

Drupal Upgrade 8.9.7 > 9.0.0 - Mit composer create-project

Bekanntlich ist end-of-life von Drupal 8 im November 2021.

Die meisten Kundenprojekte und auch unsere eigenen Projekte haben Module, die noch nicht zu Drupal 9 kompatibel sind.

Um frühzeitig Erfahrung zu sammeln, habe ich jetzt ein eigenes Projekt auf Drupal 9.0.0 aktualisiert.

Ich habe ein neues Projekt unter Drupal 9 angelegt und die individuellen Änderungen und Dateien aus sites-, module- und theme-Ordner auf die neue Version übertragen und die alte Datenbank aktualisiert.

Der Prozess gliedert sich in folgende Teile:

Mit dem Link kommst Du zu einer näheren Beschreibung:

1. Erfülle die Systemvoraussetzungen für Drupal 9
2. Update der Installation "Drupal 8" auf die letzte stabile Drupal 8 Version mit Composer
3. Update der Module auf den neuesten Stand mit Composer
4. Prüfen, ob alle Module kompatibel sind für Drupal 9
5. Deinstallieren aller Module, die nicht kompatibel (zu machen) sind
6. Prüfe, ob Themes kompatibel sind und mache evt. notwendige Anpassungen
7. Prüfen ob Twig Templates kompatibel sind
8. Erzeugen eine neue Installation "Drupal 9" mit Composer
9. Kopiere den Inhalt diverser Ordner in die neue Installation
10. Installiere die gewünschten Module mit Composer in aktueller Version
11. Lasse update.php oder drush Update laufen und debugge Fehlermeldungen einzeln
12. Ausführliche Tests

Standard-Prozess

Der Vollständigkeit halber hier eine kurze Zusammenfassung zum Standard-Prozess des Upgrades.

Siehe dazu auch die Dokumentation auf [drupal.org](https://www.drupal.org) bzw. unten genannte Links.

1. Erfülle die Systemvoraussetzungen für Drupal 9
2. Update der Installation auf die letzte stabile Drupal 8 Version mit Composer
3. Update der Module auf den neuesten Stand mit Composer
4. Prüfen, ob alle Module kompatibel sind für Drupal 9
5. Deinstallieren aller Module, die nicht kompatibel sind
6. Prüfe, ob Theme kompatibel ist und mache evt. notwendige Anpassungen
7. Prüfen ob Twig Templates kompatibel sind
8. checke Kompatibilität mit dem Modul
https://www.drupal.org/project/upgrade_status
9. Gebe Schreibrecht auf `web/sites/default/settings.php` und `web/sites/default/services.yml`
10. Ausführen des Composer Befehls zum Upgrade auf Drupal 9 mit `core-recommended` und `dev-dependencies` packages als Abhängigkeiten mit Parameter `-no-update`
11. prüfe ob `drupal/core-dev` installiert ist. Falls ja, verwende: `composer require drupal/core-dev:^9.0.0 --dev --update-with-dependencies --no-update`
12. führe das eigentliche Update aus: `composer update`
13. Dateien `htaccess`, `settings.php` etc. händisch aktualisieren.
14. Update der Datenbank
15. Stelle Berechtigungen zurück
16. Testen

Bei mir hat die Installation des Moduls `upgrade-status` nicht funktioniert oder ich habe einen wesentlichen Punkt nicht verstanden.

Es ging auch ohne, aber sicher wäre es eine Hilfe.

Beim Composer Update auf der Version Drupal 8.9.7 kam es zu Fehlermeldungen, die ich nicht einordnen konnte.

Ich bin dann einen alternativen Weg gegangen, den ich hier näher beschreibe:

Alternativer Weg:

Erfülle die Systemvoraussetzungen für Drupal 9

Die Systemvoraussetzungen findest Du hier:

<https://www.drupal.org/docs/understanding-drupal/how-drupal-9-was-made-and-what-is-included/environment-requirements-of>

Kurz zusammengefasst sind das die Folgenden:

- Apache 2.4.7
- PHP 7.3.
- MySQL Version 5.7.8+>
- Drush 10

Update der Installation "Drupal 8" auf die letzte stabile Drupal 8 Version mit Composer

Wenn nicht schon passiert, dann sollte die letzte stabile Version von Drupal 8 unter Composer Kontrolle sein.

Dazu habe ich hier einen Blog-Artikel geschrieben:

<https://www.montviso.de/blog/drupal-update-von-8613-zu-893-unter-composer-kontrolle>

Update der Module auf den neuesten Stand mit Composer

Bringe alle Module auf einen neuen Stand und prüfe bei der Gelegenheit, ob sie mit D9 kompatibel sind.

Prüfen, ob alle Module kompatibel sind für Drupal 9

Dazu kannst Du dieses Modul verwenden:

https://www.drupal.org/project/upgrade_status

Bei mir hat das nicht geklappt, aber das bedeutet nicht, dass ich es nicht beim nächsten Projekt wieder probieren würde.

Ich habe einfach nicht gefunden, wie man den Report für Module / Themes erzeugt, der nicht kompatible Stellen anzeigen soll.

Diese könnte man mit diesem Editor anpassen:

<https://www.drupal.org/docs/updating-drupal/how-to-prepare-your-drupal-7-or-8-site-for-drupal-9/deprecation-checking-and>

Man muss also nicht vorschnell verzweifeln, wenn ein Modul nicht als kompatibel angezeigt wird, oft fehlt nur die Zeile

```
core_version_requirement: ^8 || ^9
```

in der Datei meinmodul.info.yml.

Auf diese Art konnte ich z.B. Das Modul timepicker kompatibel bekommen.

Andere erfordern etwas andere Patches. Dazu unter Troubleshooting mehr.

Das gilt natürlich insbesondere auch für eigene Module, die auf jeden Fall um die o.g. Zeile erweitert werden müssen.

Deinstallieren aller Module, die nicht kompatibel sind

Gehe hier sehr vorsichtig vor, um später keine bösen Überraschungen zu erleben. Ein nicht kompatibles Modul, dass man übersehen hat, kann die schöne neue D9 Installation so verheuen, dass man sich nicht einloggen und nicht ohne weiteres deinstallieren kann.

Prüfe, ob Theme kompatibel ist und mache evt. notwendige Anpassungen

Auch ein Custom Theme / Sub-Theme benötigen diese Zeile:

```
core_version_requirement: ^8 || ^9
```

Prüfen ob Twig Templates kompatibel sind:

Das war bei mir nicht nötig.

Weitere Infos gibt es hier: <https://www.drupal.org/docs/upgrading-drupal/how-to-prepare-your-drupal-7-or-8-site-for-drupal-9/preparing-for-use-of-twig>

Erzeugen einer neuen Installation "Drupal 9" mit Composer

Lege einen neuen Ordner an pfad/zu/deiner/d9installation und installiere Hier D9 mit diesem Befehl:

```
composer create-project drupal/recommended-project  
pfad/zu/deiner/d9installation "9.0.0"
```

Du benötigst keine neue Datenbank und musst die Installation nicht im Frontend weiter ausführen.

Kopiere die Inhalt der Ordner von der alten in die neue Installation

- sites
- modules/custom
- themes/custom

Die Datei settings.php angepasst werden. Dazu von Drupal 9.0.0 die Original-Datei default.settings.php holen und mit Winmerge oder ähnlichem die Unterschiede abgleichen und die Datei mit den individuellen Anpassungen auf den neuen Stand bringen.

Ebenso mit der .htaccess verfahren, falls hier Anpassungen gemacht wurden.

Installiere die gewünschten Module mit Composer in aktueller Version

Die Module aus der alten Version, die kompatibel sind mit Drupal 9 werden nun in der jeweils aktuellen Version installiert. Das geht auch in einem Befehl mit Leerzeichen zwischen den Modulen. Composer arbeitet die ganze Liste durch, meldet evt. Probleme mit einem Modul und macht evt. Änderungen in der composer.json rückgängig, falls es nicht klappt, weil wir z.B. Ein Modul übersehen haben, dass nicht kompatibel ist.

```
composer require drupal/address drupal/admin_toolbar
drupal/adminimal_theme drupal/anchor_link drupal/block_class
drupal/bootstrap drupal/bootstrap_barrio drupal/colorbox drupal/crop
drupal/ctools drupal/devel drupal/dropzonejs
drupal/editor_advanced_link drupal/embed drupal/empty_fields
drupal/entity_reference_revisions drupal/entity drupal/entity_browser
drupal/entity_browser_enhanced drupal/entity_embed drupal/entity_print
```

u.s.w

Die Module sind automatisch enabled, weil wir ja mit der alten Datenbank arbeiten.

Diese Liste an Modulen muss natürlich identisch sein mit den zuletzt installierten Modulen unter Drupal 8.9.7. Andernfalls gibt es eine Fehlermeldung Missing Module.

Lasse update.php laufen und kläre Fehlermeldungen einzeln!

Nun kannst Du entweder mit Drush oder mit update.php ein Datenbank Update auf Core und die Module ausführen.

Idealerweise läuft es einfach durch, aber vermutlich eher nicht.

Es folgen ein paar Beispiele an Fehlermeldungen und was bei mir geholfen hat zur Beseitigung.

Das ist natürlich keine objektive Aufstellung. Schließlich ist jede Drupal-Installation ein kleiner Kosmos für sich und es gibt selten zwei identische Installationen.

Dazu ist die Kombinationsmöglichkeit an Modulen, Betriebssystem, Versionen, PHP Versionen, Themes, eigenen Erweiterungen etc. viel zu groß.

Ich möchte Mut machen, dass hässliche Fehlermeldungen häufig einfache Lösungen haben.

Es rentiert sich also, im Netz zu suchen, bzw. wenn das verursachende Modul bekannt ist, auch direkt in den Issues.

Troubleshooting

1. Key 'core_version_requirement'

Drupal\Core\Extension\InfoParserException: The 'core_version_requirement' constraint (^8.8.0 || ^9.0) requires the 'core' key not be set in modules/contrib/menu_block/tests/modules/menu_block_test/menu_block_test.info.yml in Drupal\Core\Extension\InfoParserDynamic->parse() (line 88 of core/lib/Drupal\Core\Extension\InfoParserDynamic.php).

In dem Fall betrifft es ein Modul, dass bereits als kompatibel gekennzeichnet war, nämlich das Modul menu_block. Dort fehlt allerdings im Sub-Modul menu_block_test der Key 'core_version_requirement'

D.h. Im Klartext, man muss in die Datei

modules/contrib/menu_block/tests/modules/menu_block_test/menu_block_test.info

gehen und dort die Zeile

core: 8.x

ersetzen durch

core_version_requirement: ^8.8.0 || ^9.0

2. Modul fehlt

Drupal\Component\Plugin\Exception\PluginNotFoundException: Unable to determine class for field type 'address' found in the 'field.storage.node.field_adresse' configuration in Drupal\field\FieldStorageConfigStorage->mapFromStorageRecords() (line 167 of core\modules\field\src\FieldStorageConfigStorage.php)

Lösung: Installiere das Modul mit Composer.

```
composer require drupal/address
```

3. Syn directory anlegen

Configuration sync directory

Nicht vorhanden Your /settings.php file must define the

>\$settings['config_sync_directory'] setting as a string containing the directory in which configuration files can be found.

Lege ein Verzeichnis im Files-Ordner mit Unterordner sync und verweise setze in der settings.php die Variable

```
$settings['config_sync_directory']
```

4. Inkompatibles theme

The following theme is installed, but it is incompatible with Drupal 9.0.0:

Öffne die deintheme.info.yml und ersetze dort die Zeile

```
core: 8.x
```

durch

```
core_version_requirement: ^8.8.0 || ^9.0
```

5. Inkompatibler Code in einem der Module

Möglicherweise kann man sich in das Backend einloggen und sieht keinen Fehler, aber an irgendeiner Stelle erscheint im Frontend eine Fehlermeldung wie diese:

Symfony\Component\DependencyInjection\Exception\ServiceNotFoundException: You have requested a non-existent service "path.alias_manager"

Da hilft nur in Google nach genauem Wortlaut suchen, bzw. Teile der Fehlermeldung in Anführungszeichen setzen.

Da erscheint z.B. ein Hinweis auf ein Modul, dass ich nicht installiert habe.

https://www.drupal.org/files/issues/2020-06-19/feeds_youtube-requested_...

Es ist naheliegend, dass die dort geänderte Syntax auch in einem der hier verwendeten Module zum Einsatz kommen muss.

Ich suche also mit meinem TextPad-Editor in allen Dateien des Unterordners modules nach dem String `path.alias_manager`.

Und siehe da, ich werde hier fündig:

```
contrib\onomasticon\src\Plugin\FILTER\FILTER\FilterOnomasticon.php(319):
```

Ich ersetze die Zeile

```
$aliasManager = \Drupal::service('path.alias_manager');
```

```
$aliasManager = \Drupal::service('path_alias.manager');
```

Und schon verschwindet die Fehlermeldung.

Ausführliche Tests

Ganz klar, dass nach Beseitigung der ganz offensichtlichen Fehler, die z.B. Den Update-Befehl verhindern oder im Frontend eine Fehlermeldung die Darstellung komplett verhindern, die Tests noch nicht zu Ende sind.

Ich rufe dann im Frontend und im Backend alle wichtigen Bereiche einmal auf, lege neue Inhalte an, erstelle Blöcke u.s.w. Und rufe nach jedem Schritt den Watchdog auf, um evt. Warnings oder Schlimmeres zu sehen.

Natürlich kann auch ein PHP-Error-Log oder das Error-Log des Apache-Webservers noch zusätzlichen Aufschluss geben.

Fazit

Unterm Strich bin ich sehr zufrieden mit dem Upgrade auf Drupal 9.

Verglichen mit Upgrades von Drupal 7 auf 8 ist es ein Kinderspiel, weil man ja schon etwas vertraut ist mit Composer.

Bei diesem Beispiel Projekt waren nur drei von 50 Modulen nicht kompatibel.

Die waren nicht unbedingt notwendig, was natürlich ein Glücksfall ist.

Noch wird ja Drupal 8 gepflegt und in einem Jahr sind sicher alle wesentlichen Module, die es in Drupal 8 gibt, auch umgestellt worden.

Hilfreiche Linksammlung:

<https://www.drupal.org/docs/upgrading-drupal/how-to-prepare-your-drupal-7-or-8-site-for-drupal-9/upgrading-a-drupal-8-site>

<https://drupalize.me/tutorial/upgrade-drupal-9?p=2766>

<https://lightning.acquia.com/blog/getting-ready-drupal-9>

<https://www.drupal.org/docs/understanding-drupal/how-drupal-9-was-made-and-what-is-included/environment-requirements-of>

<https://www.drupal.org/docs/updating-drupal/how-to-prepare-your-drupal-7-or-8-site-for-drupal-9/deprecation-checking-and>

<https://www.drupal.org/docs/upgrading-drupal/how-to-prepare-your-drupal-7-or-8-site-for-drupal-9/preparing-for-use-of-twig>

<https://www.drupal.org/docs/updating-drupal/troubleshooting-database-updates>