

WordPress MultiSite plus Multilingual

Inhaltsverzeichnis

1 Basic MultiSite-Install.....	1
2 Was ist nun wo?.....	2
3 Als Super Admin Theme und Plugins laden (Beispiele).....	3
4 Sprachvarianten-Websites.....	4
5 Sprachvarianten-Plugin.....	5
6 Automatische Sprachwahl für Besucher, Adminkennungen.....	7
7 Manuelle Sprachwahl.....	8
7.1 Flaggen.....	9
8 Sitemap.xml und Robots.txt.....	9
8.1 Sitemap.xml.....	9
8.2 Robots.txt.....	10
9 Nginx.....	10
10 Nützliche MultiSite-Tipps und -Plugins.....	13

1 Basic MultiSite-Install

https://codex.wordpress.org/Create_A_Network

https://codex.wordpress.org/Before_You_Create_A_Network

<https://www.elegantthemes.com/blog/resources/the-complete-guide-to-creating-a-wordpress-multisite-installation>

Manchen Unkenrufen zum Trotz, ist es meiner Erfahrung nach mit ein wenig „gewusst wie“ auch eine länger bestehende WP-Website nachträglich auf MultiSite umzustellen:

https://www.mittelstandswiki.de/wissen/WordPress_Multisite

iThemes Security und WP-Piwik verlieren dabei ihre Einstellungen. WP-Edit nicht.

- Falls Apache: Sich vergewissern dass mod_rewrite verfügbar ist:

<https://docs.bolt.cm/3.4/howto/making-sure-htaccess-works>
/home/tim/Zed/WordPress/Tools/htaccess_tester.php

- Note: Permalinks = Beitragsname nutzt auch mod_rewrite, daher meistens verfügbar.

- Ein nacktes WP installieren, Akismet & Dolly Plugins sowie überflüssige Themes löschen, Admin-Profil ergänzen, Standard-Einstellungen vornehmen, insb. Permalinks = Beitragsname.
- Etwa aktive Plugins (z.B. Coming soon) deaktivieren.
- Ergänze wp-config.php oberhalb der Zeile "Das war's, Schluss mit dem Bearbeiten!" (bzw. „That's all, stop editing!“):
/* Multisite */
define('WP_ALLOW_MULTISITE', true);
- Upload, Refresh und gehe zu: **Werkzeuge | Netzwerk-Einrichtung** (Tools | Network Setup)
- Unterverzeichnisse ist per Default gewählt, lassen.
- Falls stattdessen geblökt wird, etwa „Because your install is not new, the sites in your WordPress network must use sub-domains“, gibt es [hier](#) eine gute Lösung.
- Installieren
- Bearbeite wp-config und .htaccess wie befohlen.
Die Anweisungen für wp-config kommen unmittelbar nach 'WP_ALLOW_MULTISITE'.
Upload, Refresh => erneut Einloggen.
- Falls der Webserver nicht Apache sondern Nginx ist, merkt das WordPress leider nicht, und bietet stur die Anpassungen für .htaccess an. Siehe unten den Kapitel zu Nginx.
- Die Angebotene Anpassung für wp-config.php und ggf. für .htaccess einsetzen, hochladen. Erneut einloggen.
- MultiSite-Dialoge zur Netzwerkverwaltung sind aus dem OBEREN Toolbar erreichbar unter:
"Meine Websites | Netzwerkverwaltung" (My Sites | Network Admin). Ein Klick direkt auf dieses Menüpunkt wechselt in den Netzwerkverwaltungsmodus, mit angepasstem Sidebar.
- Dialoge für die einzelnen Websites sind aus dem oberen Toolbar erreichbar unter:
„Meine Websites | <Website Titel>“ (My Sites | <Site name>).
- In der Netzwerkverwaltung legt man neue Websites und Benutzer an.
- Unter „Netzwerkverwaltung | Plugins“ etwaige vorher deaktivierten Plugins, z.B. "Coming soon" nun "Netzwerkweit aktivieren".

2 Was ist nun wo?

1. Der Super-Admin hat (wenn gewollt) eine Doppelrolle!

- Beim Einloggen sieht er die üblichen Menüpunkte zur Verwaltung der Stammwebsite, einschließlich Seiten, Beiträge, Menüs usw.
- Darüber hinaus gewährt ihm ein besonderes Dropdown-Menü an der linken Seite der oberen Leiste ein Menüpunkt „Meine Websites | Netzwerkverwaltung | Einstellungen“

den Zugang zu einem speziellen Modus, in dem die Menüpunkte des linken Sidebars sich ändern, ausgerichtet auf die Verwaltung des Website-Netzwerks. Darin gibt es neben Menüpunkte von Plugins, die besondere MultiSite-Features anbieten, weitere Menüpunkte von WordPress selbst zu diesem Zweck. Zum Beispiel, um neue Benutzer anzulegen, und dabei eine Option um einen neuen Benutzer auch zum Super-Admin zu machen. Vor allem kann der Super-Admin hier Administratoren für die einzelnen Websites bestimmen – wobei ein Benutzer auch mehrere Websites administrieren kann. Dieser Netzwerkverwaltungsmodus ist auch am URL erkennbar, an der Komponente „/wp-admin/**network**“. Rückkehr zum „normalen“ Modus ist über dem oberen Menü möglich: Meine Websites | <Stammwebsite> | Dashboard.

2. Während Permalinks weiterhin funktionieren, erhalten Blog-Beiträge der Stamm-Website (die zuerst angelegte) eine zusätzliche Komponente „blog“ in ihren URLs: **domain.com/blog/POSTNAME**. Dies ist Absicht, um etwaige Kollisionen zu vermeiden mit weiteren MultiSites in Pfadmodus. Achtung: Obiges gilt nur für Beiträge, nicht für feste Seiten, die direkt unter dem Basis-URL erreichbar bleiben: domain.com/PAGENAME.
3. Die Stamm-Website verwendet für Uploads das traditionelle Verzeichnis /wp-content/uploads/. Alle weiteren MultiSites speichern ihre Uploads unter /wp-content/uploads/sites/, in eigenen nummerierten Unterverzeichnissen, entsprechend der automatisch vergebenen Nummer der MultiSite, wie sie in der Datenbank gespeichert ist.
4. Alle Plugins werden auf der Plugin-Seite der Netzwerkverwaltung installiert, und können netzwerkweit oder pro Website aktiviert werden.
5. Clevere Plugins merken die Multisite-Umgebung und verhalten sich entsprechend intelligent, je nach ihrer Rolle, hier ein paar Beispiele:
 - iThemes Security zeigt sich nur dem Super-Admin, da es seine Aufgabe einmal für alle einzelnen Websites macht.
 - BackWPup ist ebenso nur für den Super-Admin erreichbar.
 - All in One SEO zeigt sich dem Super-Admin nur in der Top-Bar, weil es generell pro Website mit unterschiedlichen Daten & Aufgaben konfiguriert werden muss.
 - WP Mailto Links ist überhaupt nur für Website-Admins erreichbar.
 - WP-Piwik bietet zwar die Dashboard-Widgets nicht mehr an, geht ansonsten sehr klar und souverän mit um. Für jede Website werden getrennte Statistiken geführt.

Achtung: Plugins werden auch auf der Plugin-Seite der Netzwerkverwaltung gelöscht – müssen aber vorher auf allen Websites deaktiviert werden!

3 Als Super Admin Theme und Plugins laden (Beispiele)

- iThemes Security (iTS), wie immer, plus es taucht darin ein neuer Abschnitt auf: WordPress-Netzwerk-Optimierungen (Multisite) (KVG – maue Features,

und kann angeblich ohnehin Probleme verursachen).

Backend verstecken funktioniert auch hier nach dem neuen Verfahren in ITS – das URL für ein verstecktes Backend enthält noch wp-login.php, erhält dazu ein Token-Parameter für ITS.

- Wordfence (macht Brute-Force, bei ITS ausschalten)
- Ggf. BackWPup (und Ausschluss -Pfadname bzgl. Änderungserkennung in ITS setzen)
- WP Mailto Links, All in One SEO Pack (beide in Betrieb nehmen)
- Advanced Database Cleaner, Comet Cache (beide später aktivieren)

4 Sprachvarianten-Websites

- FRAGE: Soll ich die Stamm-Website für die wichtigste Sprache nutzen?
Hm nicht so leicht zu beantworten, siehe oben Punkt 2.2. Wenn es nicht stört, dass die URLs von Beiträgen, nicht aber feste Seiten, das "/blog" zusätzlich im URL haben, kann man sie nutzen. Wenn man das nicht will, muss man (a) eine weitere Website anlegen, und (b) etwas machen dass die Stamm-Website nicht erreichbar ist. Im Folgenden gehe ich davon aus dass die Stamm-Website für die wichtigste Sprache genutzt wird.
- SO GEHT'S: Die Basis-Website für eine Sprache benutzen:
Je nach Nutzungsart / Personal kann es sinnvoll sein, ein normaler Admin dafür einzurichten, der die Netzwerkverwaltung nicht sehen muss. Siehe unten Abschnitt 6.4.
Wenn ja:
 - Neuer Benutzer (als Super-Admin) anlegen und konfigurieren
 - Zu dem Zeitpunkt kann man keine Rolle / Websitezugehörigkeit spezifizieren
 - Wohl aber nachträglich: Websites | Alle Websites | <Stamm-Website> | Bearbeiten
 - Tab "Benutzer" / Bestehenden Benutzer hinzufügen > Eingeben + Rolle: AdministratorDas klappt so, man kann also die Stamm-Website für die wichtigste Sprache nehmen.
- Lege eine zweite Website an für die zweite Sprache mit seiner Bezeichnung im Pfad, z.B. "de" | "en_GB": Meine Websites | Netzwerkverwaltung | Websites >> Neu hinzufügen
> Nur Kleinbuchstaben (a-z), Ziffern und Bindestriche sind erlaubt >> en-gb
- Sprache der Website: Man kann eine nicht-installierte Sprache wählen, die Sprachdateien dafür werden automatisch heruntergeladen.
- Auch wenn der Website-Titel („Site Title“) zufällig in zwei oder mehr verwendeten Sprachen gleich sein könnte (z.B. „Ayurveda Report“) so ist es sehr hilfreich einen Unterschied zu machen – ansonsten gibt es Stellen in den Menüs und Dialogen, wo der Titel benutzt wird – und man sieht mehrere gleichlautenden Einträge und kann sie nicht unterscheiden.
- Administrator E-Mail-Adresse: *Es wird ein neuer Benutzer angelegt, sofern nicht schon ein Benutzer mit dieser E-Mail-Adresse vorhanden ist. Der Benutzername und das Passwort werden an die angegebene E-Mail-Adresse gesendet. Also wiederhole ggf. die bereits registrierte E-Mail Adresse, wenn der Super-Admin alles alleine machen soll. Aber im Hinblick auf die spätere Pflege durch verschiedene Menschen (Admin / Mütter-*

sprachler) halte ich es für deutlich besser, eigene Website-Admins pro Website (= Sprache) anzulegen.

- Theme-Einstellungen: Am Besten eine Website fertig gestalten, die Einstellungen exportieren, und in die übrigen Websites importieren.
- Die E-Mails die an den neuen Benutzer gehen (gleich 2 davon) haben allesamt nutzlose Login-URLs 'drin, wenn man per ITS das Backend maskiert – nutze weiterhin das URL für die ursprüngliche Website, die funzt – entscheidend ist, welche Kennung sich einloggt.
- Plugin-Einstellungen, die man in der Stamm-Website als Super-Admin (oder auch Admin) getroffen hat, werden in neue Websites nicht übernommen – sie sind wohl "pro Website". Diese und auch die Profildaten des neuen Admins einpflegen.

5 Sprachvarianten-Plugin

<https://de.wordpress.org/plugins/multilingual-press/>

<https://de.wordpress.org/plugins/multisite-language-switcher/>

<https://www.roseindia.net/tutorials/I18N/locales-list.shtml>

<http://demo.icu-project.org/icu-bin/locexp>

<https://myip.info/al>

Da erhebt sich die Frage: WELCHES Plugin für diesen Zweck? Ich hatte zunächst "Multisite Language Switcher", welches oben in den Suchergebnissen steht, probiert – 8.000 Benutzer und 5 Sterne, das kann nicht falsch sein... Upps! Letztes Update vor 5 Monaten, nicht als kompatibel mit der jüngsten WP-Version ausgewiesen... Das muss nichts heißen, aber in der Praxis bin ich damit nicht zurecht gekommen. Es blökte ständig dass nur eine Website konfiguriert sei, auch wenn drei konfiguriert waren)-: Ich habe es nicht zum Laufen gebracht.

Bei nochmaliger Suche fand ich „**MultilingualPress**“ (von Inpsyde GmbH, den machern von BackWPup) – zwar nur 800+ Benutzer und 4.5 Sterne – aber in der Praxis ganz Klasse! Und ein reiner Überbau, kein „Lock In“ - alles läuft weiter wenn man das Plugin löscht – das halte ich für die richtige Philosophie. Und optional lenkt es Besucher automatisch auf die richtige Seite für die Sprache ihres Browsers (locale) um.

- Als Super-Admin "MultilingualPress" installieren und Netzwerkweit aktivieren.
- Der Super-Admin hat nun im Netzwerkverwaltungsmodus zwei neue Untermenüpunkte unter „Einstellungen“:
 - MultilingualPress: Hier kann man 6 globale Einstellungen für das Plugin wählen.
 - Sprachverwaltung: Hier verwaltet man eine Liste von „Sprachen“ - eigentlich Locales. Derzeit sind 174 vorbelegt, ohne Möglichkeit neue Hinzuzufügen. Aber alle Felder lassen sich überschreiben und speichern, so dass falls man ein Locale braucht, das nicht voreingestellt ist (z.B. en-IN – indisches English), dies sich durch überschreiben eines nicht-benötigten Locales erreichen lässt. Die Tabelle mit seinen gut beschrifteten Spal-

ten samt den Hinweisen darunter ist weitestgehend selbsterklärend – bis auf einer Sache: Bei der automatischen Sprachwahl spielt die hier eingestellte Priorität eine wichtige Rolle: Besucht z.B. ein Browser mit einzigem Locale „en-US“ eine Multisite mit en-GB und en-IN Varianten, und die beiden Varianten haben die gleiche Priorität, so wählt MultilingualPress die Sprache en-IN, weil alphabetisch gesehen „English (India)“ vor „English (UK)“ steht – wahrscheinlich nicht die beste Wahl, d.h. in so einem Fall sollte daher en-GB eine höhere Priorität erhalten als en-IN.

- MultilingualPress fügt der Adminseite „Neue Website“ einen neuen Abschnitt, der Seite „Website bearbeiten“ sogar einen weiteren Reiter hinzu. Hier kann man mitunter Sprache und optional ein alternativer Titel wählen.
- Ferner fügt es dem Übersicht „Alle Websites“ drei neue Spalten hinzu: Verknüpfungen, Sprache der Seite, und Weiterleiten.
- Verknüpfungen:
Hier werden die Alternativen Titel der Websites angezeigt, mit denen diese Website verknüpft ist. Verknüpfungen machen dann Sinn, wenn die Seiten der Websites sich i.W. spiegeln. Dies kann auch z.B. eine Mentalitätsfrage sein: Deutsch und Englisch (UK) zu verknüpfen macht sinn, weil die Mentalitäten sich ähneln und daher mit gleichem Aufbau und Bildern angesprochen werden können; während eine Seite für indisches Englisch besser für sich alleine steht, denn die ganz andere Mentalität benötigt andere Inhalte, um das Zielpublikum adäquat abzuholen.
- Sprache der Seite: Wie eingestellt in MultilingualPress, wie oben beschrieben.
- Weiterleiten: Ob die Option ausgewählt ist, diese Website auch für die automatische Weiterleitung gemäß Locale des besuchenden Browsers in Betracht zu ziehen.
- MultilingualPress globale Einstellungen (alle Checkbox-Optionen):
 1. Custom-Posttype-Übersetzer: Habe ich nicht gebraucht; sinnvoll für Event-Kalender.
 2. Erweiterte Übersetzungsfunktionen: Cool – Beim Bearbeiten eines Beitrags steht darunter ein Dialog von MultiLingualPress, in dem man die entsprechenden Beiträge verlinkter Websites direkt bearbeiten kann.
 3. Alternativer Sprachtitel: Zeige Seiten mit ihrem alternativen Sprachtitel in der Admin-Bar.
 4. Quicklink: Zeige Links zu vorhandenen Übersetzungen direkt innerhalb des Beitrags. Ob man das will, ob es sinnvoll ist, hängt wohl von der Art der Websites ab.
 5. HTTP-Weiterleitung: Leite Besucher anhand ihrer Browsereinstellung direkt zur passenden Übersetzung um. Hier wird also auf das vom Browser an die Website übermittelte Sprachpräferenz (genauer: Locale) geschaut. Das macht normalerweise Sinn! Aber ACHTUNG: Dafür wird das **Cookie** PHPSESSID immer gesetzt!
 6. Kombiniertes Papierkorb: Dieses Modul bringt neue Beitragsmetadaten und Checkboxes, um den Beitrag in den Papierkorb zu verschieben. Falls du die Checkbox aktivierst

und den Beitrag dann in den Papierkorb verschiebst, löscht MultilingualPress auch den verknüpften Beitrag in anderen Sprachen.

6 Automatische Sprachwahl für Besucher, Adminkennungen

Das Thema Sprachwahl war für mich nicht einfach. Zunächst die Stellen, wo die Umlenkung auf eine der Websites gemäß Browser-Sprachpräferenzen konfiguriert wird:

1. Global: Network Admin | Settings | MultilingualPress | HTTP Redirect (setzt Cookie)
2. Pro Website: Sites | All Sites | <Site wählen> Edit | (Tab) MultilingualPress | Redirection

Obwohl ich meinte, damit alles richtig eingestellt zu haben, hat es zunächst nicht funktioniert:

- Ich habe die Stamm-Website des MultiSite-Netzwerks für die wichtigste Sprache benutzt, liegt es daran, darf ich das nicht? [War nicht das Problem]
- Die übrigen Netzwerk-MultiSites (zwei Stück) haben beide Englisch als Sprache – sie unterscheiden sich im Locale (en-GB / en-IN) – denn in Indien muss man ganz anders in Bild und Sprache vorgehen als in England. [War nicht das Problem]

Support-Forum-Tipps: [Hier](#) und [Hier](#) und [Hier](#) waren informativ, haben mein Problem aber nicht gelöst.

Schlussendlich gefunden: Die Umleitung erfolgt nicht generell, sondern nur Seite für Seite. Damit eine Umleitung funktioniert, muss die ursprünglich besuchte Seite ein direktes Pendant in der anderen Sprache(n) besitzen. Diese Beziehungen werden im Überbau von MultilingualPress verwaltet, konkret im Dialog „Übersetzung für...“, welches MultilingualPress bei „Seite bearbeiten“ bzw. „Beitrag bearbeiten“ hinzufügt. Falls das Pendant bereits existiert (was bei der ersten Seite und dem ersten Beitrag der Fall ist), kann man die Beziehung nachträglich herstellen mit dem Button „Beziehung ändern“ und dann „Bestehenden {Seite|Beitrag} auswählen...“.

Fazit: MultilingualPress leitet (nur) zwischen Seiten um, zwischen denen eine Verknüpfung eingetragen wurde. Damit diese Umleitung funktioniert:

1. Netzwerkverwaltung | Einstellungen | MultilingualPress | HTTP-Weiterleitung (mit Häkchen, dafür gibt es ein Cookie)
2. Netzwerkverwaltung | Websites | <Website bearbeiten> | (Reiter) MultilingualPress...
 - Sprache wählen
 - Verknüpfungen – Häkchen setzen wo Verknüpfungen und gemeinsame Bearbeitung funktionieren sollen – ermöglicht gleichzeitiges Bearbeiten, autom. Weiterleitung braucht zusätzlich:
 - Weiterleitung (nur bei der primären Website mit Häkchen falls auch manuelle Sprachwahl)
3. Verknüpfung Seite für Seite durchführen (am Ende der Maske „{Seite|Beitrag} bearbeiten“)

4. Achtung: **Die Bearbeitung von Verknüpfungen / verknüpften Seiten/Beiträgen setzt voraus, dass der eingeloggte Benutzer Schreibrecht bei den jeweiligen MultiSites besitzt.** Wenn z.B. für jede MultiSite eine eigene Adminkennung angelegt wird, so fehlt ihr (der Kennung) dieses Recht und MultilingualPress hat von den Rechten her keine Möglichkeit, die Bearbeitung der Inhalte von verknüpften Seiten gleichzeitig in einem Dialog anzubieten. In dem Fall müsste jede Website für sich entstehen, und die Verknüpfungen nachträglich vom Super-Admin eingesetzt werden. Eine leichtgängigere Möglichkeit ist, eine Art „Multi-Admin“ Kennung (z.B. „translator“ oder „multiadmin“) anzulegen, und ihr das Adminrecht bei allen Websites zu geben – nicht aber „Super-Admin“ Recht, wenn die Arbeit der Netzwerkverwaltung von übergeordneter Stelle (dem Webprogrammierer) erfolgen soll, bzw. wenn der Übersetzer nicht mit ihm unvertrauten Menüpunkten behelligt werden soll.

Wer nach der Hammermethode (ohne einzelne Seiten zu berücksichtigen) arbeiten will: Es gibt ein anderes Plugin, das vermutlich so arbeitet: [Language Redirect](#) – das auch klar formuliert, dass es keinen Inhalt auf der Stamm-Website des MultiSites erlaubt. Mit anderen Worten, das Plugin lenkt immer vom Stamm-URL zu einer der MultiSites um.

7 Manuelle Sprachwahl

MultilingualPress bietet ein „Language Switcher“ (Deutsch: Sprachwechsler) Widget an, dass man wie gewohnt in ein Widget-Bereich des Themes einsetzen kann. Hier wählt man:

- die Art des Linktexts aus einer Dropdown
- ob auch eine Flagge anzuzeigen ist. Das URL zum Fahnenbildchen (als Netzwerk-Admin) jeweils setzen: Websites | <Website bearbeiten> | (Tab) MultilingualPress | Flaggen-URL
- ob die aktuelle Website anzuzeigen ist
- ob nur Links für übersetzte Inhalte anzuzeigen sind

Achtung: Die manuelle Sprachwahl wird von der automatischen Sprachwahl (gemäß Browser-Präferenz) übertrumpft, wenn das URL nicht explizit zu einer bestimmten Sprache führt (z.B. .../de/...), sondern zur Stamm-Website. Empfohlene Einstellungen:

- Netzwerkverwaltung | Einstellungen | MultilingualPress | HTTP-Weiterleitung Beliebig – ein Cookie ist der Preis für das Besucher-Komfort.
- Netzwerkverwaltung | Websites | <Website bearbeiten> | (Reiter) MultilingualPress...
- Die primäre Website braucht hier Weiterleitung (Redirection) aktiviert => Wird nur Angeboten, wenn HTTP-Weiterleitung ausgewählt ist.
- Alle anderen Websites brauchen (ggf.) Weiterleitung **deaktiviert**, sonst sind sie per manuelle Wahl nicht erreichbar, weil die automatische Weiterleitung auf die vom Browser bevorzugte Sprache umlenkt.

Mehr dazu: <https://wordpress.org/support/topic/cookie-sessions/>

7.1 Flaggen

Gratis-Fahnen in nüchternen Form und brauchbare Größe zu finden ist gar nicht so einfach, ich habe länger gesurft um diese zu finden:

- <http://www.famfamfam.com/lab/icons/flags/> Schön aber sehr klein – 16 x 11 px
Attribution optional
- <http://www.icondrawer.com/flag-icons.php> 16×16, 24×24, 32×32, 48×48, 64×64, 128×128
Attribution required for commercial use; rechteckig mit Transparenz oben/unten (crop)
- <https://www.content-pack.com/category/freebies/world-flags/> 16x10, 24x16, 48x32, ...
Attribution optional. Leider fehlt die mE nützlichste Größe 32x24.
- <https://www.gosquared.com/resources/flag-icons/> 16×16, 24×24, 32×32, 48×48, 64×64
Attribution optional. In 2 Varianten: Flat, Glossy. Sehr schön. Geliefert rechteckig mit Transparenz, wenn man selbst die Transparenz von 32x32 abschneidet, erhält man 29x19 Icons.

8 Sitemap.xml und Robots.txt

Leidiges Thema, bis man hier durchsteigt!

8.1 Sitemap.xml

Diese Datei(en) muss man dynamisch von einem Plugin erzeugen lassen. Für normalen WordPress (ohne MultiSite) erfüllen die meisten gängigen Plugins in diesem Bereich ihren Soll. Aber bei MultiSite lassen auch manche führende Plugins den User im Stich:

- **All in One SEO - mein Favorit**
Das Feature „XML Sitemaps“ muss man in jeder Website extra einschalten. Per Standard-Einstellung erzeugt das Plugin dann *ein* dynamisches Sitemap mit allen Einträgen unter dem URL "{/subdir}/sitemap.xml". Optional kann es auch mehrere indizierte Sitemaps erzeugen. Die Einstellung für eine statische Datei ist in der Multisite-Umgebung nicht wählbar (ausgegraut).
- **XML Sitemap & Google News feeds** (by RavanH)
Explizite MultiSite-Unterstützung, dürfte funktionieren.
Ungetestet, weil All in One SEO deutlich mehr macht und auch funktioniert.
- **YOAST SEO** (funktioniert NICHT)
Dashboard | Funktionen | Erweiterte Einstellungen muss aktiviert werden, trotzdem erzeugt das Plugin für Subdir-Websites leere Sitemaps.
- **Google XML Sitemaps** (von Arne Brachhold; funktioniert NICHT)
Das Plugin funzt leider nicht mit MultiSite - erzeugt leere Sitemaps.

8.2 Robots.txt

WordPress selbst erzeugt nur für die Stamm-Website eine robots.txt, die aber nur das Sitemap der Stamm-Website verlinkt. Upps. Es führen mehrere Wege zur vollständigen Verlinkung:

1. Am einfachsten und sichersten ist es, eine robots.txt für die Stammwebsite manuell zu erstellen und hochladen, welche alle Sitemaps – auch die der Subdomain-Websites, verlinkt.
2. Wer unbedingt eine robots.txt für jede Website extra haben will (auf die Gefahr hin dass die Subdomain-Websites nicht von allen Suchmaschinen als solche erkannt werden) kann sie
 - a) Manuell erzeugen und hochladen, oder
 - b) Per „Robots.txt“ Feature von „All in One SEO“ als statische Dateien erzeugen.
Achtung: Das Feature kann aktiviert werden aber bleibt unsichtbar, wenn z.B. per iThemes Security den Datei-Editor deaktiviert wird.
3. Das Plugin [Multisite Robots.txt Manager](#) verwenden (scheint mir Overkill).

9 Nginx

Achtung: Die allermeisten Anleitungen / Configs für Nginx-MultiSite im Netz (auch die im WordPress Codex <https://codex.wordpress.org/Nginx>) beziehen sich auf frühere Versionen von WordPress – vor WordPress 3.5. Mit 3.5 wurde ein neues, einfacheres Paradigma eingeführt, das nur triviale Änderungen der Nginx-Config gegenüber ein normales WordPress benötigt:

Your first site on a fresh install will put uploaded files in the traditional location of /wp-content/uploads/, however all subsequent sites on your network will be in the /wp-content/uploads/sites/ folder, in their own subfolder based on the site number, designated by the database.

This is a change from Multisite 3.0-3.4.2, where images of subsites were stored in /wp-content/blogs.dir/ and were shown in: http://example.com/files/ and http://example.com/sitename/files and so on. If you started with a Multisite install older than 3.5, it is not an error if your images show with the URL of /files/.

Was heutzutage wirklich passiert muss man durch Experiment und Beobachtung klären:

- Bilder, die auf die primäre Website (Stamm-Website) hochgeladen werden, landen unter /wp-content/uploads/, wie beim normalen WordPress.
- Wenn man die Möglichkeit von MultilingualPress nutzt, die Übersetzungen alle auf einmal zu bearbeiten, so gibt es bei jeder Sprache ein Button „Copy source post“. Dabei werden Bilder mit unverändertem URL übernommen.
- Wechselt man als Bearbeiter in eine andere Website als die Stamm-Website, und fügt direkt ein Bild über die Mediathek von WordPress ein, so wird dieses Bild wie oben er-

läutert abgelegt und direkt unter der vorhandenen Adresse verlinkt, z.B.

<https://beispielcom/es/wp-content/uploads/sites/3/2017/08/beispiel.jpg>

Diese Adresse an sich existiert nicht, da das Subdirectory der Website (hier „es“) darin steckt. Ohne das Subdir stimmt die Adresse.

Wenn wir diesen Fall nicht extra in der Nginx-Config abfangen würden, so fiel die Anfrage i.d.R. auf ein „try_files“ durch, und würde auf /index.php umgeleitet. Das sollte zwar funktionieren – WordPress PHP soll die Bilddatei ausliefern – aber wir hätten PHP unnötig für eine statische Datei bemüht. Folgende Regel in der Nginx-Config entfernt das Subdir vom Anfang der Anfrage: `rewrite ^(/[_0-9a-zA-Z-]+)?(/wp-(content|admin|includes).*) $2 last;`

- Blogseiten (Beiträge) haben alle ein „/blog“ im URL eingespreizt, und somit URLs wie: `/blog/hello-world/` oder `/es/blog/hello-world/`

Diese fallen typischerweise in Nginx zum Permalink-Catchall durch und werden zu `/index.php` umgeleitet. Das ist soweit korrekt.

Die Medien-Uploads werden nach gleichem Schema behandelt wie bei den Seiten, und werden ebenfalls somit direkt von Nginx ausgeliefert.

Fazit: Der vielfach im Netz erwähnte Mechanismus mit Nginx „map“ Direktive ist hinfällig. Daher **alle veraltet:**

- <https://www.nginx.com/resources/wiki/start/topics/recipes/wordpress/>
- <https://easyengine.io/wordpress-nginx/tutorials/multisite/static-files-handling/>
- <https://premium.wpmudev.org/blog/wordpress-multisite-wordpress-nginx/>
- <https://wordpress.org/plugins/nginx-helper/>

Es reicht also, in eine normale Nginx-WordPress-Config folgenden Block einzufügen:

```
# BEGIN MULTISITE

# NUR falls man händisch oder per "All in One SEO" eine 'robots.txt' für *jede* Website hochlädt
bzw. schreibt.
# Eine einfachere (und sicherere) Strategie ist es, in einem manuell erzeugten robots.txt für die
Stamm-Website die Sitemaps
# aller Subdomain-Websites auch zu verlinken. Dann ist es egal ob eine Suchmaschine die Subdomains
als eigenständige Websites erkennt.
# location ~ ^(/[_0-9a-zA-Z-]+)?/robots.txt {
# Prefer static file, otherwise fall back to a virtual robots.txt e.g. from WordPress itself or
All in One SEO Sitemap
# try_files $uri /index.php;
# access_log off;
# add_header Debug "MultiSite robots.txt";
# }

# Begin for WordPress SEO Sitemap (entkommentieren je nachdem welches Plugin Du benutzt)

# 1. All in One SEO
# Per Standard-Einstellung erzeugt das Plugin *ein* dynamisches Sitemap mit allen Einträgen unter
dem URL "{/subdir}/sitemap.xml"
# Die Einstellung für eine statische Datei ist in der Multisite-Umgebung nicht wählbar
(ausgegraut).
# Das URL muss abgefangen werden, sonst geht es weiter unten in den Zweig für XML-Dateien und wird
nicht gefunden.
rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap\.xml$ /index.php last;
```

```
rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap\.xml\.gz$ /index.php last;
# Wer in den Optionen "Enable Sitemap Indexes" auswählt, braucht auch folgende Regeln:
# A) "Komprimierte Sitemap erstellen" abgewählt
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_addl\.xml$ /index.php last;
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_post\.xml$ /index.php last;
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_page\.xml$ /index.php last;
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_archive\.xml$ /index.php last;
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_author\.xml$ /index.php last;
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_category\.xml$ /index.php last;
# B) "Komprimierte Sitemap erstellen" ausgewählt (default)
# .gz Dateien werden vom *Browser* heruntergeladen, nicht angezeigt, ist zu erwarten, ist OK
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_addl\.xml\.gz$ /index.php last;
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_post\.xml\.gz$ /index.php last;
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_page\.xml\.gz$ /index.php last;
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_archive\.xml\.gz$ /index.php last;
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_author\.xml\.gz$ /index.php last;
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_category\.xml\.gz$ /index.php last;

# 2. XML Sitemap & Google News feeds (by RavanH)
# Explizite MultiSite-Unterstützung, dürfte funktionieren.
# Ungetestet, weil All in One SEO deutlich mehr macht und auch funktioniert.

# 3. YOAST
# Dashboard | Funktionen | Erweiterte Einstellungen muss aktiviert werden,
# trotzdem erzeugt das Plugin für Subdir-Websites leere Sitemaps.
# rewrite ^(/[_0-9a-zA-Z-]+)?/sitemap_index\.xml$ /index.php?sitemap=1 last;
# rewrite ^(/[_0-9a-zA-Z-]+)?/([^\+?)-sitemap([0-9]+)?\.xml$ /index.php?sitemap=$1&sitemap_n=$2
last;
# rewrite ^(/[_0-9a-zA-Z-]+)?/([^\+?)-sitemap.xsl$ /index.php?xsl=$1 last;

# 4. Google XML Sitemaps (von Arne Brachhold)
# Das Plugin funzt leider nicht mit MultiSite - erzeugt leere Sitemaps )-:

# End for WordPress SEO Sitemap

# Your first site on a fresh install will put uploaded files in the traditional location of /wp-
content/uploads/, however all subsequent sites
# on your network will be in the /wp-content/uploads/sites/ folder, in their own subfolder based on
the site number, designated by the database.
# These files will be accessible via that URL.

# This is a change from Multisite 3.0-3.4.2, where images of subsites were stored in /wp-
content/blogs.dir/ and were shown in:
# http://example.com/files/ and http://example.com/sitename/files and so on. If you started with a
Multisite install older than 3.5,
# it is not an error if your images show with the URL of /files/.

# Postpend a missing "index.php" to the standard and network admin URLs
# Standard /wp-admin/ must be done extra as the directory exists, !-e $request_filename fails
rewrite ^/wp-admin/$ /wp-admin/index.php last;
rewrite ^/wp-admin/network/$ /wp-admin/network/index.php last;

# If the request is NOT for a file or directory that exists, perform any necessary multisite-
specific rewriting
if (!-e $request_filename) {

    # Redirect browser adding trailing slash to wp-admin requests, preserving any website-
subdir up front
    # If the preceding nginx config does not configure directory indexing with 'index.php'
this will not be enough
    rewrite ^(/[_0-9a-zA-Z-]+)?/wp-admin$ $scheme://$host$uri/ permanent;
```

```
# Catch any call to {/subdir}/wp-admin/ and rewrite it to /wp-admin/index.php
rewrite ^(/[_0-9a-zA-Z-]+)?/wp-admin/$ /wp-admin/index.php last;

# Delete any website-subdir in front of the known "wp-" elements
rewrite ^(/[_0-9a-zA-Z-]+)?(/wp-(content|admin|includes).*) $2 last;

# Similarly, delete any website-subdir in front of ".php" files
rewrite ^(/[_0-9a-zA-Z-]+)?(/.*\.php) $2 last;
}

# END MULTISITE
```

10 Nützliche MultiSite-Tipps und -Plugins

- In der MultiSite-Umgebung können einzelne Site-Administratoren in TinyMCE keine `<iframes>` speichern, das klappt nur für den Super-Administrator.
- [R3DF Multisite Blog Slug Remover](#) - ungetestet, da seit 2 Jahren nicht aktualisiert.
- [MultiSite Clone Duplicator](#) - auch ungetestet, aber klingt gut falls man nachträglich eine neue Sprache hinzufügen will.
- [Dropdown multisite selector](#) - eine Alternative zum oben erwähnten [Language Redirect](#).
- [Multisite Post Duplicator](#) - hervorragende Bewertung, aber unnötig, wenn man MultilingualPress verwendet, da dieser die Verknüpfung und Bearbeitung selber anbietet.
- [Multisite Plugin Manager](#) - wer mit vielen Plugins arbeitet, und sie pro Website differenziert aktivieren will, dürfte hiermit gut bedient sein.

Es gibt mehr - in WP-Plugins einfach nach [Multisite suchen](#).