



WordPress mit Composer

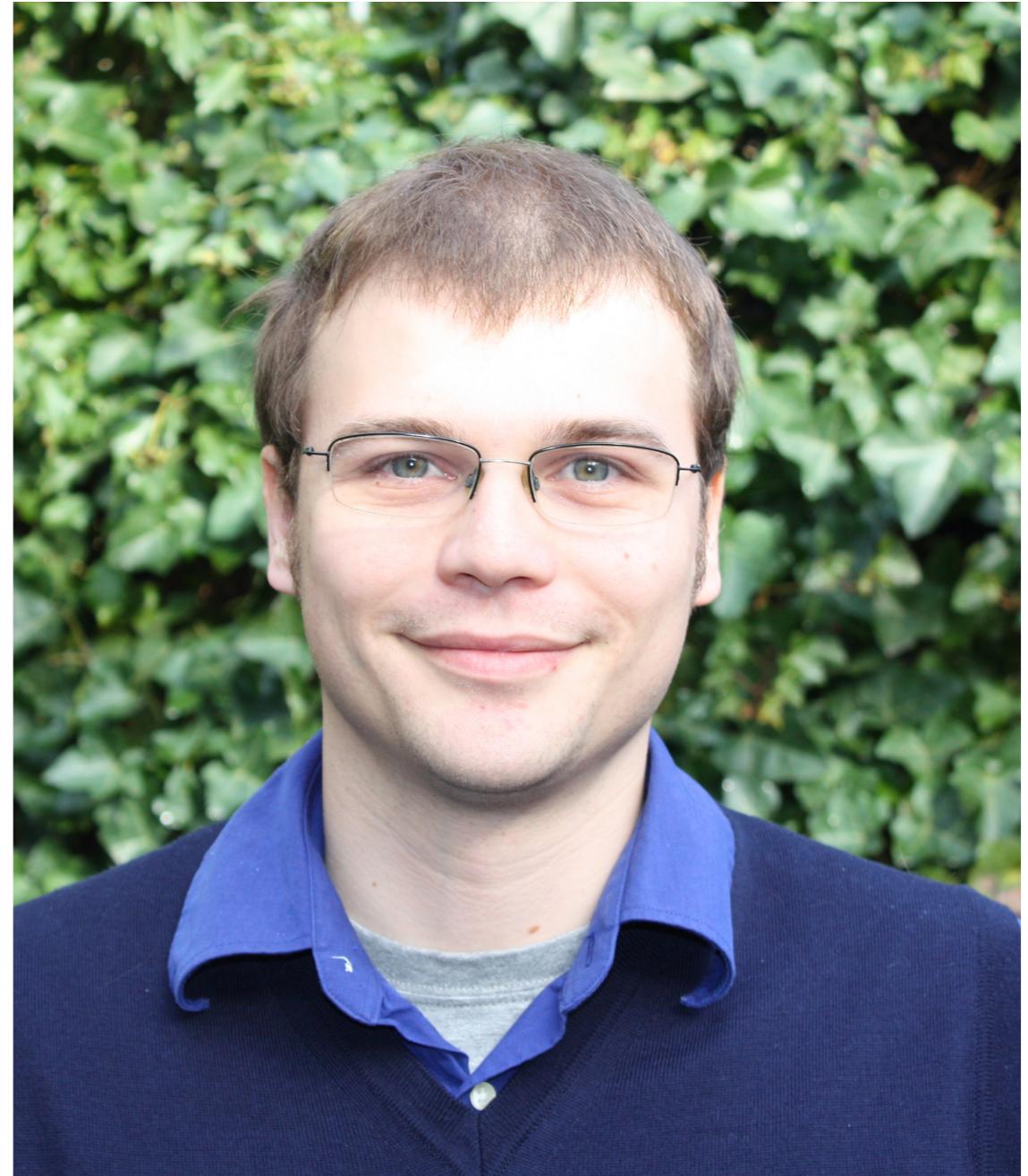
**WordPress-Installation mit
Composer aufsetzen**



Über mich

Simon Schmidt

- + Informatik Studium in Saarbrücken
- + seit 2000 im Internet
- + bei MFC seit 2009
- + (TYPO3)-Programmierung, Datenanalyse & technisches SEO





Wordpress mit Composer

Inhaltsübersicht



1. Motivation
2. composer.json aufsetzen
3. Eigenes (Child-)Theme hinzufügen
4. Sprachen berücksichtigen?
5. Umgebungsvariablen auslesen (Credentials nicht mehr in der Versionierung)



Wordpress mit Composer

Motivation

- WordPress-Updates kommen zum Teil recht überraschend und kurzfristig
- bei jedem Update kommen unerwünschte Themes und Plugins wieder mit dazu
- Updates für Sprachen häufig erst „später“ verfügbar, was insbesondere bei Security Issues schlecht ist
- Wunsch zu mehr Automatisierung



Wordpress mit Composer

1. composer.json aufsetzen

- WordPress-Projekt-Struktur definieren und in composer.json abbilden (Bsp.: <https://roots.io/bedrock/> bzw. <https://github.com/roots/bedrock>)
 - Ablageort der Plugins
 - Ablageort der WP-Installation an sich
 - Ablageort der Themes allgemein und des Child-Themes
- wpackagist.org als Repository für die Includes hinzufügen
- Core über „johnpbloch/wordpress“ einbinden
- config/application.php anlegen
- Im Ordner „web“ Files .htaccess, index.php und wp-config.php anlegen
- Übersetzungen des Cores (sofern gewünscht) über z.B. „koodimonni-language/core-de_DE“ einbinden
- Plugins sind über „wpackagist-plugin/*“ verfügbar



Wordpress mit Composer

2. Eigenes (Child-)Theme hinzufügen

- composer.json für eigenes (Child-)Theme anlegen
- Ablageort des Repositories in der Projekt composer.json hinterlegen
- Abhängigkeit von Parent-Theme sofern vorhanden hinzufügen
- Abhängigkeit von der entsprechenden Sprachversion des Parent-Themes hinzufügen
- Build Prozess für die CSS und JS-Dateien einrichten (sofern gewünscht)



Wordpress mit Composer

3. Sprachen

- Sprach-Repositories für Wordpress sind einige bereits über Composer verfügbar (<https://wp-languages.github.io> als Repository in Projekt composer.json anlegen)
- Insbesondere Core und Themes sind unter „koodimonni-language/*“ bzw. „koodimonni-theme-language/*“ bereits zu finden
- einzig bei den Plugins gibt es aktuell noch keine fertige Lösung
 - Workaround:
 - Plugin-Sprach-Updates noch über das WordPress-Backend und damit über die Versionierung
 - Plugin-Sprach-Updates über wpcli (wp language plugin update —all)



Wordpress mit Composer

4. Umgebungsvariablen verwenden und auslesen

- `.env.example` Datei im Repository für die Zugangsdaten auf dem lokalen System
- `post-update-cmd` für composer, der auf dem System eine `.env` Datei erzeugt, sobald ein „composer update“ gemacht wird sowie Generierung der Environment Salts über `wpcli`
- benötigte composer-Pakete:
 - `vlucas/phpdotenv`
 - `oscarotero/env`
 - `aaemnnosttv/wp-cli-dotenv-command`



Vielen Dank für Eure Aufmerksamkeit!

FRAGEN!?

@sfsmf

sfs@marketing-factory.de

Repository: https://github.com/sfsmf/wp_composer_starter/

Folien demnächst unter: <https://blog.marketing-factory.de>