Drupal Update von 8.6.13 zu 8.9.3 unter Composer-Kontrolle

Inhaltsverzeichnis

- Probleme der Installation
  - Veraltete Module und Drupal 8 Versionen.
  - Inhomogene Dateistruktur:
  - Mismatched entity and/or field definitions
- Bestehende Version 8.6.13 umbauen
  - Update der aktuellen Datenbank und Verzeichnisse erstellen.
  - Testverzeichnis einrichten
  - Installation der Version 8.6.13 mit Composer
  - Wiederherstellen der individuellen Konfigurationen
  - Module installieren, die Composer-Kontrolle benötigen
  - Inhomogene Struktur im Modules-Ordner reparieren
  - Installation testen
  - Backup der bisherigen Version
- Schrittweises Update bis zur aktuellen Version 8.9.3
  - Zusammenfassung des Vorgangs beim Update:
  - Vorgehen bei den einzelnen Update-Schritten:
  - Spezielles Verfahren wegen pathauto ab Drupal 8.7.9
  - Update der Module vor Installation von Drupal 8.8.0
  - Umzug auf das Produktivsystem und finale Tests
- Ausblick

Viele Projekte kommen zu uns, weil der vorherige Programmierer in der Versenkung verschwunden ist. So auch in diesem Beispiel.

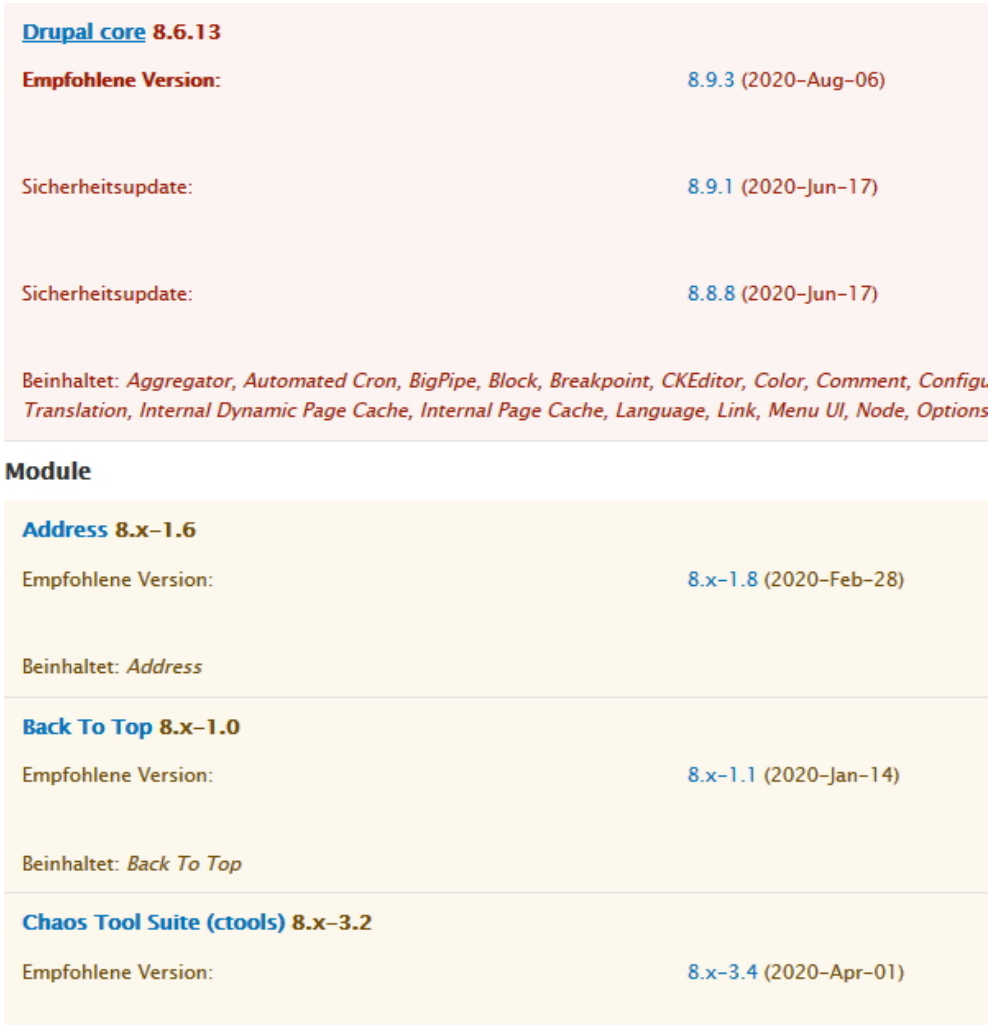
montviso
Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach
Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980
Impressum | Datenschutz

Die Webseite wurde in einem sehr frühen Stadium von D7 auf D8 migriert, war jetzt nur bis Drupal 8.6.13 aktualisiert und nicht unter vollständiger Composer-Kontrolle. Zudem gab es Probleme mit Mismaching Entity.

Abstand Pixel

30

IST-STAND



ZIEL

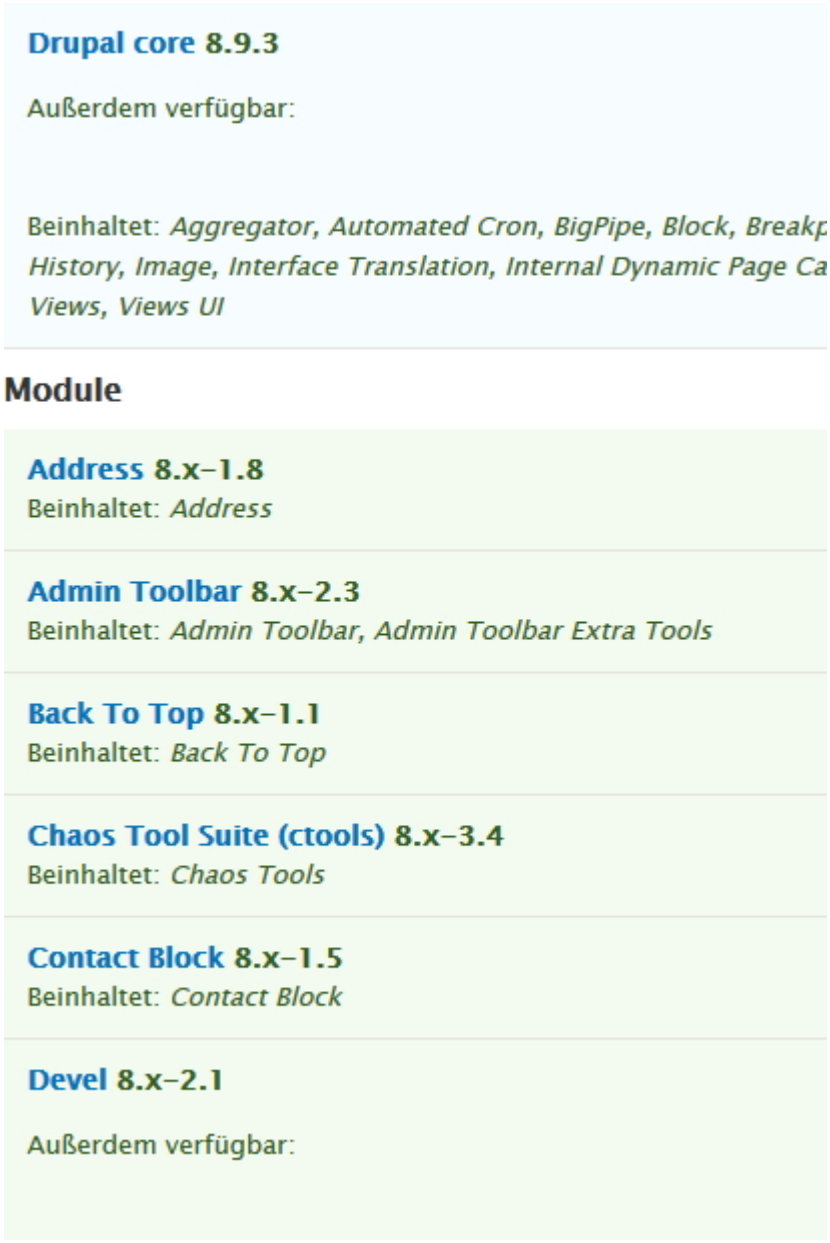
montviso

Montviso- Internetdienstleistungen

Regina Oswald
Stresemannstraße 15
DE-88400 Biberach

Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980

Impressum | Datenschutz



Abstand Pixel
30

Probleme der Installation

Veraltete Module und Drupal 8 Versionen.

Auch wenn es noch einige Zeit dauert bis zur Abschaltung von Drupal 8, konnte man natürlich die veralteten Module schon aus Sicherheitsgründen nicht so lassen.

Wir werden die Webseite wegen fehlender Module noch lange nicht auf D9 updaten, aber zumindest sollten alle Probleme aus dem Weg geräumt werden, die einem problemlosen Upgrade von der letzten D8-Version auf D9 im Wege stehen könnten.

montviso
Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach
Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980
Impressum | Datenschutz

Vor allem brannte unter den Fingern, dass das Projekt nicht unter sauberer Composer-Kontrolle war.

Inhomogene Dateistruktur:

Bis zu einer bestimmten Version von Drupal 8 war es noch möglich, die Webseite mit bekanntem Prozedere ohne Composer-Unterstützung zu erstellen und Updates zu machen.

Wer schon mit älteren Drupal-Versionen Erfahrung hat, kennt die Verzeichnisstruktur, wie sie auch unter Drupal 7 normal ist.
Alles, was nicht vom Core kommt, lag im Ordner sites, also Module, Libraries und Themes sowie natürlich die Dateien.

Module, die nicht per Composer, sondern über das Backend von Drupal mit der Ui erstellt wurden, waren nicht in der sauberen Struktur modules/contrib und modules/custom installiert, sondern lagen direkt im modules-Ordner.

Da einige Module mangels Alternativen bereits mit Composer installiert worden waren (addresses + entity_print), gab es jetzt eine bunte Mischung aus Modulen im Contrib-Ordner bzw. eine Ebene höher.

Mismatched entity and/or field definitions

Ein Blick auf die Seite www.diesedomain.de/admin/reports/status zeigt einige Fehler.

Bei einem der händischen Updates oder Update von addresses mit Composer kam es zur Fehlermeldung:

montviso

Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach

Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980

Zwar hat die Fehlermeldung nicht direkt Probleme verursacht, die Adressfelder wurden korrekt angezeigt. Aber spätere Drupal 8 Versionen ließen sich dann nicht mit Composer installieren.

Bestehende Version 8.6.13 umbauen

Der erste Schritt hat das Ziel, die aktuelle Version unter vollständige Composer-Kontrolle zu bekommen, mit einer Dateistruktur, wie sie auch bei composer create-project erzeugt wird.

Update der aktuellen Datenbank und Verzeichnisse erstellen.

Ich erwähne es, auch wenn es eigentlich selbstverständlich ist. ;-)

montviso

Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach

Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980

Testverzeichnis einrichten

Ich arbeite lokal mit Windows 10 und Apache.

- Zuerst wurde auf PHP 7.3 umgestellt, weil dies die empfohlene Version für die künftigen Versionen ist. Auch 8.6.13 läuft schon damit.
- Ich erstelle das Verzeichnis „kundexxx“ als Unterordner im Web-Root.
- In der Datei etc/host mache ich einen Eintrag
  127.0.0.1 kundexxx.localhost
  Dann kann ich die lokale Installation mit http://kundexxx.localhost aufrufen
- in der Apache-Konfiguration lege ich ein virtuelles Verzeichnis an, bei dem ich die URL kundexxx.localhost auf das Verzeichnis
  C:\wwwroot\kundexxx\web mappe.
  Wichtig: Hier auch gleich eine Log-Datei angeben, damit die Fehlermeldungen, die da kommen werden, einsehbar sind.
- In der my.ini habe ich diverse Anpassungen für die Datenbank gemacht, damit Drupal auch unter Windows halbwegs performant läuft.
- Composer und Drush waren bereits installiert und laufen zufriedenstellend.

Installation der Version 8.6.13 mit Composer

- Auf der Konsole wechsle ich in das vorher angelegte Verzeichnis
  `cd C:\wwwroot\kundexxx`
- Nun muss ich den Composer überreden, ein Projekt mit einer vollkommen veralteten Version zu erstellen.

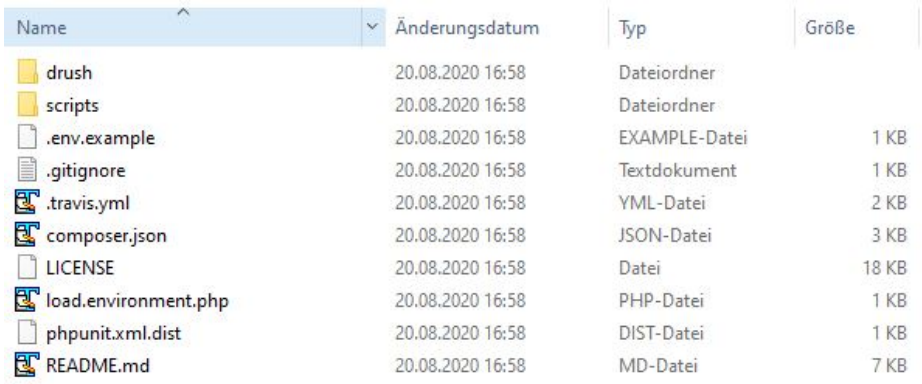
    - Dazu lege ich das Projekt erst mal an, ohne zu installieren:
      `composer create-project drupal-composer/drupal-project:8.x-dev C:\wwwroot\kundexxx --no-interaction --no-install`

      Danach sieht das Verzeichnis so aus:

montviso

Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach

Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980



Name	Änderungsdatum	Typ	Größe
drush	20.08.2020 16:58	Dateiordner	
scripts	20.08.2020 16:58	Dateiordner	
.env.example	20.08.2020 16:58	EXAMPLE-Datei	1 KB
.gitignore	20.08.2020 16:58	Textdokument	1 KB
.travis.yml	20.08.2020 16:58	YML-Datei	2 KB
composer.json	20.08.2020 16:58	JSON-Datei	3 KB
LICENSE	20.08.2020 16:58	Datei	18 KB
load.environment.php	20.08.2020 16:58	PHP-Datei	1 KB
phpunit.xml.dist	20.08.2020 16:58	DIST-Datei	1 KB
README.md	20.08.2020 16:58	MD-Datei	7 KB

- Danach wird die dabei erstellte composer.json Datei im Verzeichnis manipuliert.
  Und zwar ändern wir alle Vorkommen von Drupal Core 8.8.0 (was Composers bevorzugte Version wäre) in 8.6.13.
  Die entsprechenden Zeilen sehen so aus:
  "drupal/core": "^8.8.0",

  "drupal/core-dev": "^8.8.0"
  und soll hinterher so aussehen:
  "drupal/core": "8.6.13",
- "drupal/core-dev"": "8.6.13"

  Seht Ihr den Unterschied? Das Hütchen muss unbedingt weg.
  Andernfalls bekommt Composer den Auftrag 8.6.13 ODER eine neuere Version zu installieren.
  Dann würde trotz der Manipulation die Version 8.9.3 installiert werden, was wir aber nicht wollen.

  Vorher:

montviso

Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach

Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980

```
    ],
    "require": {
        "php": ">=7.0.8",
        "composer/installers": "^1.2",
        "cweagans/composer-patches": "^1.6.5",
        "drupal/console": "^1.0.2",
        "drupal/core": "^8.8.0",
        "drupal/core-composer-scaffold": "^8.8.0",
        "drush/drush": "^9.7.1 | ^10.0.0",
        "vlucas/phpdotenv": "^4.0",
        "webflo/drupal-finder": "^1.0.0",
        "zaporylie/composer-drupal-optimizations": "^1.0"
    },
    "require-dev": {
        "drupal/core-dev": "^8.8.0"
    },
```

Nachher

```
    ],
    "require": {
        "php": ">=7.0.8",
        "composer/installers": "^1.2",
        "cweagans/composer-patches": "^1.6.5",
        "drupal/console": "^1.0.2",
        "drupal/core": "8.6.13",
        "drush/drush": "^9.7.1 | ^10.0.0",
        "vlucas/phpdotenv": "^4.0",
        "webflo/drupal-finder": "^1.0.0",
        "zaporylie/composer-drupal-optimizations": "^1.0"
    },
    "require-dev": {
        "drupal/core-dev": "8.6.13"
    },
    "conflict": {
        "drupal/drupal": "*"
    },
```

- Die folgende Zeile muss gelöscht werden, falls vorhanden.
  `drupal/core-composer-scaffold": "8.6.13"`

- Nach der Manipulation der composer.json Datei kann man den Composer nun beauftragen, die Installation der Version 8.6.13 durchzuführen.
  `composer install`

  Nicht verzweifeln, das dauert. ;-)

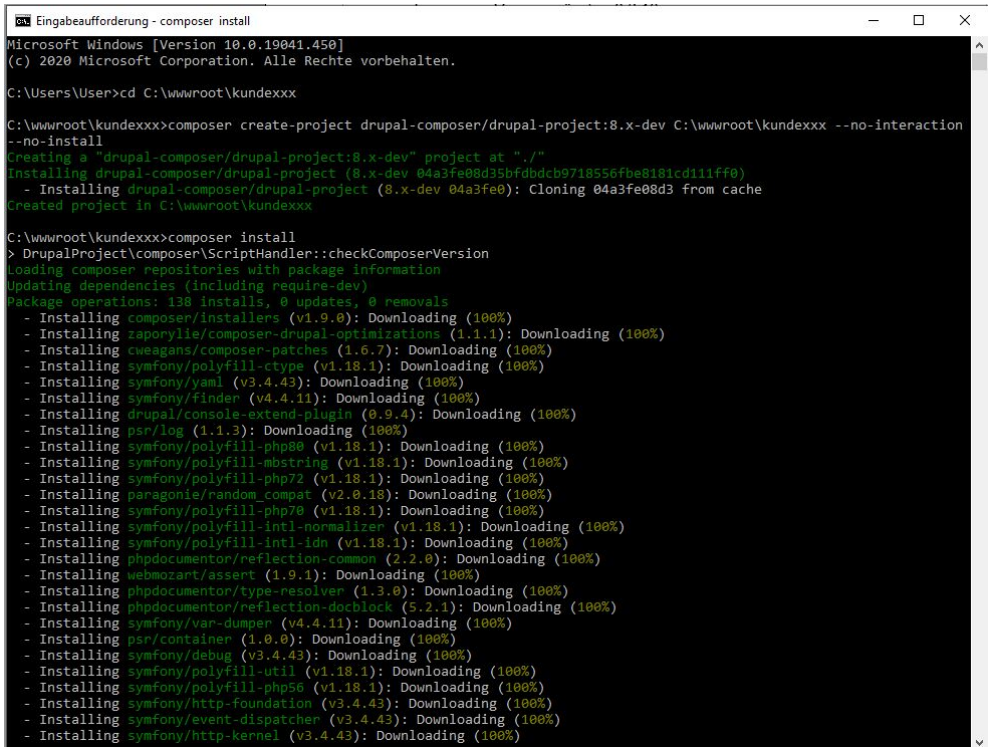
Hier Ausschnitte, wie das auf der Konsole aussieht:

montviso

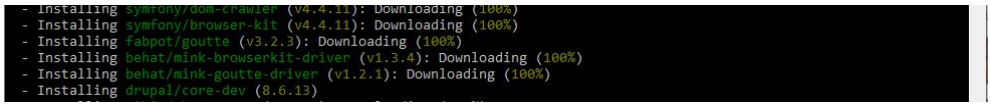
Montviso- Internetdienstleistungen

Regina Oswald
Stresemannstraße 15
DE-88400 Biberach

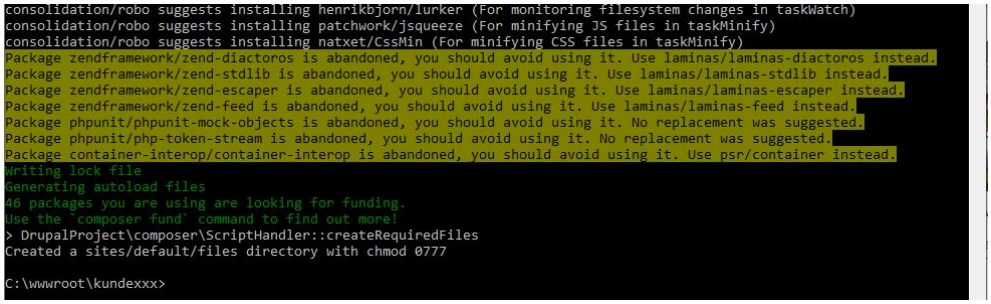
Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980



Entscheidend ist die Zeile mit Drupal Core und wir sehen, ja es wird die gewünschte Version installiert.



Kurz vor Ende gibt es gelb markierte Zeilen, die nun bei jedem Update angezeigt werden..
Es handelt sich um Fehlermeldungen bezüglich "abandoned Packages".
Nach Studium dieses Artikels bin ich zum Schluss gekommen, dass ich diese Hinweise ignorieren kann.
https://www.drupal.org/project/drupal/issues/3104015
Gegen Ende der Reise zu 8.9.3 werden die Hinweise immer weniger.



montviso
Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach
Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980
Impressum | Datenschutz

Wiederherstellen der individuellen Konfigurationen

Im Verzeichnis C:\wwwroot\kundexxx befindet sich nun die typische Verzeichnis-Struktur einer Drupal-Installation, die mit Composer - statt wie früher durch Download der Programmdateien - erzeugt wurde.

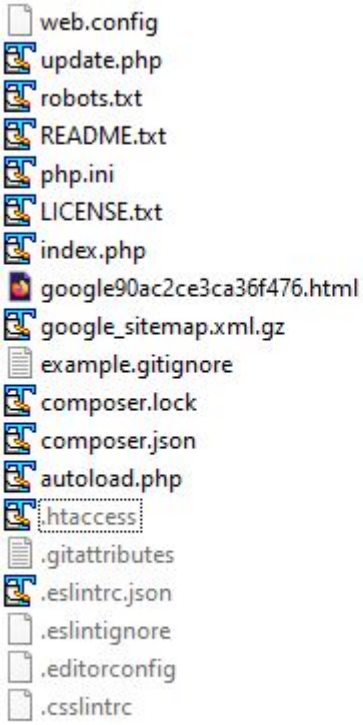
Um individuelle Dateien und Datenbank-Einträge unter der neuen Verzeichnis-Struktur wieder herzustellen, führen wir Folgendes durch

- Aus dem Ordner "sites" des Backups werden die Dateien und der Ordner "default" in den neuen Sites-Ordner kopiert.
  In der Datei "settings.php" müssen die Datenbank-Verbindungen vom Produktivsystem auf das lokale System geändert werden.
  Außerdem muss trusted host auf "kundexxx\.localhost" geändert werden.
  Auch die Pfade auf "protected"-Verzeichnis und "config_sync_directory" müssen evt. angepasst werden.
- Den Ordner "libraries" vom Backup kopieren wir direkt in "C:\wwwroot\kundexxx\web".
- Ebenso kopieren wir die Ordner aus "themes" und "modules" in die bereits angelegten Ordner gleichen Namens unter "C:\wwwroot\kundexxx\web".
- In der neuen Verzeichnisstruktur gibt es keine Dateien unter "C:\wwwroot\kundexxx\web", sondern nur Ordner.
  Deshalb kopieren wir die alten Dateien aus dem Root-Verzeichnis hier her.
  Das betrifft die folgenden Dateien:

montviso

Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach

Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980



- In der Datei ".htaccess" müssen wir die Weiterleitungen auf "https" und "www" auskommentieren, da vermutlich lokal nicht mit SSL zugegriffen wird und eine Weiterleitung auf "www" nicht gewünscht ist.
    - In der Datei autoload.php steht eine Zeile:
      `return require __DIR__ . '/vendor/autoload.php';`
      Da das vendor Verzeichnis nun nicht mehr ein Ordner auf gleicher Ebene, sondern auf übergeordneter Ebene ist, muss diese Zeile so umgeschrieben werden:
      `return require __DIR__ . '/../vendor/autoload.php';`

Module installieren, die Composer-Kontrolle benötigen

Die folgenden Module löschen wir aus dem Modul-Ordner und installieren sie mit Composer, damit die entsprechenden Libraries aus den Repositories geholt werden. Das ist nötig, weil ja unsere neue "composer.json" über der Root-Ebene nichts davon weiß, dass wir diese Module benötigen.

Wir installieren die gleichen Programm-Versionen, die bereits installiert waren und folglich auch in der Datenbank aktiviert sind:

- `composer require "dompdf/dompdf:0.7.0-beta3"`

montviso
Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach
Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980
Impressum | Datenschutz

- `composer require "drupal/entity_print:1.3"`
- `composer require "drupal/address:1.6"`

Inhomogene Struktur im Modules-Ordner reparieren

Ganz am Anfang habe ich erwähnt, dass mehrere Drupal-Module direkt im "modules"-Ordner, statt in "modules/contrib" liegen, weil sie über das Drupal Backend installiert wurden, statt händisch oder mit Composer.
Das hat zwar bislang nicht gestört, soll aber nun vereinheitlicht werden.

Wenn man die Modul-Ordner einfach in den Unterordner C:\wwwroot\kundexxx\web\modules\contrib schiebt und die Seite dann aufruft, kommt es zu einem White Screen und in den Logs gibt es eine entsprechende Fehlermeldung.

Wir müssen den Cache erst leeren und zwar sehr gründlich.
Über das Backend von Drupal ist das ja wegen der Fehlermeldung nicht möglich und erfahrungsgemäß reicht es nicht, die cache_* Tabellen in der Datenbank zu leeren.

Cache leeren

Wir leeren den Cache per Drush oder mit einem Aufruf von "http://kundenxxx.local/update.php", obwohl wir wissen, dass keine Datenbank-Aktualisierungen fällig sind.
Der Aufruf der update.php löscht nach meiner Erfahrung zuverlässig alle Caches.

Installation testen

Nun kommt der große Augenblick, wo wir unsere Installation mit neuer Verzeichnis-Struktur testen können.

Wir rufen am Browser auf: http://kundexxx.localhost

Normalerweise müsste die Seite sofort angezeigt werden, wenn wir im Vorfeld alle Fehler umschifft haben.

Da jede Installation individuell ist, würde es mich nicht wundern, wenn es bei Euch noch Hürden gibt und entweder sofort ein White Screen gezeigt wird, oder spätestens in den Protokolldateien von Drupal Fehler angezeigt werden.

montviso

Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach

Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980

Im ersten Fall sehen wir im Apache Log-File nach, das bei mir unter Windows 10 für dieses Projekt so aufgerufen wird (weil ich das in der Config des virtuellen Verzeichnisses so gewünscht habe):

C:\Program Files\Apache24\logs\kundenxxx-error.log

Mit den Fehlermeldungen, die hier gefunden werden, muss man nun im Web auf Suche gehen und auch etwas Phantasie aufbringen.

Wenn die Seite korrekt aufgerufen werden kann und die Homepage im Frontend angezeigt wird, dann loggen wir uns als Administratoren ein und besuchen als erstes den Status-Bericht. Hier sehen wir evt. Fehlermeldungen:

http://kundexxx.localhost/admin/reports/status

Ich bin schon happy, wenn nur eine Fehlermeldung erscheint:
`"PHP OPcode caching Nicht aktiviert "`

Da dieser PHP Opcode Cache momentan nicht gewünscht und nicht konfiguriert ist, ist die Fehlermeldung ja eher ein Hinweis, als ein Problem.

Danach besuchen wir nach und nach die wichtigen Seiten unserer Homepage und beobachten danach, ob im Fehler-Protokoll (watchdog) Warnings oder gar Fatal Errors erscheinen: http://kundexxx.localhost/admin/reports/dblog

Da muss dann im Einzelfall wieder überlegt bzw. im Netz nach der Fehlermeldung gesucht werden. Die Abhilfe ist individuell. In meinem Fall gab es an der Stelle keine Fehlermeldungen und die Webseite ist reif für die nächsten Update-Schritte.

Backup der bisherigen Version

Da es wirklich viel Arbeit war, die Installation soweit sauber zu bekommen, dass wir künftig einfach per Composer auf der Konsole updaten können, machen wir an dieser Stelle ein Update von der Datenbank und allen Dateien im Ordner C:\wwwroot\kundexxx\.

Es ist schon fast zu erwarten, dass bei den kommenden Updates etwas schief geht. Deshalb macht es Sinn, wenn wir nur bis zu diesem Schritt zurück gehen müssen.

montviso

Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach

Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980

Schrittweises Update bis zur aktuellen Version 8.9.3

Ich gehöre ja nicht zur geduldigen Sorte Mensch.
Deshalb habe ich früher gerne mal versucht, Update-Schritte auszulassen und nur die wichtigen "Minor Versions" einzuspielen.
Mit dem Hintergedanken, dass ja einmal gemachte Patches auch im Code bleiben.
Das ist aber ganz schnell an die Wand gefahren, weil Datenbank-Updates fehlen, die aufeinander aufbauen und letztendlich habe ich wirklich Schritt für Schritt jede Version sauber mit Composer aktualisiert und danach - sofern nötig - die Datenbank-Updates durch geführt bzw. Dateien wie .htacess, settings.php etc. unter Beibehaltung von individuellen Änderungen aktualisiert.

Bevor man das Update mit Composer durchführt, empfiehlt es sich, die Änderungen für dieses Release genau zu studieren, aus denen z.B. auch hervorgeht, ob Änderungen an den Seiten .htaccess, web.config, robots.txt oder settings.php gemacht werden und ob ein Datenbank-Update fällig ist.

Wir gehen dazu auf die Seite für unsere aktuell verbaute Version:
https://www.drupal.org/project/drupal/releases/8.6.13

Hier können wir uns rechts einen Überblick über die anstehenden Updates verschaffen und jeweils darauf klicken, um zu erfahren, um welche der folgenden Typen es sich handelt:

- Patch release
- Maintenance and security release
- Minor version

Hier kann man auch nachsehen, welche Änderungen am Programmcode oder in der Datenbank gemacht wurden.
Das kann ein wichtiger Hinweis sein, wenn es bei Composer oder Datenbank-Update zu Fehlermeldungen kommt.

Die gute Nachricht: Wenn man die Updates wirklich sauber und strukturiert Schritt für Schritt durchführt und vorher die Release-Notes liest, dann gibt es so gut wie keine Probleme.

montviso
Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach
Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980
Impressum | Datenschutz

Wir gehen also wieder auf die Konsole, wo vermutlich noch angezeigt wird, dass wir uns im Verzeichnis "C:\wwwroot\kundexxx\" befinden, wo die Composer-Befehle ausgeführt werden.
Evt. Datenbank-Updates mit drush werden dagegen im Unterordner "C:\wwwroot\kundexxx\web" ausgeführt.

Wartungsmodus

Auch wenn Du alleine an der Installation arbeitest und diese nicht öffentlich zugänglich ist, macht es Sinn, unter

http://kundenxxx.localhost/admin/config/development/maintenance

den Wartungsmodus einzuschalten.

Zusammenfassung des Vorgangs beim Update:

Hier eine kurze Raodmap, was uns erwartet:

- Drupal 8.6.14-8.6.18
- Drupal 8.7.0 – 8.7.9

  Installation von pathauto: 1:8

- Drupal 8.7.9 – 8.7.14

  Update aller Module auf den neuesten Stand

  Backup Datenbank und Verzeichnis

- Drupal 8.8.0- 8.9.3

  Ausführliche Tests und Debugging

  Backup der aktuellen sauberen Version und Live-Stellung

Status für die nächste Version checken

Wir können uns vor jedem Composer-Update anzeigen lassen, welche Abhängigkeiten aktualisiert werden können:

```
composer outdated drupal/*
```


montviso

Montviso- Internetdienstleistungen

Regina Oswald
Stresemannstraße 15
DE-88400 Biberach

Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980

Composer zeigt aber nur die Module an, die auch mit Composer installiert wurden. Ansonsten muss man sich per Hand über evt. zur Verfügung stehende Updates informieren.

Und folgender Befehl zeigt an, warum ein Update nicht funktionieren könnte:
`composer prohibits drupal/core:8.6.14`

Vorgehen bei den einzelnen Update-Schritten:

1. Composer auf der Konsole ausführen

composer require drupal/core:8.x.x –update-with-dependencies

(Ersetze x durch die gewünschte Versionsnummer)

2. Datenbank-Aktualisierung mit Drush oder update.php

Ich führe sie nach jedem Composer-Update durch, auch wenn in der Release-Beschreibung nicht ausdrücklich steht, dass es notwendig ist.
http://kundexxx.localhost/update.php
Damit ist dann auch gleich der Cache geleert.

3. Evt Änderungen an Dateien wurden gemacht:

Das betrifft .htaccess, web.config, robots.txt oder settings.php.
Von diesen Dateien müssen wir die neuen Versionen holen und unter Beibehaltung individueller Anpassungen über die alten Versionen kopieren.

3. Aufruf der Seite

http://kundexxx.localhost/admin/reports/status und auf evt. Fehlermeldungen untersuchen.

4. Aufruf des Fehlerberichtes auf gravierende Fehler:

http://kundexxx.localhost/admin/reports/dblog

5. Apache Error Log kontrollieren

Dieses habe ich immer offen im Editor mit laufen, der meldet, wenn sich Änderungen ergeben haben: C:\Program Files\Apache24\logs\kundenxxx-error.log

6. Aktualisierungs-Status der Module checken

http://kundexxx.localhost/admin/reports/updates
Hier kann man kontrollieren, ob die installierten Module Probleme mit der installierten Drupal Version verursachen. Idealerweise hat man aber schon vorher bei jedem einzelnen Modul geprüft, bis zu welcher Version es kompatibel ist.

montviso
Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach
Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980

Ob Du wirklich jeden der Kontroll-Möglichkeiten nach jedem Update-Schritt durch führst, hängt auch etwas von der Art des Releases, der eigenen Erfahrung und Deiner Risiko-Freude ab.

Spezielles Verfahren wegen pathauto ab Drupal 8.7.9

Sobald wir uns zur Version Drupal 8.7.9 vorgearbeitet haben, muss eine neue Version des Modules pasth_auto zwingend vor dem nächsten Update installiert werden.
Wenn man versucht unter 8.7.9 das Update von pathauto:1.8 mit Composer durchzuführen, kommt der Hinweis, dass mindestens Drupal 8.8.0 erforderlich ist. Also muss man im Dateiverzeichnis im Ordner web/modules/contrib den Ordner pathauto löschen und händisch durch die neue Version ersetzen und dann sofort das composer Update für 8.8.0 durchführen.

Update der Module vor Installation von Drupal 8.8.0

Sehr viele Module liegen für Drupal 8.8.0 in neuer Version vor und sollten mit Composer upgedatet werden:

```
composer update drupal/modulename –with-dependencies
```

In meinem Fall bleiben diese Module alle aktuell bis zur Version 8.9.3. Das mag aber im Einzelfall anders sein und je nach Modulen, die zum Einsatz kommen, kann es auch bei vorherigen Versionen notwendig sein, Module vor dem nächsten Core-Update auf neuen Stand zu bringen. Deshalb bitte für jedes Modul prüfen, bis bzw. ab welcher Drupal Version es kompatibel ist. Diese Informationen findet sich auf der Release-Seite des betreffenden Moduls.

Umzug auf das Produktivsystem und finale Tests

Nach Abschluss der Updates empfehlen sich weitere Tests, wie z.B. der Formulare oder individueller Programmierungen.
Auch Details wie die Tokens für og:image von Metatags können nicht genau genug überprüft werden, weil manchmal Änderungen fällig sind.

Die Tests sollten natürlich nicht nur auf der Entwicklungs-Umgebung durchgeführt werden, sondern auch auf dem zukünftigen Produktivsystem beim Hoster.

montviso

Montviso- Internetdienstleistungen
Regina Oswald
Stresemannstraße 15
DE-88400 Biberach

Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980

Zuerst sollte man den Domainnamen auf den Unterordner /web mappen, um sich den Schrecken eines White Screens zu ersparen.

Und nicht vergessen, auch auf der Produktiv-Umgebung auf PHP 7.3 umzustellen, falls es vorher nicht schon passiert ist.

Außer dem Einspielen der neuen Datenbank, braucht es auch noch die Anpassungen bezüglch SSL in der htaccess, sowie trusted Host und Pfade in der settings.php.

Individuelle Dateien wir google-Authentifzierung, sitemap.xml oder php.ini mit individuellen PHP-Konfigurationen müssen ebenfalls wieder eingespielt werden.

Ausblick

Die Composer-Struktur bietet weitere Vorteile, die man vorher nicht hatte.
Z.B. kann man in der Datei .env.example in .env umbenamen und nutzen, um Datenbank-Zugriff in der sites.php außerhalb der Root speichern, was natürlich deutlich sicherer ist.

Das kann man auch so vornehmen, dass künftig mit einem Switch zwischen Produktiv-Umgebung und Entwicklungs-Umgebung gewechselt werden kann. Schließlich war das ja nicht unser letztes Update mit Transfer zur Entwicklungsumgebung. ;-)

montviso

Montviso- Internetdienstleistungen

Regina Oswald
Stresemannstraße 15
DE-88400 Biberach

Mail: oswald@montviso.de
Tel.: +49 (0)7351 8295287
Fax: +49 (0)7351 3549980