
LWL django CMS

Das Redaktionssystem des LWL für Internet- und Intranetauftritte

Übersicht über die Infrastruktur und den Entwicklungsprozess

LWL django CMS

Das Redaktionssystem des LWL für Internet- und Intranetauftritte

Übersicht über die Infrastruktur und den Entwicklungsprozess

Stand: 06.07.2018

Version: 1.2

Landschaftsverband Westfalen-Lippe
LWL.IT Service Abteilung

Autoren: Detlef Lobmeyer, Karsten Rößmann

Angaben zum Dokument

Verantwortlich	LWL.IT Service Abteilung, Web CC	
Erstellt am	11.07.2017	
Zuletzt geändert	02.08.2018 09:23	
Zustand	<input type="checkbox"/>	in Bearbeitung
	<input type="checkbox"/>	vorgelegt
	<input checked="" type="checkbox"/>	fertig gestellt
Dokumentablage	V:\Referat3\Web-CC\lobmeyer\Vertragliches\WebDesignFrontendEntwicklung2018\InfrastrukturLWLdjan goCMS.docx	

Änderungsverzeichnis zum Dokument

Änderung			Geänderte Kapitel	Beschreibung der Änderung	Autor
Nr.	Datum	Version			
1	11.07.2017 13.07.2017	0.1	-	Initiale Produkterstellung	Lobmeyer
2	22.08.2017	0.2	2, 3	Inhaltliche Ergänzungen	Hagenberger, Hötzel, Lobmeyer, Rößmann, Wischer
3	25.08.2017	1.0	3, 4	Inhaltliche Ergänzungen und redaktionelle Schlussbearbeitung	Hagenberger, Lobmeyer, Rößmann, Wischer
4	02.08.2018	1.1	4.1	Probleme mit Windows und der Docker-Toolbox dokumentiert	Wischer
5	14.02.2018	1.1	2.2	Ausführungen zu Bootstrap und Raster	Rößmann
6	06.07.2018	1.2	2.2, 3.1, 3.2	Build-Prozess und Bootstrapverwendung detailliert	Hagenberger, Hötzel, Lobmeyer, Rößmann, Wischer

Inhaltsverzeichnis

1	Ausgangslage, Gegenstand und Ziel.....	1
1.1	Ausgangslage beim LWL.....	1
1.2	Gegenstand und Ziel	1
2	Infrastruktur	1
2.1	Redaktionskomponenten.....	1
2.2	Frontend-Entwicklung.....	1
2.3	Betriebs-Infrastruktur	3
3	Entwicklungsprozess.....	3
3.1	LWL django CMS (Systemkern)	3
3.2	Änderung und Neuerstellung von Frontend-Elementen.....	4
3.2.1	Bestehende Standard-Designvorlagen.....	4
3.2.2	Bearbeitung von Frontend-Elementen.....	4
3.2.3	Entwicklung neuer Frontend-Elemente.....	4
3.2.4	Methodisches Vorgehen.....	4
4	Softwareübergabe.....	5
4.1	Voraussetzungen für die Übergabe.....	5
4.2	Übergabe	6

Landschaftsverband Westfalen-Lippe LWL.IT Service Abteilung	LWL django CMS Infrastruktur und Entwicklungsprozess	Stand: 06.07.2018 Version: 1.2
---	---	-----------------------------------

1 Ausgangslage, Gegenstand und Ziel

1.1 Ausgangslage beim LWL

LWL django CMS ist das um LWL-spezifische Anforderungen ergänzte Open-Source-Produkt django CMS. Etwa 430 Redakteurinnen und Redakteure verwalten mit der Redaktionskomponente (Stand Juli 2018) 180 Internetauftritte mit rund 28.000 Seiten. Rund 210 Benutzer der Redaktionskomponente sind für die Pflege von 36 Intranetauftritten registriert. Ferner wird das LWL-Intranet für unterschiedliche Formen der LWL-internen Kommunikation genutzt. Das LWL-Intranet besteht aus ca. 3500 Seiten.

Die LWL.IT Service Abteilung (LWL.IT) betreibt und pflegt das Content Management System mit Redaktionssystem (Backend), Datenhaltung und Frontend-Komponente sowie die auftrittsspezifischen Grundlagen und Rahmen.

1.2 Gegenstand und Ziel

Dieses Dokument beschreibt die Software LWL django CMS sowie die im Umfeld genutzten Software-Werkzeuge aus frontendspezifischer Sicht. Es soll bei der Kalkulation und Erbringung von IT-Dienstleistungen unterstützen.

2 Infrastruktur

2.1 Redaktionskomponenten

LWL django CMS bietet folgende Redaktionskomponenten:

- Browserbasierte Redaktionsoberfläche
- Benutzer- und Zugriffsrechteverwaltung
- LWL-CD-konforme Frontend-Ausgaben

Per integrierter Export-Funktionalität können Inhalte einzelner Seiten exportiert werden. Eine weiter gehende Schnittstelle zum Entladen von Daten wird aktuell entwickelt.

Eine Seitenzugriffs-Statistik ist angebunden. Die Verwendung der Software Matomo (vormals Piwik) ist beim LWL obligatorisch.

2.2 Frontend-Entwicklung

Für die Anzeige der Webseiten kommen django CMS-Standard-plugins sowie LWL-spezifische plugins unter Verwendung folgender Sprachen und Werkzeuge zum Einsatz:

- HTML5
- CSS3
- django template language
- JavaScript (jquery)
- HTML-Framework Bootstrap (V 3.3.7, Stand: 02.08.18) mit

Landschaftsverband Westfalen-Lippe LWL.IT Service Abteilung	LWL django CMS Infrastruktur und Entwicklungsprozess	Stand: 06.07.2018 Version: 1.2
---	---	-----------------------------------

- Bootstrap-Klassen
- Bootstrap-Grid
 - 24 Spalten mit je 30 Pixeln Abstand zwischen den Spalten, nämlich 15 am linken und 15 am rechten Spaltenrand
 - durch Angabe der Spaltenzahl (als $n/24$) ergibt sich die jeweilige Spaltenbreite als prozentualer Anteil, um einer responsiven Ausgabe Rechnung zu tragen.
- Bootstrap-Komponenten (bestimmte Komponenten werden bereits per django-bootstrap-plugins bereitgestellt, z. B. Karussell, Akkordeon, Button)
- SASS als Stylesheet-Sprache, mit SCSS-Syntax

Mit den build-tools

- gulp
- babel
- webpack
- und vorhabenspezifisch weiteren tools (mit der LWL.IT abzustimmen)

werden die mit

- javascript
- SCSS / CSS

entwickelten Quellen zusammen gebunden.

Web-Grafiken werden in einzelnen Sprite-Dateien organisiert. Diese werden mit den o. a. build-tools zu einer gemeinsamen Sprite-Datei zusammengefasst.

Runtime-Umgebung für die build-tools ist node.js .

Anzuwenden sind jeweils die im LWL-Projekt verwendeten Versionen.

Der LWL entwickelt Websites mit iterativen Entwicklungsprozessen responsiv mit dem mobile-first-Ansatz. Um die Responsivität zu gewährleisten ist das LWL django CMS wie oben beschrieben via Bootstrap auf einem 24er Raster (entspricht 24 Spalten) implementiert. Die Standardbreite beträgt 1280px. Im Standardlayout bestehen der linke Navigationsbereich und der rechte Marginalbereich jeweils aus fünf Spalten mit den zugehörigen 15px-Abständen. Beispiel: [Internetauftritt der LWL.IT](http://www.lwl.it) (www.lwl.it)

Inhalte und Gestaltung sind im LWL django CMS strikt getrennt. Zur Verwaltung des kompletten Quellcodes inkl. der Gestaltungselemente wird das Versionsverwaltungssystem Git genutzt.

Für Entwicklungsarbeiten im CMS (Backend-Komponente, Frontend-Komponente) wird eine Docker-Umgebung benutzt, in der ein lokaler Datenbank-Server und der django-Web-Server laufen. Geänderter Code wird direkt in dem Docker-Container des django Servers ersetzt, wodurch ein schneller Arbeitszyklus besteht.

An das LWL django CMS sind andere Anwendungen des LWL (z. B. Veranstaltungskalender, Presseforum, Newsletter) über anwendungsseitige Schnittstellen angebunden. Vorgegebene LWL-Standards sind:

- SOLR-Suche

Landschaftsverband Westfalen-Lippe LWL.IT Service Abteilung	LWL django CMS Infrastruktur und Entwicklungsprozess	Stand: 06.07.2018 Version: 1.2
---	---	-----------------------------------

- für die Präsentation von Veranstaltungen: LWL-Veranstaltungskalender
- für die Verbreitung von Pressemitteilungen: LWL-Presseforum
- als Newslettersystem: newsletter2go
- als Blogsystem: Aldryn-Blog
- für die Präsentation von Karten (Geodaten):.djangocms-googlemaps

Abweichungen von diesen Standards bedürfen der Zustimmung der LWL.IT Service Abteilung.

2.3 Betriebs-Infrastruktur

Sowohl im Internet als auch im LWL-Intranet wird LWL django CMS in zwei Systemumgebungen bereitgestellt, einer Fachtestumgebung und einer Produktionsumgebung.

LWL django CMS wird für das LWL-Intranet betrieben mit:

- 3 virtuellen Linux-Systemen als Applikationsserver
- 2 virtuellen Linux-Systemen als Cluster mit Apache als Reverse-Proxy
- ein MariaDB/Galera-Cluster auf 3 virtuellen Linux-Systemen als RDBMS für DjangoCMS
- ein GlusterFS-Fileserver auf 2 virtuellen Linux-Systemen als Fileserver für Bilder/Dokumente/etc., die nicht in der django CMS-Datenbank abgelegt sind.

LWL django CMS wird für das Internet betrieben mit:

- 3 virtuellen Linux-Systemen als Applikationsserver
- 2 virtuellen Linux-Systemen als Cluster mit Apache als Reverse-Proxy
- ein MariaDB/Galera-Cluster auf 3 virtuellen Linux-Systemen als RDBMS für DjangoCMS
- ein GlusterFS-Fileserver auf 2 virtuellen Linux-Systemen als Fileserver für Bilder/Dokumente/etc., die nicht in der django CMS-Datenbank abgelegt sind.

3 Entwicklungsprozess

3.1 LWL django CMS (Systemkern)

An dieser Stelle werden in Abgrenzung zum nachfolgenden Kapitel die Entwicklungsarbeiten am Systemkern behandelt.

Die Software django CMS wird von der entsprechenden Open Source-Entwicklergemeinschaft gepflegt und weiterentwickelt. Die LWL.IT selbst oder ein entsprechender Vertragspartner der LWL.IT erledigen das für die LWL-spezifischen Systemergänzungen.

Landschaftsverband Westfalen-Lippe LWL.IT Service Abteilung	LWL django CMS Infrastruktur und Entwicklungsprozess	Stand: 06.07.2018 Version: 1.2
---	---	-----------------------------------

Software-Änderungen und -Ergänzungen jeglicher Art werden nach Entwicklung

- aus der jeweiligen lokalen Entwicklungsumgebung im zentralen beim LWL betriebenen Git-Repository (lwlgitlab) zusammengeführt
- bei Bedarf in die Fachtestumgebung übernommen
- und schließlich nach entsprechenden Tests und Freigabeprozessen von der LWL.IT Service Abteilung in die Produktionsumgebung überführt.

3.2 Änderung und Neuerstellung von Frontend-Elementen

3.2.1 Bestehende Standard-Designvorlagen

Den CMS-Redakteurinnen und Redakteuren stehen für den Aufbau und für die Gestaltung von Web-Auftritten bzw. Web-Seiten Standard-Designvorlagen zur Verfügung, die ein einheitliches Erscheinungsbild sicherstellen, die Design-Kosten niedrig halten und den Aufbau neuer Web-Auftritte beschleunigen sollen.

3.2.2 Bearbeitung von Frontend-Elementen

Mit den in [Kap. 2.2](#) beschriebenen Sprachen und Werkzeugen können Aussehen und inhaltliche Logik eines existierenden Elements bearbeitet werden.

3.2.3 Entwicklung neuer Frontend-Elemente

In Einzelfällen erfordert eine individuelle Anforderung einer LWL-Einrichtung (z. B. LWL-Museum) oder einer LWL-nahen Einrichtung die Entwicklung eines neuen Elements oder im Ausnahmefall eine neue Designvorlage, die ggf. auch für weitere Auftritte wiederverwendet werden soll.

Auch in diesem Fall sind die in [Kap. 2.2](#) beschriebenen Sprachen und Werkzeuge zu verwenden.

Sollte im Zuge der Entwicklung ein neues Plugin oder eine neue Funktionalität erforderlich werden, wird nach Kap. 3.1 verfahren.

3.2.4 Methodisches Vorgehen

In einer lokalen Entwicklungsumgebung werden neue Frontend-Elemente erzeugt bzw. bestehende Frontend-Elemente bearbeitet. Im Regelfall werden kleinere Anpassungen (im Sinne von [Kap. 3.2.2](#)) durch die LWL.IT vorgenommen.

Bei umfangreicheren Arbeiten an Frontend-Elementen wird ein Branch gebildet, der nach Abschluss mit dem Masterbranch zusammengeführt wird.

Frontend-Elemente werden immer im LWL django CMS bearbeitet. Nachfolgend sind die Schritte für ein iteratives Vorgehen bei der Umsetzung einer komplexeren Anforderung genannt:

Nr.	Vorgang
1.	Konzeption der Website (z.B. Branch erforderlich?)
2.	Programmierung fehlender Funktionalitäten

Landschaftsverband Westfalen-Lippe LWL.IT Service Abteilung	LWL django CMS Infrastruktur und Entwicklungsprozess	Stand: 06.07.2018 Version: 1.2
---	---	-----------------------------------

3.	Einrichten des Branch mit den notwendigen Frontendelementen
4.	Erstellung einer Low-Fidelity-Testversion mit allen notwendigen Seitentypen in dem Branch, bzw. Anpassung von Elementen
5.	Test der Low-Fidelity-Version bzw. der Elemente; Freigabe (für weitere Schritte) nach zufriedenstellendem Testergebnis
6.	Fertig getestete und freigegebene Komponenten/Elemente gestalten
7.	gestaltete Komponenten/Elemente insgesamt erneut testen; Freigabe (für weitere Schritte) nach zufriedenstellendem Testergebnis
8.	Abschließendes Design
9.	erzeugte High-Fidelity-Version bzw. angepasste Elemente abnehmen
10.	Zusammenführen der Entwicklung mit dem Masterbranch
11.	abschließende QS
12.	Fachliche und technische Freigabe

Anmerkung zur obigen Tabelle: Die Schritte 4. und 5. sowie 6. und 7. sind iterativ, d. h. sie werden im Zuge der Tests ggf. mehrfach durchlaufen.

Gründe für dieses Vorgehen sind:

- Einheitliche Entwicklungsumgebungen der an der CMS-Entwicklung beteiligten Akteure
- Einhaltung der IT-Sicherheitsvorgaben des LWL
- Einhaltung der CMS-Standards
- Einhaltung der angewendeten technologischen Standards
- Vermeidung von Mehraufwänden bei der Übernahme des Frontend-Codes
- Vermeidung doppelter Entwicklungsaufwände (z.B. wegen eines statischen HTML-Prototyps)

4 Softwareübergabe

4.1 Voraussetzungen für die Übergabe

Benötigte Tools in der lokalen Entwicklungsumgebung sind:

- Docker,
- Git,
- Code-Editor.

Unter Windows in den Versionen < 10 ist die Docker-Umgebung „Docker-Toolbox“ nicht nativ lauffähig, da diese das Einbinden von Ordnern in den Docker-Container nicht korrekt durchführt. Die LWL.IT empfiehlt daher die Verwendung von Microsoft Windows 10 oder Apple MacOS. Bei beiden kann die Docker-Integration nativ verwendet werden.

Die LWL.IT räumt externen Entwicklern den Lesezugriff für das Kopieren der aktuellen Quellen (Klon des Repositories) ein. In den Quellen sind sämtliche Konfigurationsdateien enthalten, um einen lokalen Django-Webserver innerhalb eines Docker-Containers zu starten.

Landschaftsverband Westfalen-Lippe LWL.IT Service Abteilung	LWL django CMS Infrastruktur und Entwicklungsprozess	Stand: 06.07.2018 Version: 1.2
---	---	-----------------------------------

4.2 Übergabe

Der geänderte bzw. erzeugte Quellcode wird via Merge-Request übergeben.
Anschließend wird der übergebene Code durch die LWL.IT in den Masterbranch überführt.