WordPress: Mit wp-cli in Datenbank suchen und ersetzen

Es gibt verschiedene Situationen, in denen es nötig ist, Daten in einer WordPress-Datenbank per Suchen und Ersetzen auszutauschen. Zu den gängigen Anforderungen gehören die beiden Folgenden:

- Umzug einer Seite, z.B. von einer Domain zu einer anderen, etwa, um eine Seite, die auf einem Liveserver liegt, in einer lokalen WordPress-Entwicklungsumgebung laufen zu lassen
- Umstellung von http:// auf https://
- Duplicator kommt mit Größe der Daten nicht klar

Ein Hinweis vorweg: auch wenn Du DevKinsta oder Local (und damit wp-cli [wp-cli.org]) auch unter Windows verwenden kannst, beschreibe ich hier das Vorgehen am Mac.

Ich zeige Dir anhand des sehr tollen neuen Tools DevKinsta, wie Du per suchen und ersetzen die ursprüngliche Domain in Deiner lokalen Entwicklungsumgebung korrigieren kannst. Ich simuliere hier eine Situation, vor der ich selber kürzlich stand: Ich wollte eine Kundenseite, eine WordPress-Multisite, vom Liveserver auf meine lokale Entwicklungsumgebung übertragen. Dabei gab es folgende zwei Probleme, die ich mit meinem bisherigen Workflow und Tools (Duplicator [de.wordpress.org/plugins/duplicator] und local [localwp.com]) nicht in den Griff bekam (das ansonsten geniale Gratis-WordPress-Plugin Duplicator stelle ich Dir in zwei anderen Blogbeiträgen genauer vor: geisler-design.de/video-tutorial-lokale-wordpress-seite-mit-plugin-duplicator-auf-webserver-umziehen bzw. geisler-design.de/wordpress-live-seite-lokal-bearbeiten, meinen Blogbeitrag zu local findest Du hier: geisler-design.de/wordpress-lokal-installieren-local-flywheel):

1. Duplicator verarbeitet in der Basisversion keine Multitsites
2. local arbeitet nicht mit Multisites

Dass ich Duplicator nicht einsetzen konnte, habe ich schnell mit einem einfachen Workaround gelöst. Die Datenbank per PHPMyAdmin aus dem Verwaltungsbereich des Webhosters exportiert, die gesamte Installation per FTP auf den eigenen Rechner geladen.

Aber für local brauchte ich dringend Ersatz. Hier die entsprechende Info von Mitte ’20, dass local nicht ohne Weiteres für den Einsatz von Multisites eingesetzt werden kann localwp.com/community/t/import-multisite/20545). Und habe in dem nagelneuen Tool DevKinsta eine Superlösung gefunden!

DevKinsta: die neue lokale Entwicklungsumgebung für WordPress-Seiten

Tipp an Dich: Schau Dir DevKinsta (kinsta.com/devkinsta) unbedingt an: großes Kino! Ich habe mich sehr schnell zurechtgefunden und finde das Tool sehr, sehr gelungen! Kleine Bemerkungen zu DevKinsta: im Vergleich zu local sind bei DevKinsta die Optionen etwa bei der Variante »Benutzerdefinierte Seite« nicht so umfangreich. Und bei DevKinsta lässt sich auch nicht die Art des Servers nicht wählen: es steht nur NGINX zur Verfügung und auch die Datenbankversion ist nicht wählbar. Alle drei Punkte kann ich aber verkraften ;-) Aber: Dass DevKinsta sehr gut mit Multisites zurecht, gab für mich den Ausschlag, mich mit DevKinsta zu beschäftigen.

Beispielszenario: URLs per suchen und ersetzen austauschen

Im Folgenden spiele ich die o.g. Situation durch: ich möchte eine WordPress-Seite vom Liveserver in meine lokale Entwicklungsumgebung überführen und kann duplicator nicht verwenden. Ich gehe in folgenden fünf Schritten vor. Dabei verwende ich eine Standard-WordPress-Installation, lasse hier also das Thema Mulitsite außen vor. Übrigens

Dipl. Des. Karsten Geisler
0221 – 977 1978
Schwerinstraße 40
50735 Köln

info@geisler-design.de
geisler-design.de

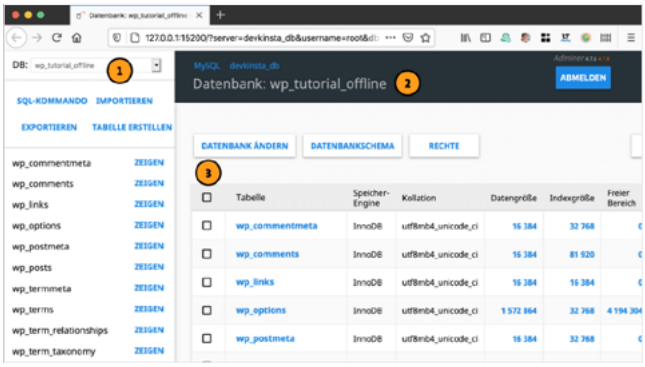
funktioniert dieses Vorgehen bei anderen Kombinationen prinzipiell genauso, also z.B., wenn Du die lokale Datenbank auf den Liveserver spielen möchtest. Voraussetzung beim folgenden Vorgehen ist, dass Du die Datenbank – etwa per PHPMyAdmin aus dem Adminbereich Deines Webhosters – schon heruntergeladen hast.

1. Die Inhalte der automatisch erzeugten Datenbank löschen
2. Die heruntergeladene Datenbank in die gerade geleerte Datenbank importieren
3. Die alte URL (der Liveseite) per wp-cli durch die neue URL (unter der die Seite des lokalen DevKinsta-Webservers erreichbar sein) austauschen

Das klingt doch übersichtlich :) Also, auf gehts!

1. Inhalte der automatisch erzeugten Datenbank löschen

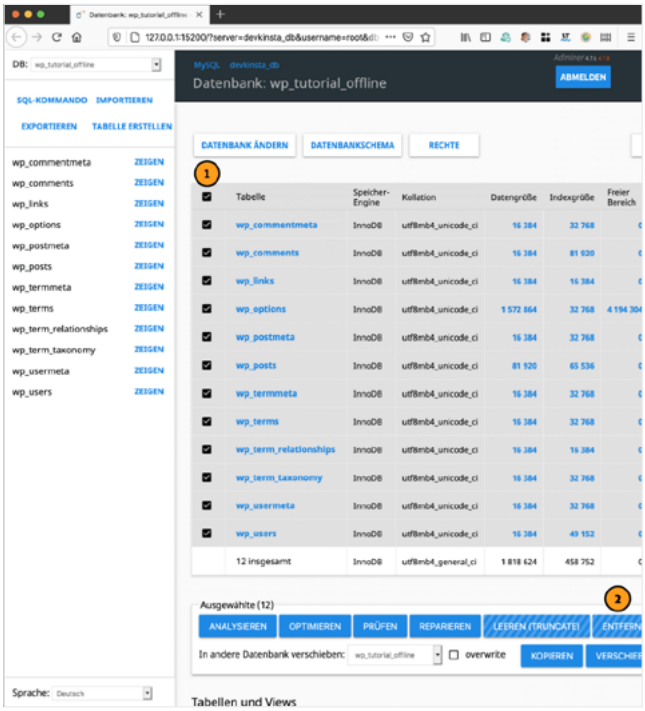
Um die Inhalte der bisherigen Datenbank meiner Entwicklungsumgebung zu löschen, rufe ich mir das entsprechende Tool auf. In DevKinsta ist das Adminer (falls Du mit PHPMyAdmin arbeitest: Du wirst alles ziemlich genau so wiedererkennen):



Unter (1) und (2) siehst Du, welche Datenbank bzw. welche Daten angezeigt werden. Im Dropdown-Menü (1) kannst Du ggf. auch eine andere Datenbank, die auf Deinem Datenbankserver läuft, aufrufen. Zum Löschen der Inhalte – nicht der Datenbank – klicke ich auf die Checkbox im Tabellenkopf neben Tabelle (3).

Dipl. Des. Karsten Geisler
0221 – 977 1978
Schwerinstraße 40
50735 Köln

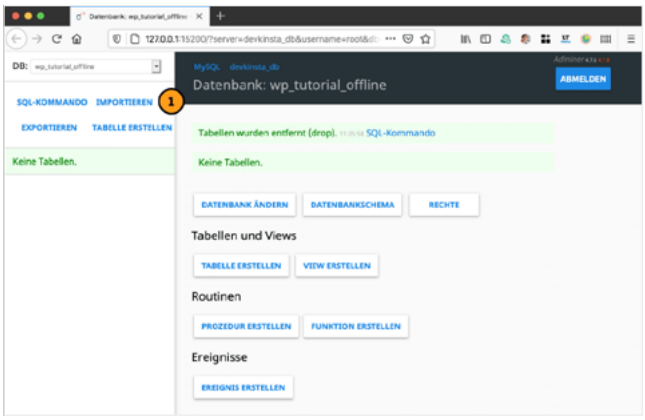
info@geisler-design.de
geisler-design.de



Hier sind alle Tabellen ausgewählt (1) und mit einem Klick auf »Entfernen« (2) werden diese – ähm – entfernt. Übrigens ist dieser Schritt nicht unbedingt notwendig: Ich hätte auch direkt die heruntergeladenen Datenbank importieren können. Dadurch würden die bisherigen Daten gelöscht. Mir ist es so lieber, weil ich sicher sein kann, dass so wirklich nichts an Altlasten überbleibt und es macht das Vorgehen hier sehr deutlich.

2. Die heruntergeladene Datenbank in die leere Datenbank importieren

Nach dem Klick auf »Entfernen« im vorigen Schritt, ändert sich die Darstellung in adminer:

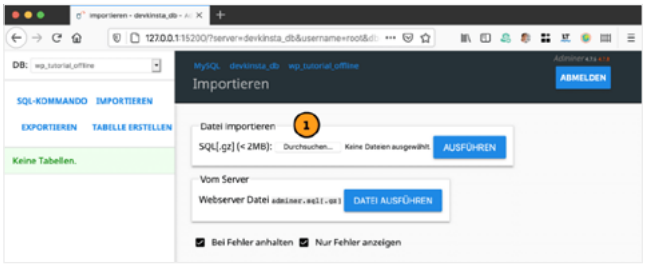


Um die Daten der heruntergeladenen Datenbank in die leere Datenbank zu importieren, klickst Du auf »Importieren« (1).

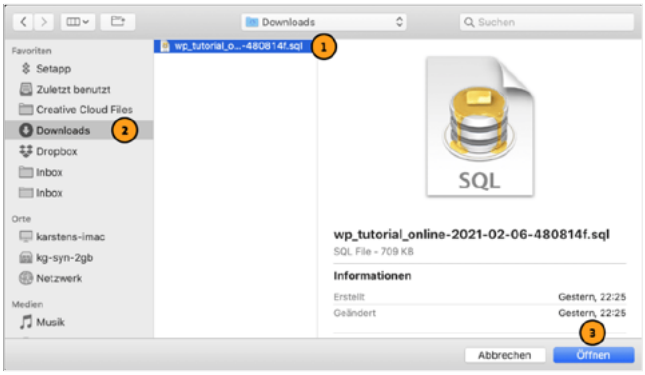
Dipl. Des. Karsten Geisler
0221 – 977 1978
Schwerinstraße 40
50735 Köln

info@geisler-design.de
geisler-design.de

Jetzt bietet Dir Adminer die Auswahl der entsprechenden Daten zum Importieren an.

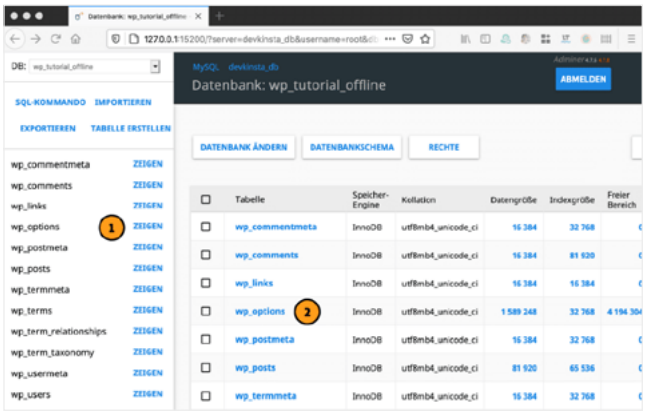


Durch Klick auf »Durchsuchen« (1) öffnet sich ein Dateibrowser:



Meine Datenbank (1) habe ich mir zuvor aus dem Adminbereich meines Webhosters per PHPMyAdmin heruntergeladen, daher liegt sie in meinem Download-Ordner (2). Durch Klick auf »Öffnen« (3) wirden die Daten direkt importiert. Je nach Größe der Datenbank kann das etwas dauern.

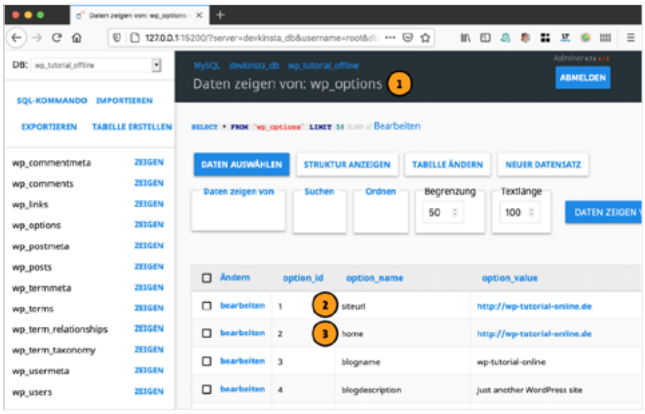
Tipp: Wenn Du schon Erfahrung mit wp-cli hast, importiere die Datenbank auch mit wp-cli. Das geht damit deutlich schneller als mit adminer oder PHPMyAdmin.



Dass ich jetzt die Daten aus der Online-Datenbank importiert habe, ist in dieser Ansicht nicht zu sehen. Das überprüfe ich jetzt, indem ich mir die Inhalte der Tabelle »wp_options« ansehe. Dafür klicke ich auf »Zeigen« (1) neben dem Tabellennamen in der linken Spalte oder ich klicke auf den Tabellennamen »wp_options« (2) in der rechten Übersicht.

Dipl. Des. Karsten Geisler
0221 – 977 1978
Schwerinstraße 40
50735 Köln

info@geisler-design.de
geisler-design.de



Oben zeigt mir Adminer an, welche Datenbanktabelle ich vor mir habe: »wp_options«. Und unten sehen ich, dass in den Tabellenzeilen, in denen die »siteurl« bzw. »home« gespeichert sind, die Daten der Onlineversion der Website gespeichert sind »http://wp-tutorial-online.de«.

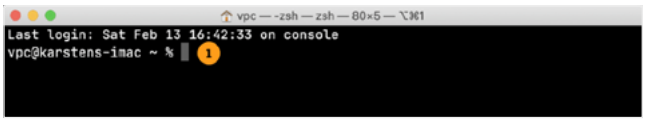
Nur zur Info: das sind genau die Daten, die Du im WordPress-Backend unter »Einstellungen › Allgemein« in »WordPress-Adresse (URL)« und »Website-Adresse (URL)« eingibst.

Jetzt hier einfach die korrekten Daten einzugeben, kann zwar funktionieren, ist aber keine gute Idee. Die URL müssen nämlich noch an zig anderen Stellen ausgetauscht werden. Etwa bei den Links in der Navigation und den Bildern. Du siehst gleich in wp-cli, in welchen Tabellen überall noch Änderungen vorgenommen werden müssen. Deshalb lasse ich hier die alten URLs stehen: Ich ersetze gleich alle URLs per wp-cli.

3. Mit wp-cli URLs suchen und ersetzen

O.k. – jetzt kommen wir zum eigentlichen Suchen und Ersetzen von URLs. Zunächst startest Du das Terminal – das von Apple auf allen Mac vorinstallierte Commandline-Tool. Terminal findest Du unter »Progamme › Dienstprogramme › Terminal«.

Nach dem Aufruf von Terminal siehst Du in etwa Folgendes vor Dir:



Kurzer Exkurs zum Terminal

Hier ein paar einführende Infos zu den Angaben, in der zweiten Zeile (1). Für das weitere Vorgehen ist das Verständnis keine Voraussetzung, aber sicher hilfreich, wenn Du gerade anfängst, mit dem Terminal zu arbeiten.

Die Infos, die Terminal immer beim Start jedes neuen Fensters ausgibt, sehen kryptisch aus, sind aber eigentlich gar nicht so schwer zu entziffern. In der ersten Zeile ist eigentlich selbsterklärend – und weiter nicht für uns von Interesse.

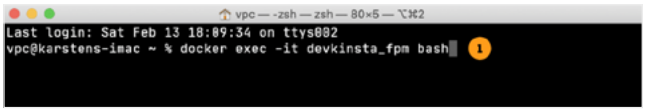
Bei mir steht in der zweiten Zeile `vpc@karstens-imac ~ %. vpc` ist mein interner Benutzer-Name am Mac und `karstens-imac` ist der interne Name meines Macs. Hier wird also einfach ausgegeben, welche*r Benutzer*in an welchem Mac angemeldet ist. (Hintergrund dieser Info: mit dem Terminal kannst Du Dich auch bei anderen Benutzeraccounts anmelden – und auch bei anderen Macs.) `~` ist das Zeichen für Benutzerordner. Mein Terminal weist also einfach in des Homeverzeichnis, das Du im Finder über »Gehe zu › Benutzerordner« aufrufen kannst. Dann kommt noch das `%`. Hier ist es einfach das Zeichen, dass hier der Eingabebereich für Befehle beginnt. Alles, was ich hinter dem `%` eingebe und mit Enter bestätige, wird das Terminal versuchen, als Befehl zu interpretieren und auszuführen. Hier siehst Du auch den Textcursor. O.k., soviel zu den Basics des Terminals.

Dipl. Des. Karsten Geisler
0221 – 977 1978
Schwerinstraße 40
50735 Köln

info@geisler-design.de
geisler-design.de

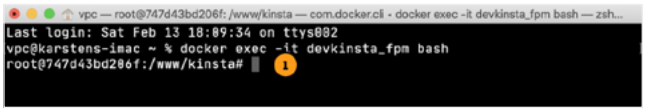
Um nun wp-cli zu benutzen, müssen wir erstmal sozusagen auf den Datenbankserver von DevKinsta kommen. Das ist zum Glück sehr einfach, weil uns die Website kinsta.com/knowledgebase/devkinsta/wp-cli genau den nötigen Code liefert:

Also: Code kopieren, am Textcursor im Terminal einsetzen:



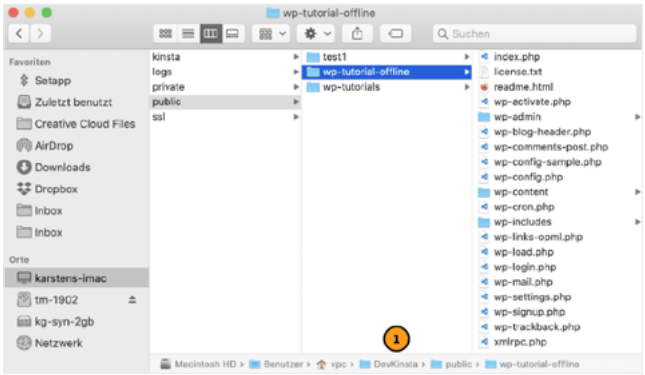
Anschließend mit Enter bestätigen.

Was hier genau passiert, interessiert uns nicht en detail, nur soviel: DevKinsta und auch Local arbeiten intern mit sogenannten Dockerimages. Das sind nach außen abgeschlossene Container, in denen z.B. ganze Betriebssysteme laufen können. DevKinsta und Local benutzen Dockercontainer, in denen Web- und Datenbank-Server laufen. Und im Befehl oben siehst Du zu Beginn den Befehl »docker«. Durch die Ausführung dieser Zeile gelangen wir also in den DevKinsta Webserver! Nach Eingabe des Befehls und Bestätigung mit Enter siehst Du nach ein paar Momenten eine neue Zeile im Terminal:



Yes! In der neuen Zeile siehst Du, dass sowohl der Benutzer also auch der Rechner andere sind als in der Zeile zwei: ich bin auf dem lokalen Webserver von DevKinsta! (Der Code für den Rechner (747…) wird von DevKinsta bei der Installation generiert und wird bei Dir vermutlich anders aussehen.) Das Verzeichnis auf dem Webserver wird hier mit `/www/kinsta` angegeben. Kleine Info am Rande: Auf dem Server wird die Befehlszeile, die bisher mit `%` eingeleitet wurde, jetzt mit `#` eingeleitet.

Jetzt geht es darum, noch in die richtige WordPress-Installation zu kommen. Dazu schaue ich mir an, wie das DevKinsta-Verzeichnis im Mac-Finder aussieht. Den Pfad beliebiger Verzeichnisse kannst Du Dir im Finder anzeigen lassen, indem Du im Menü auf »Darstellung › Pfadleiste« anwählst:



Das Verzeichnis, das im Terminal mit `/www/kinsta` angeben wird, entspricht auf dem Mac dem DevKinsta-Verzeichnis, das direkt in meinem Benutzerordner liegt (1). Um nun im Terminal in mein Verzeichnis »wp-tutorial-offline« hineinzukommen, gehe ich vom `/www/kinsta`-Verzeichnis ins »public-« und dann ins »wp-tutorial-offline«-Verzeichnis.

Mit »cd« das Verzeichnis wechseln

Im Terminal ist das sich von einem in ein anderes Verzeichnis-Bewegen eine ganz zentrale Aktion – und entspricht exakt dem Vorgehen im Finder. Der Unterschied besteht eigentlich nur darin, dass Du diese sich Bewegen im Terminal per Befehl machst. Und der lautet `cd` (für »change directory«):

Dipl. Des. Karsten Geisler
0221 – 977 1978
Schwerinstraße 40
50735 Köln

info@geisler-design.de
geisler-design.de

Hier habe ich einfach cd `public/wp-tutorial-offline/` ins Terminal eingeben. Nach Bestätigung mit Enter bin ich genau in der WordPress-Installation, in der ich per wp-cli die URLs tauschen möchte.

Hier (1) siehst Du, dass ich jetzt im richtigen Verzeichnis bin.

Tipp: Mit »ls« die Inhalte eines Verzeichnisses anzeigen

Ich kann mir ganz einfach die Inhalte des aktuellen Verzeichnisses anzeigen lassen. Dafür verwende ich `ls` (für »list«). Dieser Schritt ist vollkommen optional. Ich zeige ihn Dir hier, weil Du so schnell überprüfen kannst, ob Du da bist, wo Du hinwillst oder um Dich einfach umzuschauen:

In diesem Screenshot habe ich hinter der letzten Zeile einfach `ls` eingegeben und mit Enter bestätigt. Daraufhin werden mir alle (sichtbaren) Dateien und Unterverzeichnisse (hier in Violett) alphabetisch geordnet, angezeigt.

wp-cli mit wp aufrufen

Und ab hier kommt jetzt endlich wp-cli zum Zuge. Für den Austausch von Zeichenketten bringt wp-cli den mächtigen Befehl `search-replace` mit. Da dieser Befehl ein Unterbefehl von wp-cli ist, musst Du das dem Terminal mitteilen. Das machst Du über das Voranstellen von `wp`. Also sieht der Anfang von Suchen und Ersetzen mit wp-cli so aus: `wp search-replace`. Jetzt müssen wir natürlich noch angeben, was genau wir suchen und womit wir die gefundenen Textstellen ersetzen wollen. Das machen wir einfach mit zwei weiteren Angaben, die mit Leerzeichen getrennt hinter den bisherigen Code geschrieben werden:

In meinem Fall möchte ich `wp-tutorial-online.de` mit `wp-tutorial-offline.local` ersetzen. Also schreibe ich:

Wenn Du wie ich mit DevKinsta arbeitest, musst Du dem bisherigen Befehl nun noch eine Option mitgeben: `--allow-root`. Ohne diesen Zusatz erhältst Du von wp-cli eine Warnmeldung. (Wenn Du mit local statt mit DevKinsta arbeitest, brauchst Du diese Option nicht!)

Der bisherige komplette Befehl sieht jetzt so aus:

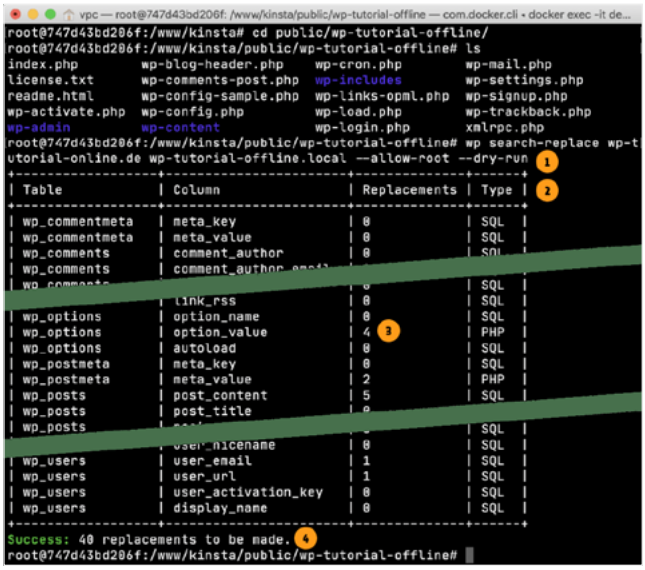
```
wp search-replace wp-tutorial-online.de wp-tutorial-offline.local
--allow-root
```

Um die Vorzüge von wp-cli kennenzulernen, empfehle ich Dir noch eine weitere Option: `--dry-run`. Damit checkt wp-cli erst einmal, wo überall Deine Suchphrase in der Datenbank vorkommt:

O.k.! Diese Zeile lasse ich im Terminal jetzt mal laufen und bekomme folgende Ausgabe:

Dipl. Des. Karsten Geisler
0221 – 977 1978
Schwerinstraße 40
50735 Köln

info@geisler-design.de
geisler-design.de



Bei (1) siehst Du den kompletten Code. Nach Drücken von Enter arbeitet wp-cli und gibt praktisch sofort eine Tabelle (2) aus, in der Du siehst, wo überall Deine Suchphrase vorkommt (3). Und ganz unten siehst Du noch die Angabe, wie viele Fundstellen es überhaupt gibt. In meinem Fall sind es 40!

Wenn soweit alles o.k. ist, rufe ich mir den letzten Befehl noch mal auf – dafür einfach die Pfeiltaste nach oben drücken. Dadurch ersparst Du dir viel Tipperei. Dann nur noch die Option `--dry-run` löschen und dann wp-cli noch mal laufen lassen. Danach habe ich die URLs in der kompletten Datenbank ausgetauscht – und kann mit der lokalen Entwicklung starten!

Well, das ist eine Menge Holz, oder? Dennoch: ich hoffe sehr, dass ich Dir mit dieser Anleitung wp-cli schmackhaft machen konnte! Bei Fragen freue ich mich über Deinen Kommentar und werde versuchen, Dir weiterzuhelfen!

Dipl. Des. Karsten Geisler
0221 – 977 1978
Schwerinstraße 40
50735 Köln

info@geisler-design.de
geisler-design.de