

DBWT Laragon Setup

with Composer, PostgreSQL, NPM, Apache,
Redis, Laravel

Nilusche Liyanaarachchi (INF)

19.03.2022

Table of Contents

- Disclaimer [P.3](#)
- Laragon Installation [P.4](#)
- PHP-Version upgrade to Version ^8.0 [P.6](#)
- Setup first Laravel Project [P.10](#)
- PostgreSQL Setup [P.12](#)
- NPM Setup [P.21](#)
- Redis Setup [P.22](#)

- There are many different ways to setup Laravel on a machine
- Laragon offers a wide variety of development tools that can be also installed separately but is intended to ease the development process
- You do not have to use every part of Laragon if you already use a separate installation of a particular module

Laragon

Installation

- You can download the „Full-Version“ of Laragon with the link below:
<https://laragon.org/download/>

Edition

Download Laragon - Full (147 MB)

- **Laragon Full (64-bit):** Apache 2.4, Nginx, MySQL 5.7, PHP 7.4, Redis, Memcached, Node.js 14, npm, git, bitmana...

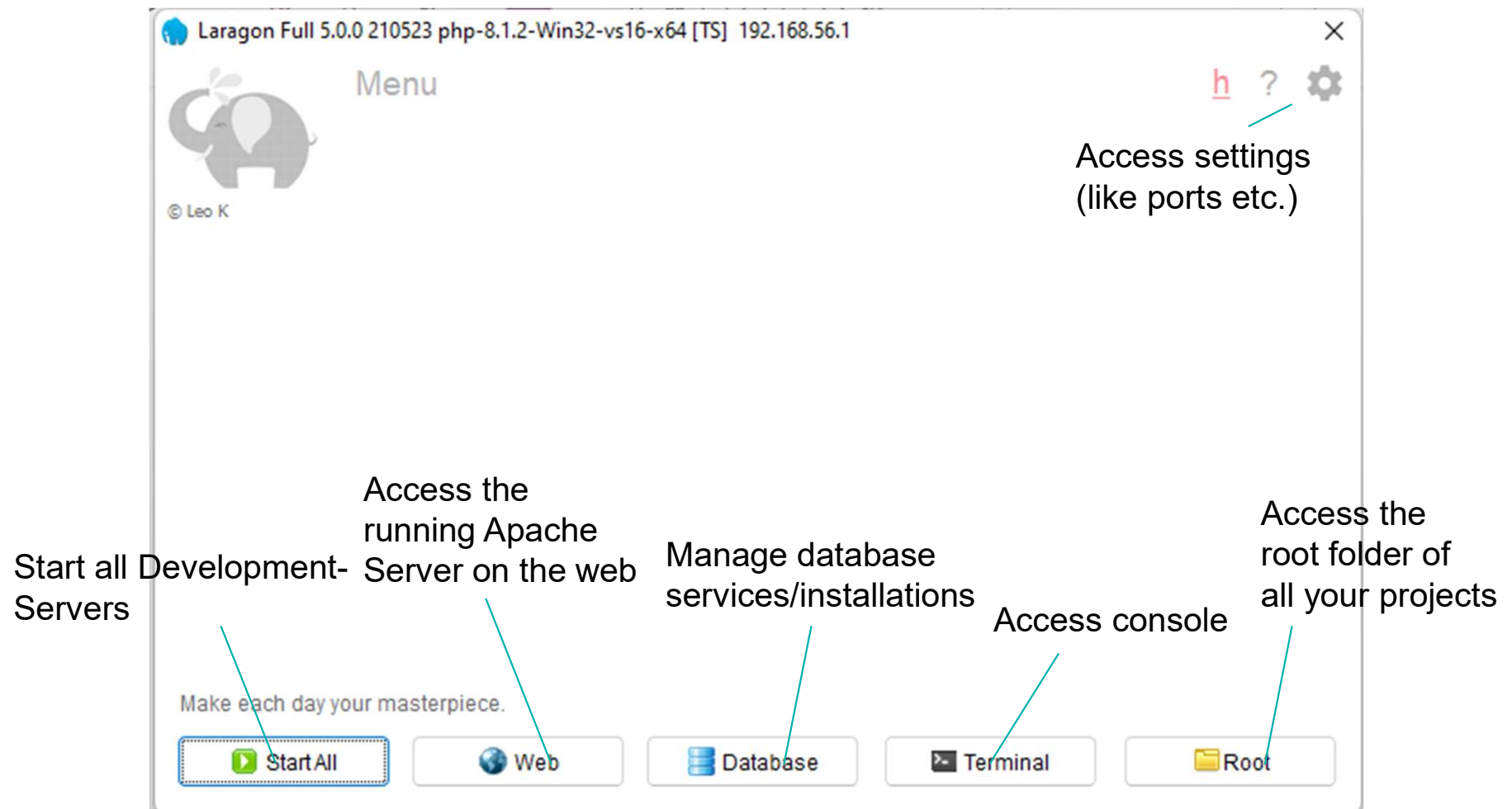
Download Laragon - Portable (38 MB)

- **Laragon Portable:** PHP 5.4, MySQL 5.1, bitmana - Good for getting started with PHP, then you can add newer versions of PHP/MySQL easily later using “Tools > Quick add”
-

Laragon

Installation

- Once installed, here is a quick preview of the Home-Panel:



PHP Upgrade

Preparing for Laravel 9

- Per default, Laragon comes with a PHP installation of version 7.4
- Since the release of Laravel 9 we require at least a version of 8.0
- If you already have the newest version installed, you can skip this part
- <https://windows.php.net/download>

VS16 x64 Thread Safe (2022-Mar-16 10:09:14)

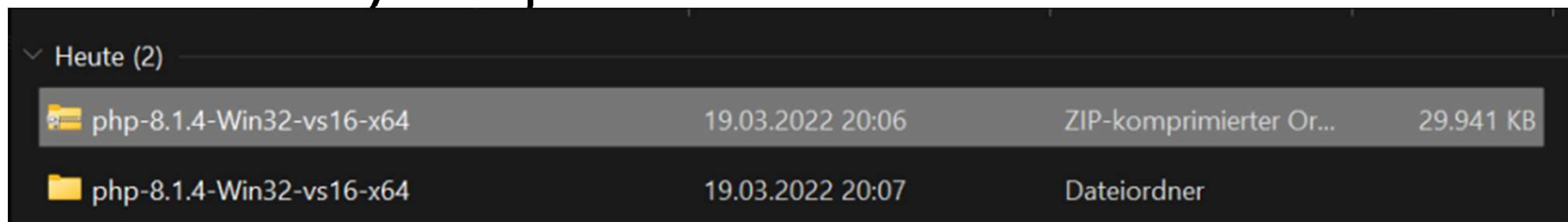
- [Zip](#) [29.24MB]
sha256: b57b67ebd153b793f7f002e3bb345a53163118f0ebd2260c60bf1a9af876185f
- [Debug Pack](#) [23.74MB]
sha256: d263279a92b2ac722657ce1108c3a244a842b35b65eedb6e11a75af738556727
- [Development package \(SDK to develop PHP extensions\)](#) [1.21MB]
sha256: e60ef48d436ff5a2298f3a5ac881f165e0691ce6234b3641e8bd1cbb167182d2

Generally, you should
Download a Thread Safe
Version

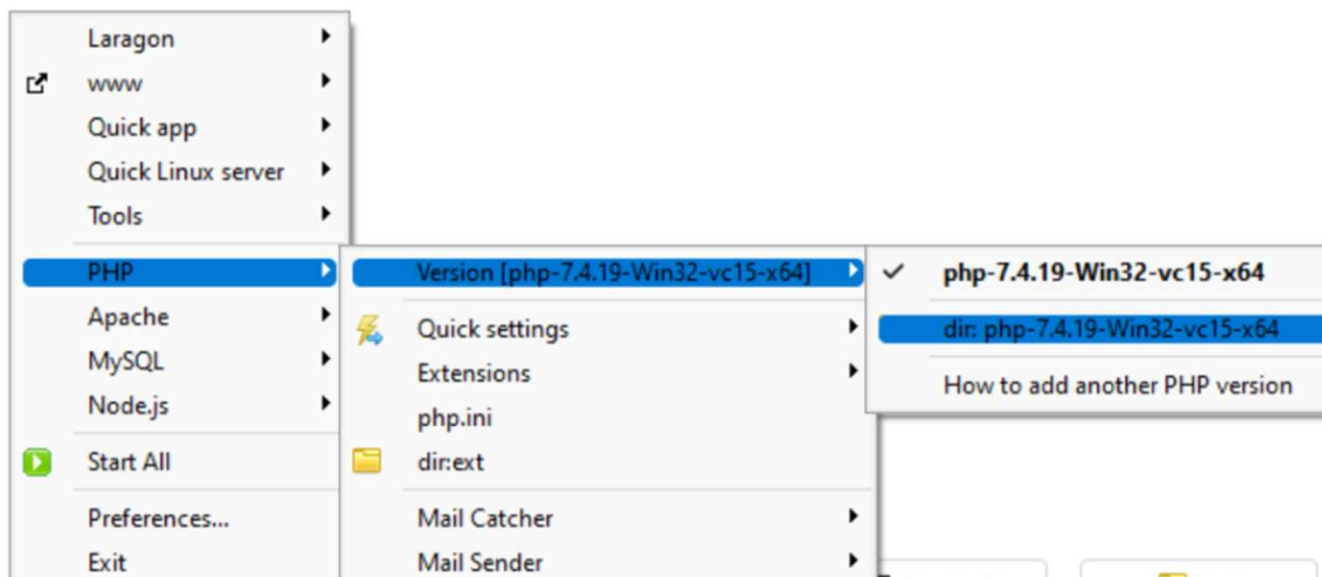
PHP Upgrade

Preparing for Laravel 9

- Once downloaded you can extract the zip to a location of your preference



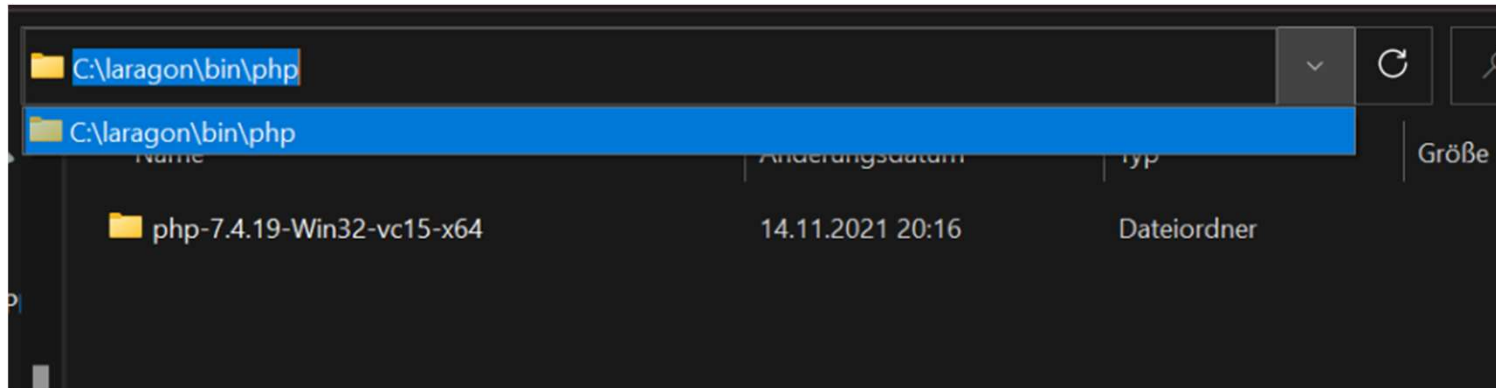
- Right click at the home panel to access your current PHP-Folder



PHP Upgrade

Preparing for Laravel 9

- In your „Laragon\bin\php“ directory you can paste the newly downloaded version of PHP

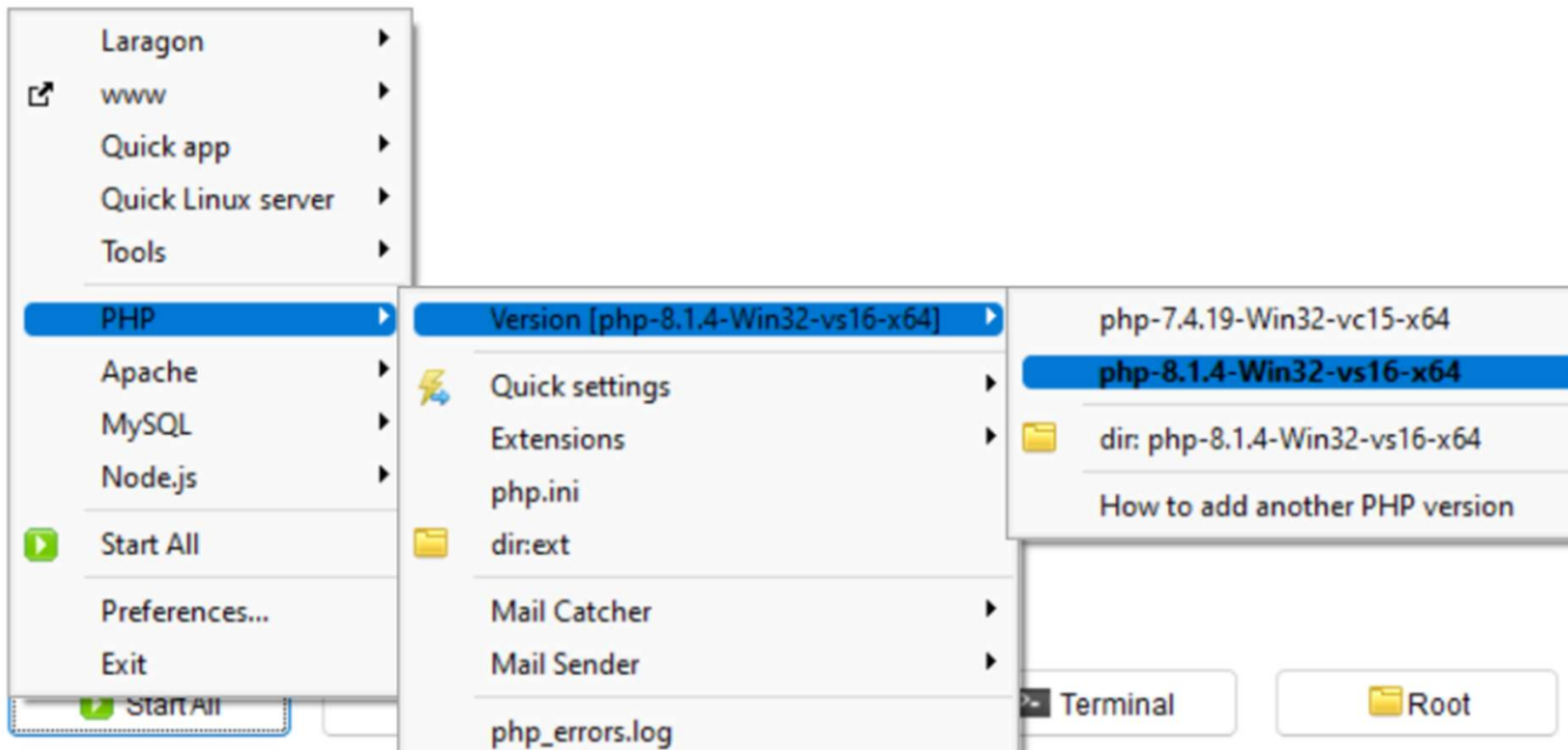


Name	Änderungsdatum	Typ	Größe
php-7.4.19-Win32-vc15-x64	14.11.2021 20:16	Dateiordner	
php-8.1.4-Win32-vs16-x64	19.03.2022 20:07	Dateiordner	

PHP Upgrade

Preparing for Laravel 9

- Now all we need to do is to select the new version at the home panel (right click again)



Laravel Setup

Setup your first Laravel Project

- To create our first project open the console and type in following line to install a project called „HelloLaravel“ (You can use another name)
- The option „prefer-dist“ tries to download and unzip archives of the dependencies using GitHub or another API when available

```
D:\laragon\www>composer create-project --prefer-dist laravel/laravel HelloLaravel
```

- This might take a while ...

```
- Installing nesbot/carbon (2.57.0): Extracting archive
- Installing monolog/monolog (2.4.0): Extracting archive
- Installing league/mime-type-detection (1.9.0): Extracting archive
- Installing league/flysystem (3.0.12): Extracting archive
- Installing nette/utils (v3.2.7): Extracting archive
- Installing nette/schema (v1.2.2): Extracting archive
- Installing dflydev/dot-access-data (v3.0.1): Extracting archive
- Installing league/config (v1.1.1): Extracting archive
- Installing league/commonmark (2.2.3): Extracting archive
- Installing laravel/serializable-closure (v1.1.1): Extracting archive
- Installing laravel/framework (v9.5.1): Extracting archive
- Installing laravel/sail (v1.13.7): Extracting archive
```

Laravel Setup

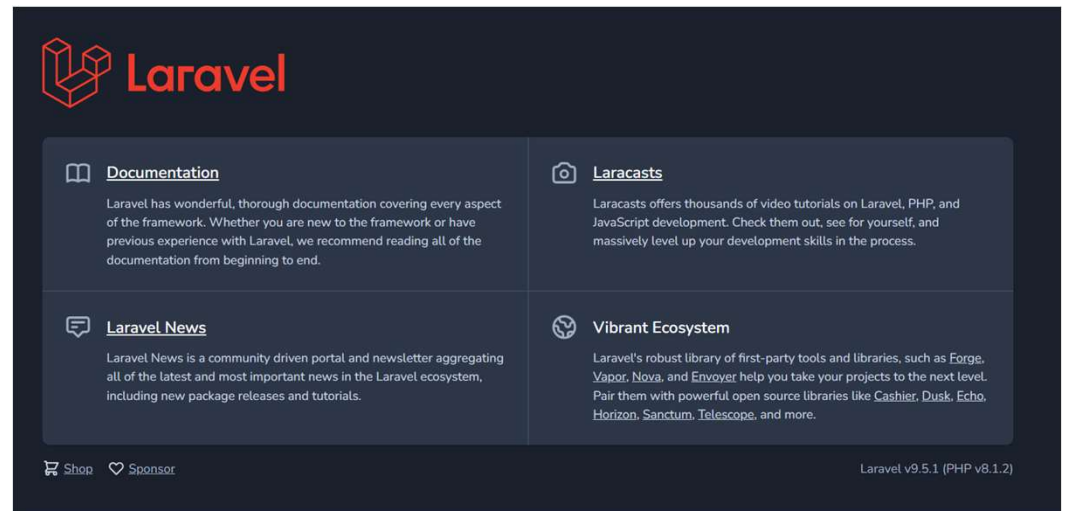
Setup your first Laravel Project

- Change the directory to the new project and start the Development-Server

```
D:\laragon\www>cd HelloLaravel  
  
D:\laragon\www\HelloLaravel>php artisan serve
```

```
Starting Laravel development server: http://127.0.0.1:8000  
[Sat Mar 19 20:29:24 2022] PHP 8.1.2 Development Server (http://127.0.0.1:8000) started
```

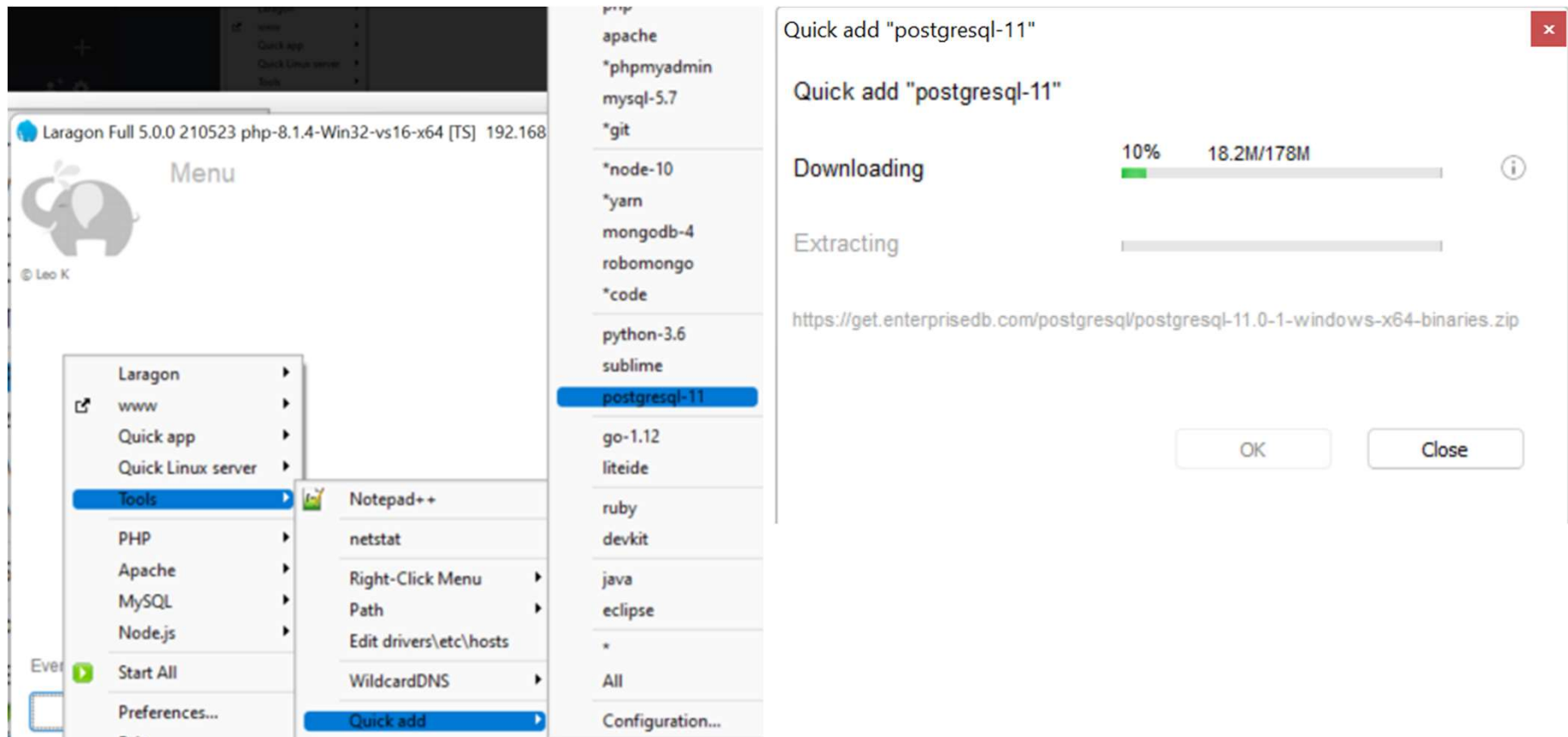
- Access the Server in your browser with <http://127.0.0.1:8000> and your good to go
- You can also stop the Server with Ctrl+C



PostgreSQL

Setup a Database-Connection

- If you choose to use another DBMS like PostgreSQL you can easily set it up with Laragon as well (Tools->Quick add -> postgresql-11)



PostgreSQL

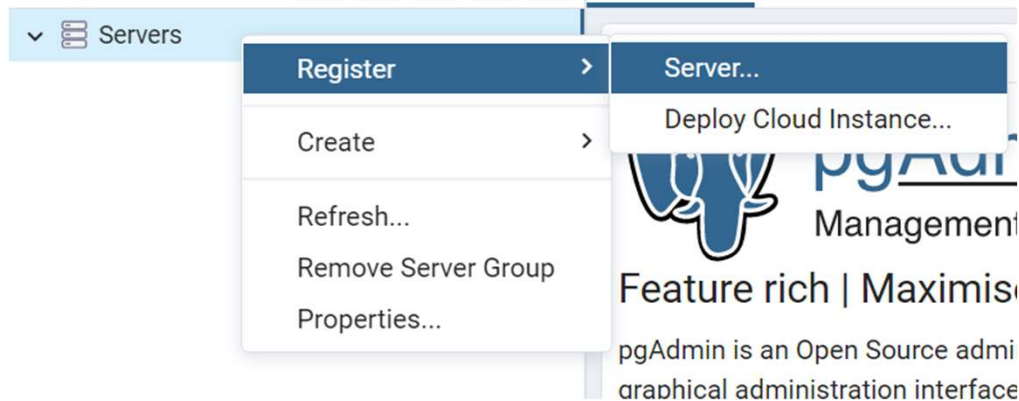
Setup a Database-Connection

- With the options „Start All“ the PostgreSQL Server will be started as well
- We can use Pgadmin to manage our PostgreSQL databases:
<https://www.pgadmin.org/download/pgadmin-4-windows/>
- Next we will create a server on pgadmin so that it can recognize the running instance of PostgreSQL

PostgreSQL

Setup a Pgadmin

- Right click on „Servers“ to register a new Server



- Here you give your server a name and under „Connection“ the hostname/address „localhost“ (the port should be 5432 per default)

PostgreSQL

Setup a Pgadmin

Register - Server

General Connection SSL SSH Tunnel Advanced

Name DBWT

Server group Servers

Background ☐

Foreground ☐

Connect now? ☒

Comments

Register - Server

General Connection SSL SSH Tunnel Advanced

Host name/address localhost

Port 5432

Maintenance database postgres

Username postgres

Kerberos authentication? ☐

Password

Save password? ☐

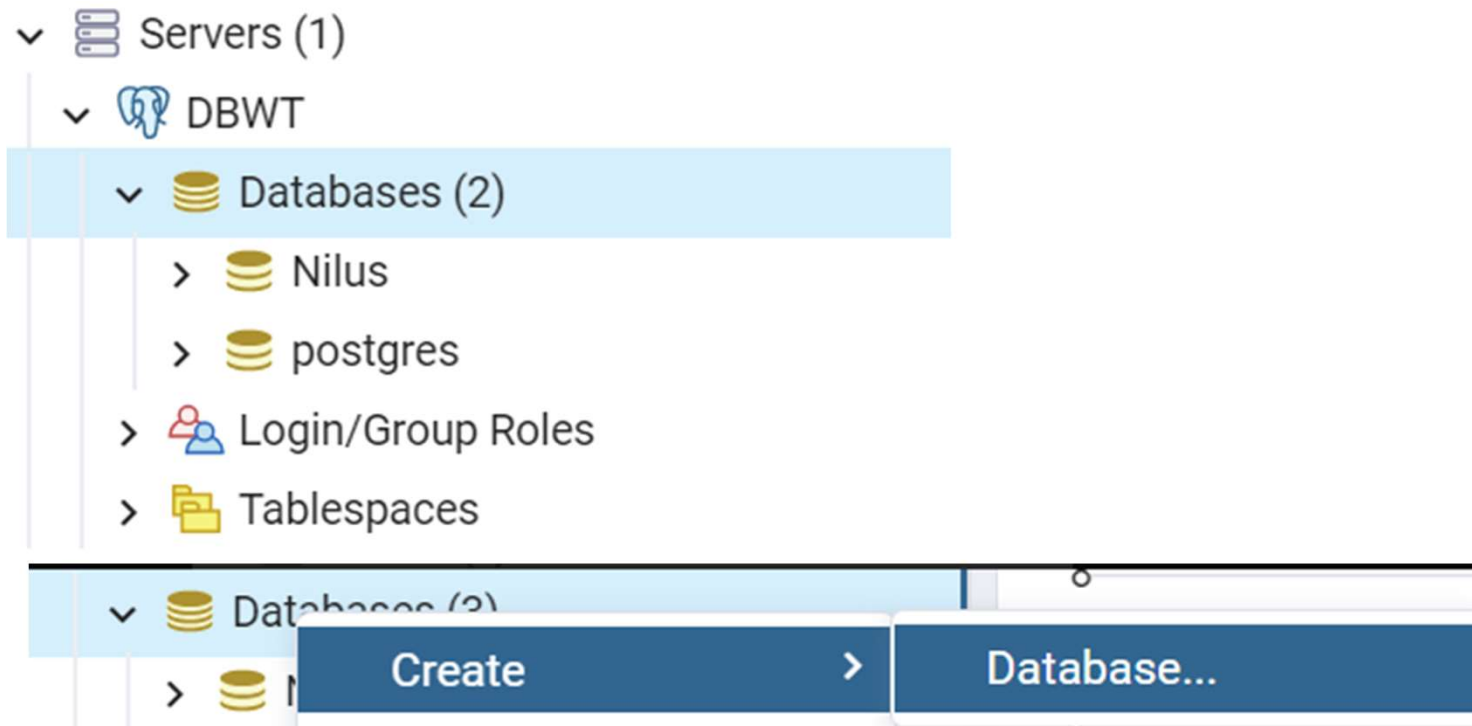
Role

Service

PostgreSQL

Setup a Pgadmin

- Now you can create a database for the laravel project



- I have called my database „hellolaravel“

PostgreSQL

Connecting with the application

- In the .env file of your project you can now setup the database connection
- (if you use VSCode you can open the File Directory in VSCode directly by typing „code .“ in the console)
- In the .env file we will change the settings from mysql to pgsql

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
```



```
DB_CONNECTION=pgsql
DB_HOST=127.0.0.1
DB_PORT=5432
DB_DATABASE=hellolaravel
DB_USERNAME=postgres
DB_PASSWORD=
```

Databasename

Username

PostgreSQL

Connecting with the application

- To test our database connection we can simply run migrations
- Since I will not create any migrations-files i will use a laravel authentication package called „jetstream“ which comes with a user-migration
- To install jetstream run:

```
D:\laragon\www\HelloLaravel>composer require laravel/jetstream
```

- Now lets migrate our user table:

```
D:\laragon\www\HelloLaravel>php artisan migrate
```

PostgreSQL

Connecting with the application

- If the database connection is properly setup the migrations should run flawlessly without error

```
Migration table created successfully.  
Migrating: 2014_10_12_000000_create_users_table  
Migrated: 2014_10_12_000000_create_users_table (100.61ms)  
Migrating: 2014_10_12_100000_create_password_resets_table  
Migrated: 2014_10_12_100000_create_password_resets_table (90.48ms)  
Migrating: 2019_08_19_000000_create_failed_jobs_table  
Migrated: 2019_08_19_000000_create_failed_jobs_table (112.20ms)  
Migrating: 2019_12_14_000001_create_personal_access_tokens_table  
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (142.30ms)
```

- Check pgadmin for a „users_table“

PostgreSQL

Connecting with the application



The table “user” has been successfully migrated and the application connected to the database

NPM

Installing precompiled webpackages

- Node Package Manager manages JavaScript-packages using Node.js
- Laragon also comes with a version of node preinstalled
- Just run „npm install“ in the console to setup the package.json file with npm
- To find out more about compiling assets in laravel with npm have a look at the Laravel-Documentation:
<https://laravel.com/docs/9.x/mix>

- If you right-click the Home Panel and check php->extensions, you will not find redis because it does not come up with Laragon by default, so you have to add it manually.
- Go to <https://pecl.php.net/package/redis> and find your current PHP version and the latest stable available

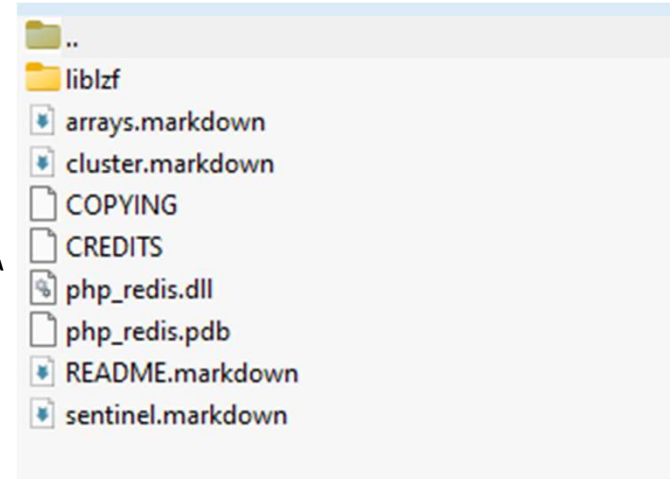
Available Releases				
Version	State	Release Date	Downloads	
5.3.7	stable	2022-02-15	redis-5.3.7.tgz (268.8kB)  DLL	[Changelog]
5.3.7RC2	beta	2022-02-12	redis-5.3.7RC2.tgz (268.8kB)  DLL	[Changelog]
5.3.7RC1	alpha	2022-02-02	redis-5.3.7RC1.tgz (268.8kB)  DLL	[Changelog]
5.3.6	stable	2022-01-17	redis-5.3.6.tgz (267.8kB)  DLL	[Changelog]

- Click the Windows icon DLL to get the available distribution for your PHP version

Redis

Enabling Redis in Laragon

- After the download, you should find the „php_redis.dll“ file
- Copy or move the file to „D:\laragon\bin\php\php-8.1.2-Win32-vs16-x64\ext “ (replace with you own path) and add the line „extension=php_redis.dll to the „D:\laragon\bin\php\php-8.1.2-Win32-vs16-x64\php.ini“ php.ini file

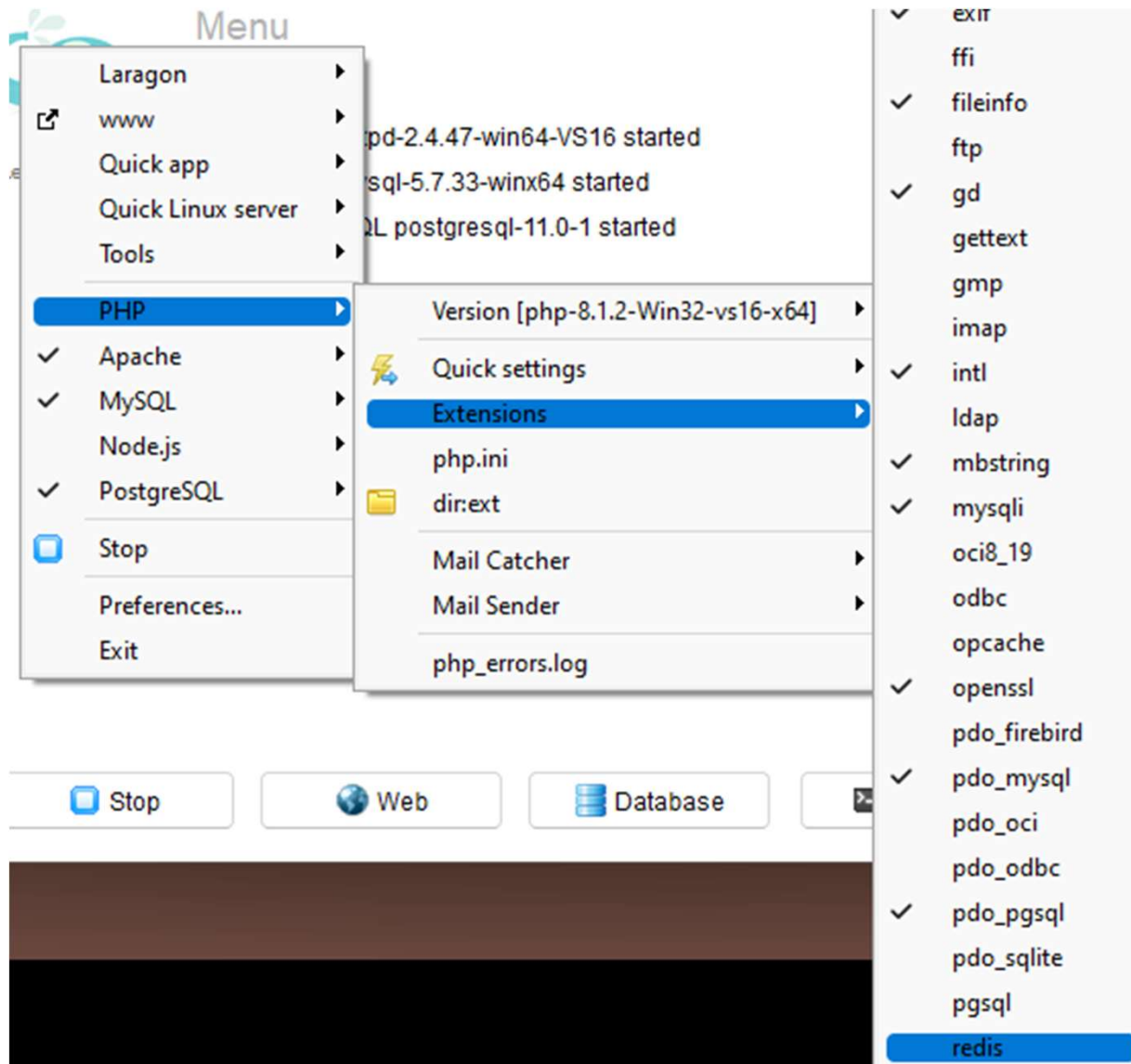


```
;extension=php_redis.dll  
;extension=bz2  
extension=curl
```

- You should see that the Redis extension is available now

Redis

Enabling Redis in Laragon



Redis

Starting Redis in Laragon

- To run the Redis server, you can use the console to execute the `redis-server.exe`

```
D:\laragon\www\HelloLaravel>redis-server.exe
[12724] 19 Mar 22:08:33.169 # Warning: no config file specified, using the default config. In order to
fy a config file use redis-server.exe /path/to/redis.conf

Redis 3.2.100 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 12724

http://redis.io

[12724] 19 Mar 22:08:33.176 # Server started, Redis version 3.2.100
[12724] 19 Mar 22:08:33.176 * The server is now ready to accept connections on port 6379
```

- You can also stop it using `Ctrl+C`

FH Aachen
Fachbereich Elektrotechnik und Informationstechnik

Nilusche Liyanaarachchi (INF)
<https://github.com/Nilusche>

www.fh-aachen.de