

# Advanced Web Programming Setup



**Name**

Dicha Zelianivan Arkana

**NIM**

2241720002

**Class**

2i

**Department**

Information Technology

**Study Program**

D4 Informatics Engineering

---

# 1 Web Server

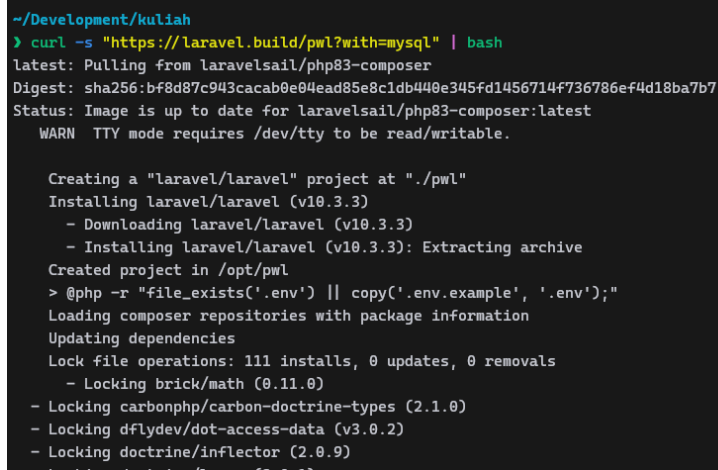
I'm using Linux, so Laragon is not available for me. I've been using Laravel Sail for my development environment when using Laravel this past 2 years or so. It's Docker-based so it can run anywhere and portable. Moreover, it's very easy to use and doesn't require a lot of configuration.

It uses the builtin PHP web server since it's good enough for development. Although it is recommended to use NGINX instead for production.

I installed `sail` using this command, which will basically scaffold the project while setting up the required services such as the database.

```
curl -s "https://laravel.build/my-application-name?with=mysql" | bash
```

Here's an example output, it should start installing the required dependencies and pulling docker images. It will take a while since it's going to download a lot of things.



```
~/Development/kuliah
> curl -s "https://laravel.build/pwl?with=mysql" | bash
latest: Pulling from laravelsail/php83-composer
Digest: sha256:bf8d87c943cacab0e04ead85e8c1db440e345fd1456714f736786ef4d18ba7b7
Status: Image is up to date for laravelsail/php83-composer:latest
WARN TTY mode requires /dev/tty to be read/writable.

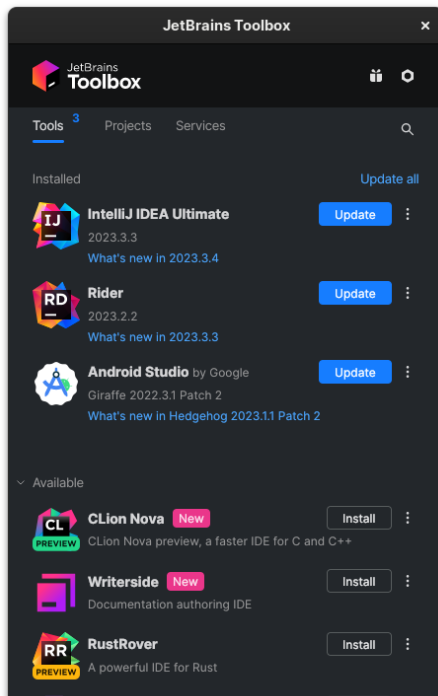
Creating a "laravel/laravel" project at "./pwl"
Installing laravel/laravel (v10.3.3)
- Downloading laravel/laravel (v10.3.3)
- Installing laravel/laravel (v10.3.3): Extracting archive
Created project in /opt/pwl
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.11.0)
- Locking carbonphp/carbon-doctrine-types (2.1.0)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/inferno (2.0.9)
```

The project can still be used with or without Laravel Sail. It's just an option if you don't want to install PHP, Composer, and MySQL on your local machine.

---

## 2 Text Editor / IDE

I use IntelliJ IDEA Ultimate for my main IDE. I already use it for other languages so why bother installing another IDE which is basically just a stripped down version of it. In terms of developer experience, they're the same. Probably some small difference in project template but that's it. Not using it anyway.



## 3 Git

I'm using Git installed from the official Fedora repository, which at the time of writing this, is version 2.43.0 as shown below.

```
~/Development/kuliah/pwl via @ v20.11.0 via v8.2.15  
> git --version  
git version 2.43.0
```

---

## 4 Committing to Github

I'm using Git from the CLI to commit to Github. I've made the repository previously on Github. I'm using the following commands to commit to Github.

```
~/Development/kuliah/pwl via @ v20.11.0 via v8.2.15
> git init
Initialized empty Git repository in /home/elianiva/Development/kuliah/pwl/.git/
pwl on  master (?) via @ v20.11.0 via v8.2.15
> git add .
pwl on  master (+) via @ v20.11.0 via v8.2.15
> git commit -S -m "chore: init"
[master (root-commit) 1e9c17f] chore: init
 80 files changed, 11245 insertions(+)
 create mode 100644 .editorconfig
 create mode 100644 .env.example
 create mode 100644 .gitattributes
 create mode 100644 .gitignore
 create mode 100644 README.md

pwl on  master via @ v20.11.0 via v8.2.15
> git remote add origin git@github.com:elianiva/pwl.git
pwl on  master via @ v20.11.0 via v8.2.15
> git push origin master
Enumerating objects: 103, done.
Counting objects: 100% (103/103), done.
Delta compression using up to 4 threads
Compressing objects: 100% (85/85), done.
Writing objects: 100% (103/103), 72.71 KiB | 4.28 MiB/s, done.
Total 103 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
To github.com:elianiva/pwl.git
 * [new branch]      master -> master
```

It basically initialised the repository, added the files to the staging area, creating a signed commit with a commit message, adding a remote repository, and then pushing it to Github.