

Drupal 8: Modul-Kombination für Basis-Website

Um unseren Kunden - möglichst Budget schonend und schnell - einen Rahmen für eine Webseite bieten zu können, die den modernen Ansprüchen an Performance, Responsibility, Schnittstellen zu Social Media etc. gerecht wird, haben wir in den letzten Monaten eine Art Dummy-Version unter Drupal 8 erstellt, die mit einem Paket an Modulen ausgestattet ist.

Nach zwei Jahre Erfahrung mit eigenen Projekten und Kundenprojekten unter Drupal 8 sind wir sehr begeistert über das smarte Handling und sind der Meinung, dass davon nicht nur Enterprise Projekte profitieren sollten.

Inzwischen stehen zahlreiche Module in stabiler Version zur Verfügung, so dass solche Anpassungen in der Regel nicht notwendig sind.

Falls doch, lassen sich schnell, sicher und konform zu Standards Extra-Wünsche mit eigenen kleinen Modulen oder Theme-Funktionen und Twig-Templates erfüllen.

Das Ansinnen, so ein Paket mit sinnvollen Modulen zu bieten, ist ja nicht neu, zahlreiche sogenannte Distributionen haben den Anspruch, genau das zu erfüllen.

So haben wir z.B. die Distribution Varbase installiert und ausführlich getestet.

Wie es aber bei Distributionen häufig der Fall ist: Sie können viel mehr, als man tatsächlich braucht.

Aus Gründen zügiger Updates, Performance, Übersichtlichkeit in der Verwaltung und Sicherheit des Systems ist es aber wünschenswert, nur solche Module zu verwenden, die auch wirklich gebraucht werden.

Varbase verwendet auch teilweise gepatchte Module, die noch im Alpha- oder Beta-Stadium vorliegen, und garantiert somit die Stabilität. Teilweise kommen auch selbst programmierte Erweiterungen zum Einsatz, um zusätzliche Features zu liefern.

Damit begibt man sich aber in eine große Abhängigkeit von einem Anbieter, die wir gerne vermeiden möchten.

Langjährige Erfahrung mit unseren Kunden und aus der Zusammenarbeit mit Partner-Agenturen, zeigen, dass die Ansprüche fast immer ähnlich sind. Daraus resultiert die folgende Liste an Modulen, die sich bewährt haben. Basierend auf diesem Paket haben wir auch unsere Webseite zum Jahresanfang 2019 auf Drupal 8 migriert, die allerdings bewusst einfach gehalten ist und bei weitem nicht alle Funktionen nutzt.

Universelle Module, die teilweise Voraussetzung für andere Module sind.

- **Chaos tool suite:**
Ctools ist eine Ansammlung von APIs (Programmierschnittstellen), die Code für Entwickler und Bibliotheken für andere Module zur Verfügung stellen. Viele Drupal-Module sind davon abhängig.
- **Entity:**
Das Modul erweitert die Entity-API vom Drupal Core und ermöglicht so einen einheitlichen Umgang mit den Eigenschaften der Entities (Nodes, User etc.)
- **Entity Browser:**
Das Modul ermöglicht das Browsen durch vorhandene Entities (Medien, Content, Dateien) und die Selektion, z.B. im WYSIWYG
- **Entity Browser Enhanced:**
Erweitert den Entity Browser z.B. so, dass Multiselect von Medien möglich wird.
- **Libraries API:**
Ermöglicht die Verwaltung externer Libraries.
- **Tokens:**
Erweitert die Tokens, die bereits im Drupal Core mit geliefert werden. Tokens erlauben z.B. die Erstellung suchmaschinenfreundlicher URL's, aber noch viel mehr.

Module für bessere Verwaltung des Backends:

- **Admin Toolbar:**
Erweitert die Toolbar des Cores (also die obere Menüleiste im Backend), so dass ein DropDown Menü schnelleren Zugriff erlaubt.

Einfache Pflege der Inhalte:

Drupal bietet im Core schon immer Werkzeuge, mit denen sich Inhalte gut strukturiert pflegen lassen.

Darüber hinaus besteht oft das Bedürfnis, Page Builder Funktionen zu haben, wie Redakteure sie auch von Wordpress oder ähnlichen Systemen gewöhnt sind.

Hierfür bieten sich die Module Display Suite, Paragraphs und Panels an.

Je nach Ausrichtung des Projektes oder evt. Erfahrung / Wünsche der Redakteure, wählen wir eines der mächtigen Module:

- **Page Manager:**
Vorraussetzung z.B. für das Modul Panel
- **Panel:**
Das Modul Panels ermöglicht es dem Administrator, benutzerdefinierte Layouts zu erstellen. Es ist eine Art Drag & Drop-Content-Manager, mit dem man ein Layout visuell gestalten und Inhalte in diesem Layout platzieren kann.
- **Paragraphs:**
Stellt vordefinierte Paragraphen-Typen zur Verfügung, aus denen der Redakteur bei der Pflege einer Inhalts-Seite beliebig oft wählen kann, statt alle Inhalte im Body Feld pflegen zu müssen.
So erreicht man eine einheitliche Struktur mit beliebig vielen Blöcken, die auch so komplex sein können, wie Slider, Galerien, Akkordeons und vieles mehr.
- **Display Suite:**
Ein Drag + Drop Interface erlaubt die Anordnung beliebiger Inhalte in vordefinierten oder benutzerdefinierten Templates.

Wer tiefer in die Welt des Layout Buildings unter Drupal 8 einsteigen möchte, findet hier eine tolle Aufstellung mit Vergleich der drei genannten Module.

<https://www.electriccitizen.com/blog/layout-drupal-8-part-2>

Auf unserer Firmen Homepage verwenden wir Paragraphs in Kombination mit [Bootstrap Paragraphs](#).

Erweiterte Versionen-Kontrolle

- **Revision Log Default:**
Erzwingt das Befüllen eines Revision Logs, welches Aufschluss gibt über die

Änderungen, die ein Redakteur an der Seite vorgenommen hat.
Sehr hilfreich, wenn mehrere Personen an gleichen Inhalten arbeiten.

Responsive Menüs

- [Superfish](#)
Integriert die Superfish-Library und ermöglicht responsive und Touchscreen-fähige Dropdown-Menüs.

Praktische Erweiterung für Blöcke

- [Menu Block:](#)
Damit kann man aus vorhanden Menüs Blöcke erstellen, die dann an beliebiger Stelle eingebaut werden können.
- https://www.drupal.org/project/block_class
Erlaubt das hinzufügen von Klassen zu Blocks , was vor allem im Zusammenhang mit den Hilfsklassen vom Bootstrap Menü sehr praktisch ist.

Den Standard-Editor CKEditor erweitern

Neben Modulen aus der Media-Serie, um z.B. die Einbettung von Videos oder Verlinkung auf der Seite zu vereinfachen, sind außerdem hilfreich:

- [CKEditor Anchor Link:](#)
Liefert einen Dialog zur Auswahl von Anker für die Navigation innerhalb der Seite
- https://www.drupal.org/project/ckeditor_media_embed
Fügt einen Dialog hinzu, mit dem man Ressourcen von Drittanbietern (Videos, Tweets, Bilder) einbetten kann.
- [Embed:](#)
Ein Framework Modul mit dem man verschiedene Dinge im Drupal 8 WYSIWYG Editor einbetten kann. Davon sind weitere Module abhängig, wie z.B:
- [Entity Embed:](#)
erlaubt die Einbettung von Entities im WYSIWYG Editor.
Der Entity Browser ist ein verwandtes Modul mit größerer Flexibilität, aber

häufig nicht notwendig. Dann ist Entity Embed ein Ersatz.

- [Linkit](#):
Bietet ein einfaches Interface zum Einbinden interner oder externer Links.
- [D8 Editor Advanced link](#)
Erweitert die Core Möglichkeiten zum Hinzufügen von Attributen bei Links

Entwickler-Tool

Fast immer installieren wir das Modul Devel, das u.a. Aufschluss über Variablen gibt, die in Hooks und eigenen Funktionen verwendet werden können

- [Devel](#)

Medien-Verwaltung:

Unter Drupal 8 steht ein ganzes Set an Modulen zur Verfügung, das inzwischen auch stabil vorliegt. Sie bauen auf Media Entity auf, das seit Drupal 8.4 im Core eingebaut ist.

- [Colorbox](#):
Das Modul bindet die Library Colorbox ein, mit der benutzerdefinierte Lightbox erstellt werden können. Basiert auf jQuery.
- [Dropzonejs](#):
Erlaubt über die gleichnamige Library das Hochladen beliebig vieler Dateien mit Drag&Drop und bietet dabei auch eine Vorschau.
- [IMCE](#):
Altbekannter Image/File Upload Browser, der auch die privaten Ordner einzelner User berücksichtigt und dabei grundlegende Datei-Operationen ermöglicht.
- [Video Embed Field](#)
Erstellt einen eigenen Feld-Typen, mit dem Videos von Youtube oder Vimeo eingebaut werden können.
- [Flexslider](#):
Bindet die Flexslider Library für responsive Slideshows ein, mit diversen Möglichkeiten, die Animation zu konfigurieren, die auch in Touch-Medien funktionieren.

Zusätzliche Module erlauben die Interaktion mit Views etc.

Das Modul ist mit heutigem Stand noch in der Beta-Version, aber wir haben es in mehreren Projekten im Einsatz und sind sehr zufrieden.

Durchsuchbare und selektierbare Darstellung der Inhalte.

Die Werkzeuge Views für Reports, gefilterte Listen, Blöcke etc. sind selbstverständlich unter Drupal 8 nach wie vor vorhanden. Views ist in das Core gewandert.

Für die Suche auf der Seite, die nicht mit Views passiert, verwenden wir diese beiden Module:

- [Search API](#): mit [Search APIS Pages](#):

Sie ersetzen das Core Modul Search, das nicht so flexibel ist.

Die Search API kann auf vielfältige Weise erweitert und angepasst werden und eröffnet alle Möglichkeiten für künftige Bedürfnisse und auch wachsende Ansprüche an die Performance, z.B. Suche mit Solr statt Datenbank.

Kommunikation mit den Besuchern - aber bitte ohne Spam

- [Webform](#):

Auch wenn das Contact-Formular im Drupal Core die schnelle Erstellung von Formularen erlaubt, so nutzen wir inzwischen doch bei allen Projekten das Modul Webform, das jetzt auch stabil vorliegt.

Damit lassen sich (Mail-)Templates flexibler umsetzen und wer schon mal versucht hat, mit Contact eine Email-Benachrichtigung an den Absender zu generieren, wird schmerzhaft feststellen, dass diese nur für angemeldete Besucher möglich ist.

Deshalb setzen wir lieber von Anfang an die mächtigere Variante ein, um später flexibel zu sein.

- [Honeypot](#):

Kein Formular ohne Honeypot - ist unsere Devise.

Honeypot ist die benutzerfreundliche Alternative zu Captchas, welche ich auch gerne als Conversion-Verhinderungs-Tool bezeichne. ;-)

SEO Module, damit die Webseite auch gefunden wird:

- **Metatag:**
Mit dem Meta-Tag Modul kann die Pflege von Meta-Tags für Suchmaschinen, Social Media etc. einerseits mit Hilfe von Tokens automatisiert werden. Gleichzeitig können auf verschiedenen Ebenen der Inhalts-Hierarchie auch individuell abweichende Daten eingegeben bzw. Tokens definiert werden.
- **Pathauto:**
generiert automatisch suchmaschinenfreundliche URL's auf Basis benutzerdefinierter Muster, mit Hilfe von Tokens
- **Xmlesitemap:**
Das XML-Sitemap-Modul erstellt eine Sitemap, die der Spezifikation sitemaps.org entspricht. Diese kann automatisch an Ask, Google, Bing (ehemals Windows Live Search) und Yahoo! übermittelt werden. Es kann sehr flexibel eingestellt werden, welche Inhalte automatisch in die Sitemap aufgenommen werden. Bei sehr umfangreichen Projekten (Produktkataloge mit mehreren 10000en Inhalten hatten wir in der Vergangenheit Performance Probleme. Hier kann es Sinn machen, auf das externe Tool <https://www.xml-sitemaps.com/news-20180517.html> zurück zu greifen, das via Cronjob auf dem Server automatisiert ausgeführt werden kann. Das entscheiden wir von Fall zu Fall, je nach Umfang der zu erwartenden Inhalte. In den meisten Fällen ist das Modul eine gute Wahl.

Social Media zur Internaktion mit Fans:

- **Addtoany:**
Liefert Share Buttons für Facebook, Twitter, Google+, Pinterest, WhatsApp und viele mehr.

Analyse-Tools zur Erfolgskontrolle

wie z.B.

- **Google Analytics:**
Fügt das Webstatistik-Tracking-System von Google Analytics zur Website hinzu.

Basis-Theme zur Umsetzung schöner Designs:

Wir arbeiten inzwischen sowohl bei Wordpress, als auch bei Drupal fast ausschließlich mit Bootstrap als Basis-Theme.

Um zukunftsfähig zu sein und die hilfreichen Funktionen von Bootstrap 4 zu nutzen, verwenden wir seit Neuestem das Theme

[Bootstrap Barrio:](#)

Je nach Kundenwunsch und Größe des Projekts wird es ohne SASS und stattdessen mit der

[Bootstrap Library:](#) in Kombination mit der vorkompilierten Bootstrap Library

Oder aber die Version für SASS, welches wir auf der lokalen Entwicklungsumgebung eingerichtet haben.

[Bootstrap Barrio mit SASS](#)

Dazu an anderer Stelle demnächst mehr.

Alle aufgeführten Module erfordern nicht zwingend eine Composer-Installation.

Sie können also auch bei Hostern installiert werden, deren Konsolen-Unterstützung nicht ausreicht, sei es, weil kein Zugriff existiert oder keine performante PHP Version auf der Konsole läuft (Beispiel 1&1).

Wir empfehlen dennoch, von Anfang an eine Composer unterstützte Installation und ebensolche Updates, da es erfahrungsgemäß schwieriger ist, die Projekte zu einem späteren Zeitpunkt unter Composer-Kontrolle zu nehmen, wenn dann doch Module, wie z.B. address gebraucht werden, die zwingend Composer zur Verwaltung der Libraries erfordern. Normalerweise ist diese Methode auch effizient.

Wenn bei einem Hoster gehostet werden soll, der keine Composer Installation unterstützt, dann installieren wir unter unserem Webespace und holen das Projekt für Updates auf unseren eigenen Webespace, was allerdings mit Zusatzkosten verbunden ist.

Auf unserem Webespace halten wir künftig immer eine Version mit den genannten Modulen vor, die unter Composer Kontrolle installiert und gepflegt wird.

Diese Version mit vorkonfigurierten Benutzerrechten, Kontaktformular, Bestandteilen für Datenschutzgrundverordnung und Impressums-Pflicht kann dann jederzeit zum Kunden kopiert werden.

Nicht benötigte Module werden entfernt.