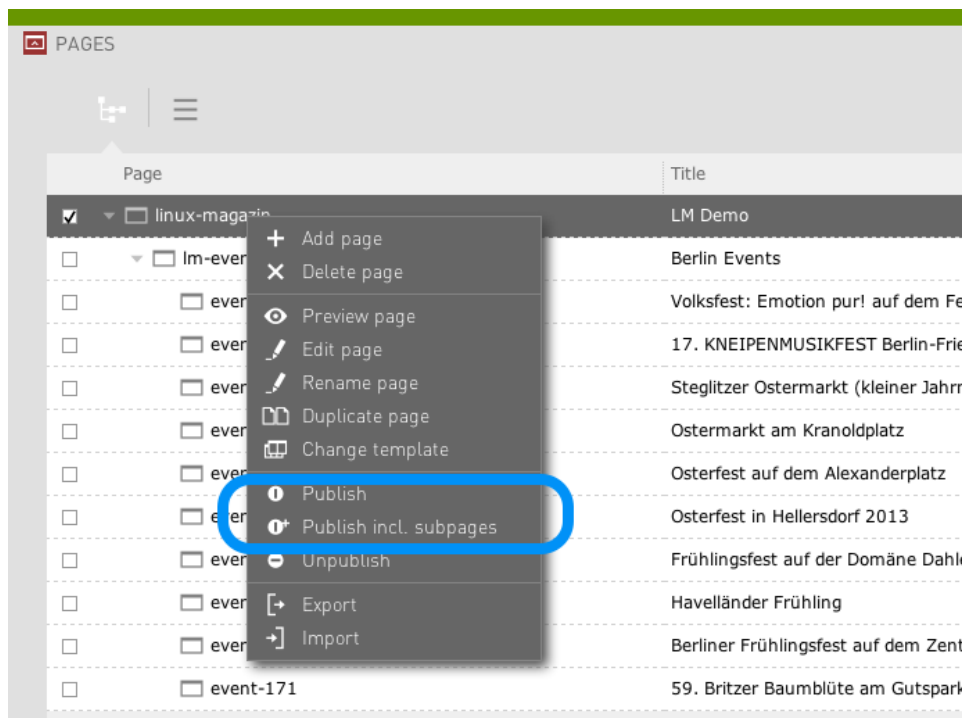
Magnolia untergliedert sich normalerweise in mehrere Server-Instanzen, wobei ein Server die Rolle des "Authors" übernimmt. Auf dem Author wird Content und Konfiguration bearbeitet und dann anschliessend (ggf. nach Begutachtung durch weitere Personen) auf eine oder mehrere andere Server publiziert.

Author and public instances



Weitere Konfigurationsbeispiele findet man hier: <http://documentation.magnolia-cms.com/display/DOCS/Instance+diagrams>

Werden auf dem Author-Server Änderungen vorgenommen (Web-Seiten erstellt, geändert gelöscht, etc.) so müssen diese auf die Public-Instanz „gepublished“ werden.



Nach dem Publish-Prozess stehen die Änderungen auch auf dem Public Server (also der fertigen Website) zur Verfügung.

**Hinweis:** Diesen Prozess kann man auch automatisieren oder im Code abbilden, der Einfachheit halber habe ich es aber bei der „normalen“ Methodik belassen.

## Standard Templating Kit

(Offizielle Dokumentation: <http://documentation.magnolia-cms.com/display/DOCS/Standard+Templating+Kit>)

Die einfachste Variante, die Aufgabe zu lösen, ist die Verwendung von Magnolia's Standard Templating Kit (STK). Das ist ein Best-Practice-Framework, das viele gängige Layouts und Komponenten zur einfachen Verwendung beinhaltet und zudem automatisch Unterstützung für mobile Endgeräte bietet. Durch reines Anpassen der vorhandenen Templates und Komponenten lässt sich somit die Entwicklungsgeschwindigkeit enorm verkürzen und die Qualität des Endresultats oft verbessern.

Das STK beinhaltet bereits Templates für die Darstellung von Events:

- Event Overview Template: <http://demopublic.magnolia-cms.com/demo-project/news-and-events/events-overview.html>
- Event Template: <http://demopublic.magnolia-cms.com/demo-project/service/dummy-events/another-event-12.html>

Das STK (bzw. das Magnolia Templating generell) kann zudem mit Java-Klassen "unterlegt" werden, wodurch sich mächtige Programmlogiken implementieren lassen.

Das Event Overview Template funktioniert standardmässig z. B. so, dass innerhalb der betreffenden Website automatisch Seiten mit dem Event-Template ermittelt und dann in einer übersichtlichen Liste dargestellt werden.

Um die Aufgabe zu lösen, muss man also nichts weiter tun, als die Events einzulesen und daraus Seiten mit dem Event-Template zu generieren. Anschliessend reichen ein paar einfache Customizations der bestehenden Templates, um das gewünschte Ergebnis zu erzielen. Anpassungen müssen z. B. gemacht werden, um abweichende/neue Attribute wie z. B. den Bezirk anzuzeigen.

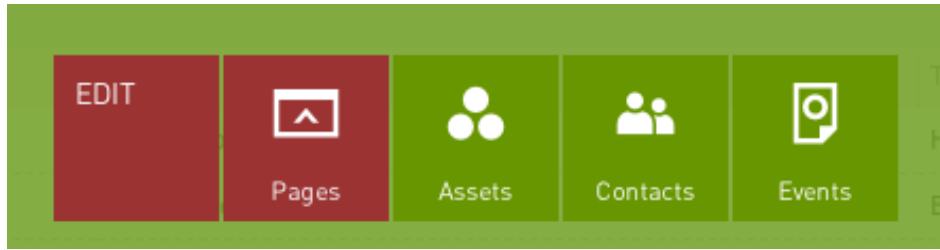
Die Kalender-Widgets implementiert man dann als Komponente, die sich dem bestehen Events Overview Template hinzufügen lässt.

Das STK kann zum Grossteil über Anpassung der Konfiguration auf die eigenen Bedürfnisse zugeschnitten werden, es verhält sich dabei analog zu aus der objektorientierten Programmierung bekannten Mechanismen. Man kann vorhandenes mit "extend" und "override" anpassen und dabei Duplikate vermeiden, indem man nur das Delta neu definiert.

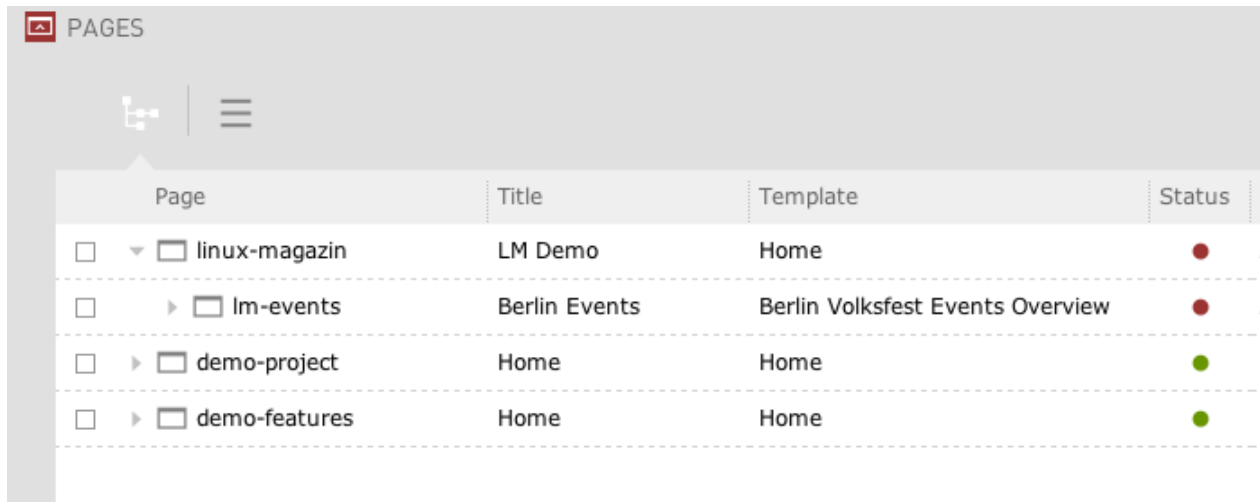
Prinzipiell habe ich die Aufgabe auch so gelöst wie in den vorherigen Abschnitten beschrieben.

## Betrachten und Editieren von Webseiten

Im Magnolia Content Repository enthaltene Webseiten kann man über die Pages App verwalten:



Nach Aufruf der Pages App gelangt man zum Website-Tree:



Page	Title	Template	Status
<input type="checkbox"/> linux-magazin	LM Demo	Home	●
<input type="checkbox"/> Im-events	Berlin Events	Berlin Volksfest Events Overview	●
<input type="checkbox"/> demo-project	Home	Home	●
<input type="checkbox"/> demo-features	Home	Home	●

Die für die Aufgabe relevante Website befindet sich unter "linux-magazin". Die anderen beiden Websites dienen nur zur Verdeutlichung der im STK vorhandenen Templates und der Experimentierfreude des Autors...

Die rote Anzeige unter Status bedeutet übrigens, dass der betreffende Content nicht publiziert worden ist (siehe Author- und Public-Instanzen).

Importierte Events werden unter dem Pfad "Im-events" abgelegt. Zu Testzwecken kann man diesen Knoten vor der Ausführung des Imports löschen, da nur Events als Seite angelegt werden, deren ID noch nicht vorhanden ist.

Vorhandene Events können direkt auf der zugehörigen Website editiert werden:



PAGE HEADER 

## 18. Tulpenfest Potsdam

 Print  Bookmark

**Apr 13** **18. Tulpenfest Potsdam**  

**Apr 13**

<b>Date</b>	April 13, 2013 - April 13, 2013
<b>Time</b>	Sa 10:00-20:00 So 11:00-19:00
<b>Bezirk</b>	Potsdam
<b>Location</b>	im Holländischen Viertel
<b>URL</b>	<a href="http://www.jan-bouman-haus.de">http://www.jan-bouman-haus.de</a>
<b>E-Mail</b>	<a href="mailto:info@jan-bouman-haus.de">info@jan-bouman-haus.de</a>
<b>Veranstalter</b>	Förderverein zur Pflege niederländischer Kultur in Potsdam e. V., Tel.: 0331/280 37 73, Fax: 0331/280 58 72
<b>Bemerkung</b>	Eintritt 4 €, Kinder bis 14 Jahre frei

Nach Klick auf das Icon zum Editieren erscheint der zum Template gehörende Dialog mit der Möglichkeit, Daten zu verändern.

The screenshot shows a dialog box titled "Event Intro" with a close button (X) and a help icon (?). The dialog contains the following fields:

- Page Headline \*: 18. Tulpenfest Potsdam
- Event Title \*: 18. Tulpenfest Potsdam
- Start Date&Time \*: 2013-04-13 (calendar icon)
- End Date&Time: 2013-04-13 (calendar icon)
- Location \*: im Holländischen Viertel
- Link Title: (empty)
- Page URL: www.jan-bouman-haus.de
- Event Abstract \*: (empty text area)
- Image: (empty text input) [SELECT NEW...]
- Image Location: left of the text (dropdown menu)
- Event ID: 411
- Zeit: Sa 10:00-20:00 So 11:00-19:00
- Bezirk: Potsdam
- Veranstalter: Förderverein zur Pflege niederländischer Kultur in Potsdam e. V., Tel.: 0331/280 37 73. Fax: 0331/280 58 72

At the bottom right, there are two buttons: "CANCEL" and "SAVE CHANGES".

Zur Erinnerung: All diese Templates, Komponenten und Dialoge waren bereits standardmässig in Magnolia enthalten, so dass ich nur das Delta zum Bestehenden definieren musste. Eine enorme Zeitersparnis!

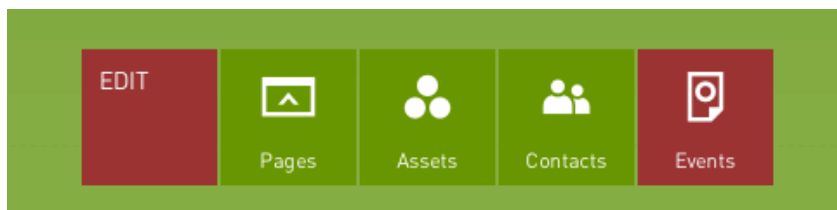
## Magnolia 5 Apps

Eine der wesentlichen Neuerungen von Magnolia 5 sind Apps (<http://documentation.magnolia-cms.com/display/DOCS/Apps>). Mit dieser Funktionalität lassen sich einfach (z. B. nur über Konfiguration) Anwendungen bauen, die zudem bereits eine schöne Benutzeroberfläche mitbringen.

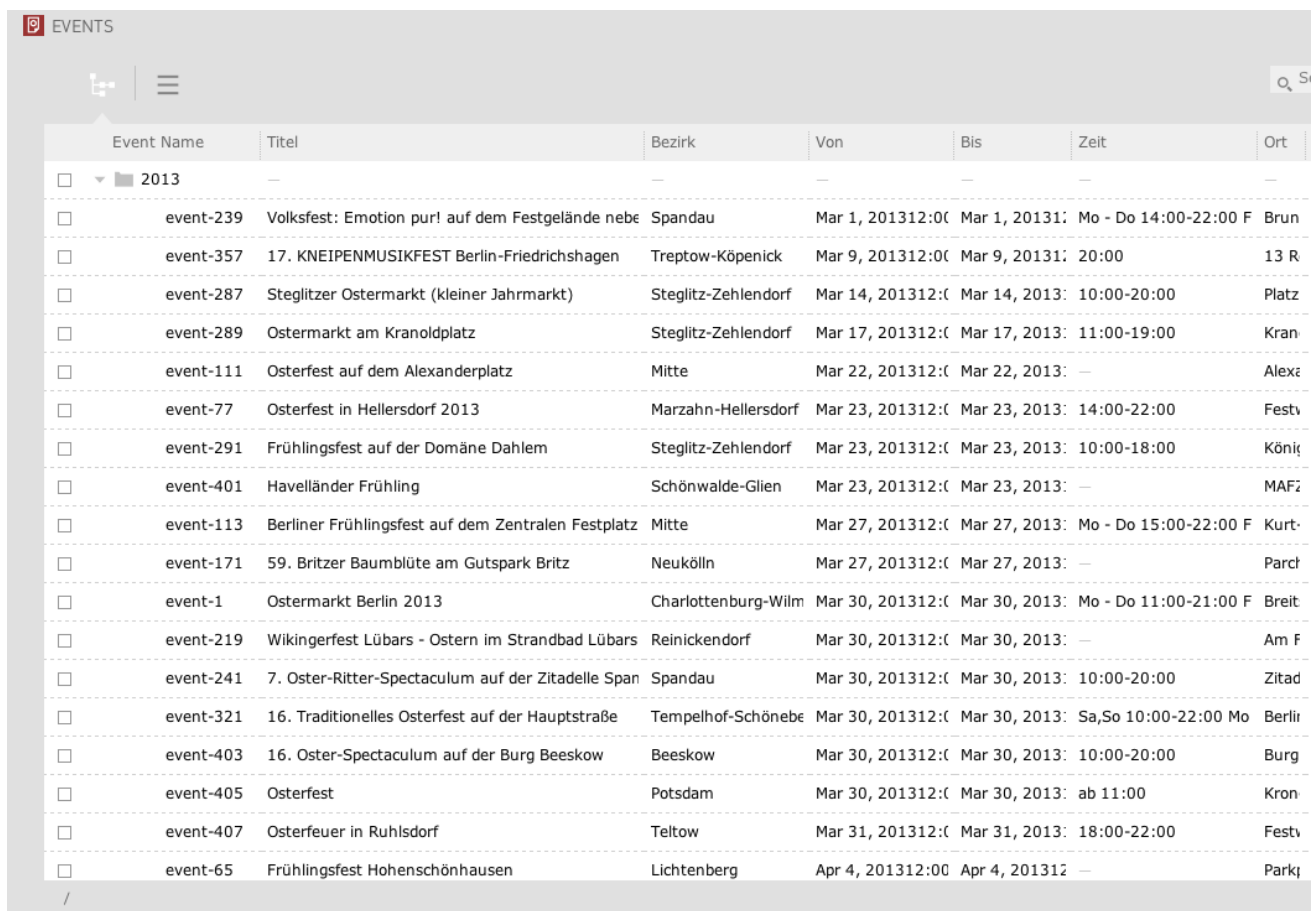
Um diese neue Möglichkeit zu demonstrieren habe ich zusätzlich noch eine solche App erstellt und den Import so gestaltet, dass die Event-Daten auch in einer Event-App gespeichert werden können. Mit dieser App ist es möglich, die vorhandenen Volksfest-Daten zu durchsuchen, zu editieren und auch zu löschen. Bearbeitet man ein Event, so werden die Daten der zugehörigen Webseite automatisch angepasst.

### Zugriff auf die Events App

Die Events App findet man nach dem Login an der Author Instanz unter der Gruppe "Edit".



Ein Klick auf die Schaltfläche wechselt direkt zur zugehörigen Anwendung.

A screenshot of the 'EVENTS' application interface. It features a table with columns for 'Event Name', 'Titel', 'Bezirk', 'Von', 'Bis', 'Zeit', and 'Ort'. The table contains a list of events for the year 2013, including details like event ID, title, district, dates, and location. A search bar is visible in the top right corner.

Event Name	Titel	Bezirk	Von	Bis	Zeit	Ort
2013						
event-239	Volksfest: Emotion pur! auf dem Festgelände nebe	Spandau	Mar 1, 2013 12:00	Mar 1, 2013 1:	Mo - Do 14:00-22:00	F Brun
event-357	17. KNEIPENMUSIKFEST Berlin-Friedrichshagen	Treptow-Köpenick	Mar 9, 2013 12:00	Mar 9, 2013 1:	20:00	13 R
event-287	Steglitzer Ostermarkt (kleiner Jahrmarkt)	Steglitz-Zehlendorf	Mar 14, 2013 12:00	Mar 14, 2013:	10:00-20:00	Platz
event-289	Ostermarkt am Kranoldplatz	Steglitz-Zehlendorf	Mar 17, 2013 12:00	Mar 17, 2013:	11:00-19:00	Kran
event-111	Osterfest auf dem Alexanderplatz	Mitte	Mar 22, 2013 12:00	Mar 22, 2013:	—	Alexa
event-77	Osterfest in Hellersdorf 2013	Marzahn-Hellersdorf	Mar 23, 2013 12:00	Mar 23, 2013:	14:00-22:00	Festv
event-291	Frühlingsfest auf der Domäne Dahlem	Steglitz-Zehlendorf	Mar 23, 2013 12:00	Mar 23, 2013:	10:00-18:00	Köniç
event-401	Havelländer Frühling	Schönwalde-Glien	Mar 23, 2013 12:00	Mar 23, 2013:	—	MAFz
event-113	Berliner Frühlingsfest auf dem Zentralen Festplatz	Mitte	Mar 27, 2013 12:00	Mar 27, 2013:	Mo - Do 15:00-22:00	F Kurt-
event-171	59. Britzer Baumbüte am Gutspark Britz	Neukölln	Mar 27, 2013 12:00	Mar 27, 2013:	—	Parç
event-1	Ostermarkt Berlin 2013	Charlottenburg-Wilm	Mar 30, 2013 12:00	Mar 30, 2013:	Mo - Do 11:00-21:00	F Breit
event-219	Wikingerfest Lübars - Ostern im Strandbad Lübars	Reinickendorf	Mar 30, 2013 12:00	Mar 30, 2013:	—	Am F
event-241	7. Oster-Ritter-Spectaculum auf der Zitadelle Span	Spandau	Mar 30, 2013 12:00	Mar 30, 2013:	10:00-20:00	Zitad
event-321	16. Traditionelles Osterfest auf der Hauptstraße	Tempelhof-Schönebe	Mar 30, 2013 12:00	Mar 30, 2013:	Sa,So 10:00-22:00	Mo Berli
event-403	16. Oster-Spectaculum auf der Burg Beeskow	Beeskow	Mar 30, 2013 12:00	Mar 30, 2013:	10:00-20:00	Burg
event-405	Osterfest	Potsdam	Mar 30, 2013 12:00	Mar 30, 2013:	ab 11:00	Kron
event-407	Osterfeuer in Ruhlsdorf	Teltow	Mar 31, 2013 12:00	Mar 31, 2013:	18:00-22:00	Festv
event-65	Frühlingsfest Hohenschönhausen	Lichtenberg	Apr 4, 2013 12:00	Apr 4, 2013 12:	—	Park

Die importierten Events lassen sich in verschiedenen Ansichten darstellen oder auch durchsuchen. Zudem können sie bearbeitet oder entfernt werden:

The screenshot shows a web form titled "Edit event" with a help icon. The form is labeled "EVENT" and contains the following fields:

- Event ID: 489
- Event name: event-489
- Event Title \*: Mud Day: der erste Matshtag Deutschlands - ein Familienevent
- Datum von \*: 2013-08-18 (with a calendar icon showing 31)
- Datum bis: 2013-08-18 (with a calendar icon showing 31)
- Zeit: 12:00-17:00
- Location \*: Spargel- und Erlebnishof Klaistow
- Bezirk: Klaistow
- Link Titel: (empty)
- Link / WWW: www.buschmann-winkelmann.de
- Veranstalter: Buschmann & Winkelmann GmbH, Tel.: 033206/610 70, Fax: 033206/610 89
- Email: info@buschmann-winkelmann.de
- Bemerkungen: (empty text area)

At the bottom right, there are two buttons: "CANCEL" and "SAVE CHANGES".

Editiert und speichert man die Angaben in diesem Formular, wird automatisch die zugehörige Webseite ermittelt und die Abgaben zum Event aktualisiert.

**Hinweis:** Aus Zeitgründen habe ich darauf verzichtet, volle CRUD-Funktionalität (z. B. mit Erstellen von neuen Events) zu implementieren. Angaben wie die Event-ID habe ich der Einfachheit halber als readonly definiert.

Bei der Umsetzung der Aufgabe als App würde man in einem Real-World-Szenario die Events nur selektiv zusätzlich als physische Webseiten anlegen (dann, wenn man als Editor einzelne Events bearbeiten möchte). Die ohne Änderung übernommen Events würden dann mit "dynamischen" Templates implementiert, d. h. die Event-Daten würden zur Laufzeit aus der Events App geholt. Bei der Events App hier handelt es sich um einen erweiterten Showcase, der die Mächtigkeit der neuen Features von Magnolia 5 demonstrieren soll.

## Umsetzung der Aufgabe und Testmöglichkeiten

Der Ablauf der implementierten Lösung in Kurzfassung:

- Die Berliner Volksfest-Daten werden über den integrierten Scheduler im Hintergrund importiert
- Für jedes Event wird, falls die ID noch nicht vorhanden ist, eine neue Seite angelegt (auf die Prüfung von Datenänderungen wurde aus Gründen der Einfachheit verzichtet)
- Über die in Magnolia's STK vorhandenen Templates ("Event Overview" und "Event") werden die beim Import erstellten Event-Seiten bereits in einer Übersicht angezeigt
- Eine Komponente für die Datumsauswahl mit Kalender-Widgets wurde neu erstellt und zum "Event Overview Template" hinzugefügt
- Die im STK vorhandenen Templates / Komponenten wurden so angepasst, dass sie die notwendigen Anpassungen für die speziellen Berliner Events enthalten (dabei musste nur das Delta der bereits bestehenden Templates konfiguriert werden)
- Zusätzlich wurde noch eine Content App für Events erstellt, die Daten hierfür werden optional beim Import erzeugt (Showcase eine möglichen Alternativlösung mit dem Magnolia 5 App Framework)

### Aufbau des Projektes im Quellcode

Magnolia Projekte werden in aller Regel auf Basis von Maven realisiert. Das Magnolia CMS ist extrem modular aufgebaut, so dass der eigentliche Kern schlank bleibt und nur die notwendigen Komponenten hinzugefügt werden müssen.

Auch beim Vorliegenden Projekt wurde ein Modul entwickelt, das alle implementierten Templates und Code enthält und die notwendigen Dependencies definiert.

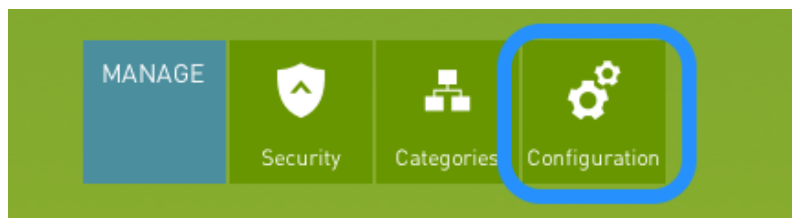
Hat man Maven auf dem eigenen Rechner installiert, kann man sich das Projekt mit dem Befehl "mvn clean install" auf der Shell zusammenbauen lassen und erhält so auch die enthaltenen Artefakte (in diesem Fall auch eine komplette Web-Anwendung).

### Scheduler

Die Daten der Volksfeste in Berlin werden via "*Command*" (spezielle Java Klasse) und den integrierten Scheduler ins Magnolia CMS importiert.

Die Java Klasse für den eigentlichen Import heisst: **EventImporterCommand.java**

Der Import kann über die Konfiguration angepasst / gesteuert werden. Die Konfiguration erreicht man nach der Anmeldung über Auswahl der Konfigurations-App





Nach Expansion des Knotens /modules/scheduler/config/jobs/importEvents sieht man die für den Import relevanten Einstellungen:

importEvents	
params	
contentPath	/2013
contentRepository	events
importContent	true
importURL	http://www.berlin.de/sen/wirtschaft/service/maerkte/strassenfeste/index.php/index/all.json?q=
path	/linux-magazin/lm-events
repository	website
active	true
catalog	linux-magazin
command	importEvents
cron	0 0 11 1/1 * ? *
description	Import events from Berlin Open Data interface

### Erklärung der Konfiguration

<b>active</b>	gibt an, ob der Job derzeit aktiv ist (Werte: true / false)
catalog	Angabe für den Namensraum des Commands
command	Name es referenzierten Commands aus dem Linux Magazin Modul
<b>cron</b>	definiert, wie oft der Job ausgeführt wird
description	Beschreibung
params/contentPath	Ablage-Ordner in der Content-App für die Events
params/contentRepository	Workspace (Speicherort im Repository) wo die Daten der Content-App liegen
params/importContent	gibt an, ob die Events aus dem Import auch in der Content-App gespeichert werden sollen (true / false)
params/importURL	URL für den Datenimport über die REST-Schnittstelle
params/path	Speicherort für die durch den Import angelegten Webseiten
params/repository	Workspace für Websites

Die wichtigsten Parameter für den Import sind active und cron. Die Angabe "0 0 11 1/1 \* ? \*" bedeutet, dass der Import-Job jeden Tag einmal um 11:00 Uhr laufen soll.

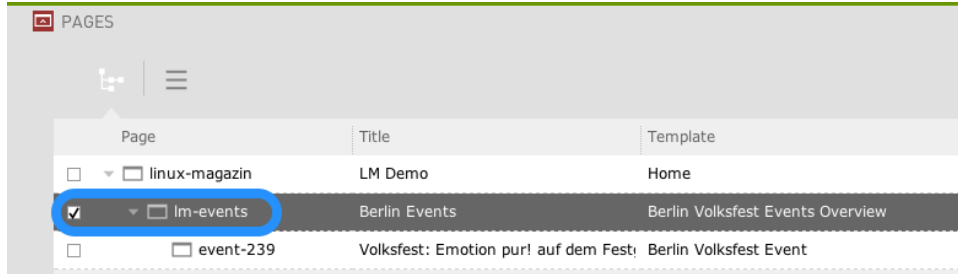
Tipp: Unter der Adresse <http://www.cronmaker.com> kann man sich menügesteuert eigene Cron-Kommandos generieren lassen. Die Verwendung von "0 0/5 \* 1/1 \* ? \*" lässt z. B. den Job alle 5 Minuten laufen.

Falls man den Job ausschalten will, einfach den Wert für "active" auf "false" setzen (Doppelklick auf den Wert und Eingabe).

## Testen des Daten-Imports

Standardmäßig verfügt die Testanwendung bereits über einige Daten. Möchte man den kompletten Import-Vorgang einmal durchspielen, kann man die vorhandenen Daten jedoch problemlos löschen.

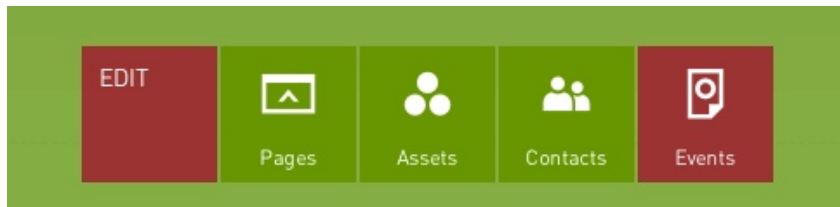
### Webseiten



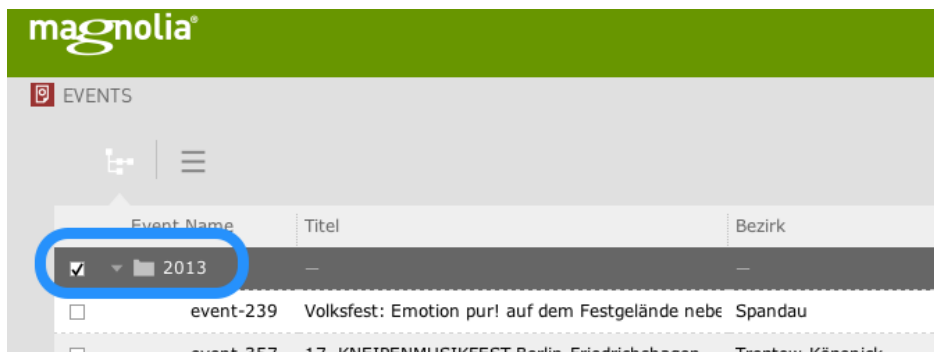
Der "lm-events" - Knoten kann gelöscht werden, er wird dann anschliessend vom Import wieder neu angelegt (falls noch nicht vorhanden).

**Wichtig:** In der vorliegenden Konfiguration muss der Knoten für "linux-magazin" vor dem Import bereits vorhanden sein.

### Events Content App



Möchte man ggf. importierte Daten entfernen, kann man einfach den Ordner "2013" löschen.

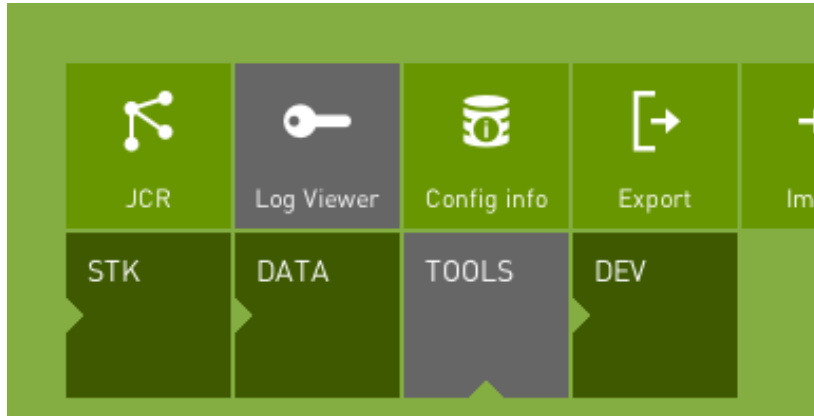


Dann ggf. die Angaben für den Parameter cron in der Scheduler-Konfiguration anpassen (siehe Abschnitt Scheduler), damit der Job schneller zur Ausführung kommt. Zudem muss der Parameter "active" den Wert "true" haben.

Falls alles gut gegangen ist, sollte anschliessend für jedes Event eine Seite erzeugt worden sein und, falls so konfiguriert, auch die Daten für die Event App.

Führt man den Job wiederholt aus, werden nur noch Events, deren ID nicht im System gefunden wurde, neu hinzugefügt.

Den Status von Jobs bzw. das Systemlog kann man entweder auf der Konsole einsehen, falls man den Tomcat Server entsprechend gestartet hat (z. B. mit `./catalina.sh run`) oder mit der Log Viewer App:



Anschliessend das Debug-Log auswählen:

