## Initiales Repo aufsetzen

Kevin Pfeifer - Version 1.0, 2019-07-03

- 1. Ein neues Repository bei Bitbucket erstellen und dieses lokal auschecken (Beispiel: kcc)
- 2. Den Master Branch von unserem wordpress Repository herunterladen https://bitbucket.org/sunlimeitservices/wordpress/src/2019\_git\_local\_setup\_oop/ Über das linke Menü "Downloads" → Branches → "master" ZIP
- 3. Den Inhalt dieses ZIPs in das zuvor ausgecheckte neue Repository entpacken.

## Darauf achten dass die .gitignore mit kopiert wird!

- 4. Die mktheme.sh öffnen und die Werte NAME, SLUG und DESC durch die eigenen Theme Strings ersetzen.
- 5. Die mktheme.sh ausführen ⇒ ./mktheme.sh und danach löschen.
- 6. Diesen Stand in das Repository commiten und pushen.
- 7. Am Batman Server mit dem User batcave anmelden und in den Ordner chroot gehen.
- 8. Dort das zuvor neu erstellte Repository auschecken. Damit sollte an unserem Beispiel ein Ordner kcc im chroot Order liegen.
- 9. npm i am Batman in dem ausgecheckten Repository durchführen.
- 10. Nun sollte die Seite zumindest über folgende URL aufrufbar sein: https://<repo-name>.sunlime.io
- 11. Über https://batman.sunlime.at/phpmyadmin/ sich mit dem batcave Benutzer anmelden und eine neue Datenbank erstellen. An unserem Beispiel heist diese Datenbank dann batcave\_kcc
- 12. Den Wordpress Installer auf der Domain https://<repo-name>.sunlime.io durchführen mit der zuvor neu erstellten Datenbank und den Anmeldedaten vom batcave Datenbank-Benutzer (siehe 1Password)

  Die Serveradresse muss hier 127.0.0.1 sein da (wegen der aktuellen chroot-Umgebung) localhost nicht funktioniert.
- 13. In dem lokal ausgecheckten Repo das gulpfile.js öffnen und im Task uploads, db und default folgende Werte anpassen:

```
gulp.task('uploads', function() {
   var remote_user = 'batcave';
   var remote_server = 'batman.sunlime.at';
   var remote_path = '/home/batcave/chroot/kcc/docroot/wp-content/uploads/';
   ...
});
```

```
gulp.task('db', function() {
  var remote_user = 'batcave';
  var remote_server = 'batman.sunlime.at';
 var remote_path = '/home/batcave/chroot/kcc/docroot'; // without / at the end
 var domain = 'https://sunlime:sunlime.io@kcc.sunlime.io'; // without / at the end
  var mysql_running = fs.existsSync('/tmp/kcc.mysql.sock');
  if(!mysql_running) {
    if(!fs.existsSync('mysql_data')) {
      exec.execSync('mkdir mysql_data && /usr/local/mysql/bin/mysqld --skip-networking --initialize -h
./mysql_data');
   }
    exec.exec('/usr/local/mysql/bin/mysqld --skip-networking --socket=/tmp/kcc.mysql.sock --skip-grant-
tables -h ./mysql_data');
 }
});
                                                                                                           JS
gulp.task('default', function() {
 exec.exec('killall -9 mysqld; /usr/local/mysql/bin/mysqld --skip-networking --socket=/tmp/kcc.mysql.sock
--skip-grant-tables -h ./mysql_data');
});
```

- a. Beim uploads Task muss der remote\_path der absolute Pfad zum Uploads Ordner sein
- b. Beim db Task muss der remote\_path der absolute Pfad zum Document-Root sein
- c. Der MySQL-Socket muss hier überall /tmp/<repo-name>.mysql.sock sein, hier also /tmp/kcc.mysql.sock
- 14. Ebenso muss der Pfad zur mysql.sock Datei in der sqldump.php und php.ini Datei auch angepasst werden:
  - a. <repo-name> ⇒ an unserem Beispiel kcc

```
<?php
              /** PHP version of mysqldump cli that comes with MySQL ...*/
15
16
17
18
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
40
41
42
43
44
45
46
47
               namespace Ifsnop\Mysqldump;
               use ...
              if(php_sapi_name() === 'cli' && !file_exists( filename: 'docroot/wp-config.php') && file_exists( filename: 'doc
$chars = 'ABCDEFGHIJKLMNDPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!"#$%&()**,-./:;<=>?@[]^_`{|}~';
                                                                                                                                                                                                              'docroot/wp-config-sample.php')) {
                  $sample = file( filename: 'docroot/wp-config-sample.php');
                  foreach($sample as $key => $line) {
   if(strpos($line, needle: '$table_prefix') !== false) {
    $sample[$key] = '$table_prefix = \''.exec( command: "grep 'DROP TABLE IF EXISTS' sqldump.sql | cut -d _ -f 1 | cut -d '`' -f 2").'_\';'.PHP_EOL;
                     if(strpos($line, needle: 'Füge hier deine Zeichenkette ein') !== false) {
    $sample[$key] = str_replace( search: 'Füge hier deine Zeichenkette ein', substr(str_shuffle($chars), start: 0, length: 64), $line);
                  unset($sample[0]):
                  array_unshift( &array: $sample, ...vars: 'define(\'WP_SITEURL\', \'http://localhost:8000\');'.PHP_EOL);
array_unshift( &array: $sample, ...vars: 'define(\'WP_HOME\', \'http://localhost:8000\');'.PHP_EOL);
array_unshift( &array: $sample, ...vars: '<?php'.PHP_EOL);
                  file_put_contents( filename: 'docroot/wp-config.php', $sample);
               foreach(file( filename: php_sapi_name() === 'cli' ? 'docroot/wp-config.php' : 'wp-config.php') as $line) {
   if(strpos($line, needle: 'DB_HOST') !== false) eval($line);
   if(strpos($line, needle: 'DB_MASSWORD') !== false) eval($line);
   if(strpos($line, needle: 'DB_NASSWORD') !== false) eval($line);
   if(strpos($line, needle: 'DB_NAME') !== false) eval($line);
48
49
50
51
52
53
54
55
56
57
58
60
61
62
63
64
65
66
              if(php_sapi_name() === 'cli') {
    shell_exec( cmd: 'echo CREATE DATABASE IF NOT EXISTS '.DB_NAME.' | /usr/local/mysql/bin/mysql -S /tmp/<repo-name>.mysql.sock');
    shell_exec( cmd: '/usr/local/mysql/bin/mysql -S /tmp/<repo-name>.mysql.sock '.DB_NAME.' < sqldump.sql');</pre>
                   else {
$dump = new Mysqldump( dsn: 'mysql:host='.DB_HOST.';dbname='.DB_ME,  user: DB_USER,  pass: DB_PASSWORD);
                   $dump->start();
                * PHP version of mvsaldump cli that comes with MvSOL
                 * mysql mysqldump pdo php7 php5 database php sql
               * @category Library

* @package Ifsnop\Mysqldump

* @author Diego Torres <ifsnop\@github.com>

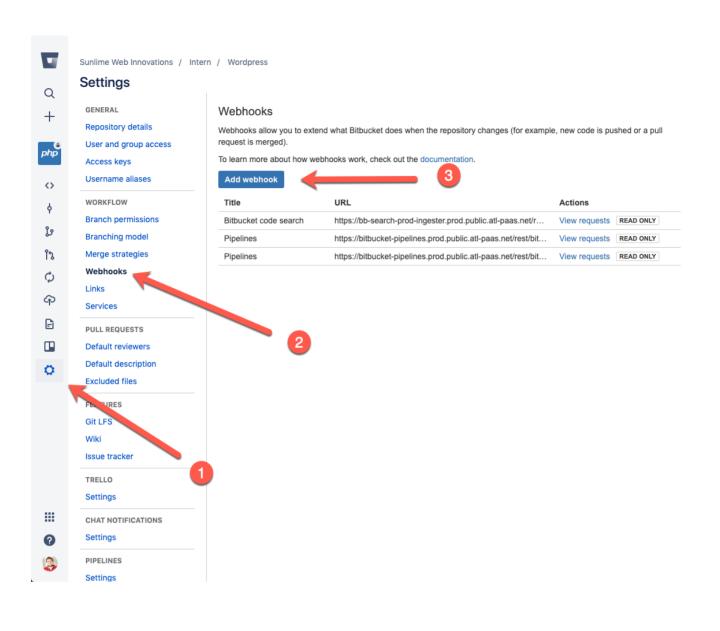
* @license http://www.gnu.org/copyleft/gpl.html GNU General Public License

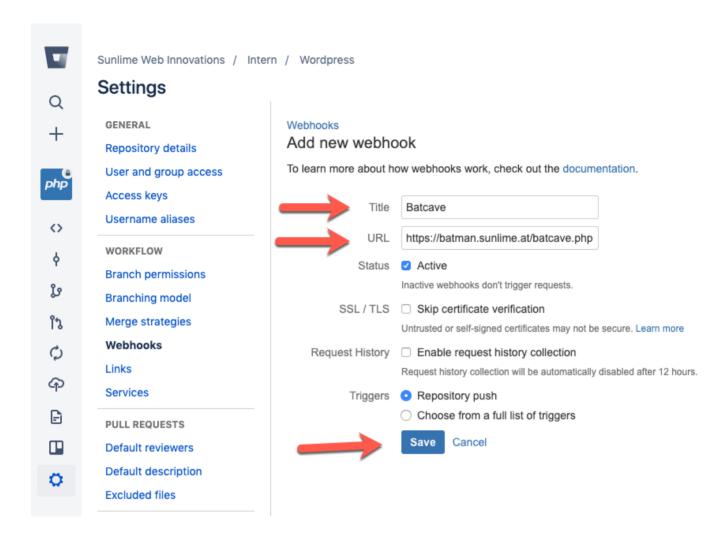
* @link https://github.com/ifsnop/mysqldump-php
```

```
zend_extension = xdebug.so
xdebug.remote_enable = 1

max_execution_time = 300
mysqli.default_socket = "/tmp/<repo-name>.mysql.sock"
pdo_mysql.default_socket = "/tmp/<repo-name>.mysql.sock"
```

- 15. Die Node-Modules installieren npm i
- 16. Der Befehl gulp db sollte nun ohne Fehlermeldung durchlaufen.
- 17. Mit dem Befehl gulp startet der lokale Webserver und die lokale Entwicklung der Webseite kann gestartet werden.
- 18. Damit die in das Repo gepushten Änderungen auch auf den "DEV"-Server durchgeführt werden muss ein Webhook bei Bitbucket eingerichtet werden.





a. Title: Batcave

URL: https://batman.sunlime.at/batcave.php

Version 1.0 Last updated 2019-07-16 13:36:21 +0200