Gustavo Gama

PRACTICAL TEST – LARAVEL

A) Instructions to run the code and explanations.

1) Requirements:

Docker and docker compose.

- You can simply download Docker Desktop and run it. Easiest way.

https://www.docker.com/products/docker-desktop/

2) Download this project from the git or clone it.

3) How to run the app?

- Go into the root folder of the project:
  > *cd todo-list-app*

- The .env file contains some configurations
- In the root folder of the Laravel project, there is a script named “setup.sh” , simply run it in the terminal (powershell or other…).

  > *./setup.sh*

This script will build the containers and activate the logs on the console for the app.

IMPORTANT FOR WINDOWS USERS (OR IF YOU CANT RUN BASH SCRIPTS…):
Just copy the docker compose commands from the setup.sh file in the terminal (e.g. powershell) and run them (those cmds work directly on the terminal).

Behind the scenes multiple services are being launched, the app itself, Laravel project (the backend + frontend).

This means that some installations are required (check Dockerfile for more details), specially for the frontend with Vue, such as npm and node which take a considerable time (around 500-600 seconds) during the first run of this script to build the container (the other runs are cached most likely so it’s faster).

Additionnaly, a database is created (Mysql), this also takes some time to setup (around a minute). Check image 1 shows the command execution to start the container, after a waiting message appears.

Please keep track of the logs and look out for the log that says that the app is ready. Check image 2.

```
tilted@tilted-MS-7B45:~/todo_list_laravel/todo-list-app$ ./setup.sh
Starting the containers, the build might take a while (around 500seconds +- , specially for the first time due to packages installation such as npm and node for the app frontend)... \n
[+] Building 18.5s (15/15) FINISHED
 => [app internal] load build definition from Dockerfile
 => => transferring dockerfile: 946B
 => [app internal] load .dockerignore
 => => transferring context: 2B
 => [app internal] load metadata for docker.io/library/composer:latest
 => [app internal] load metadata for docker.io/library/php:8.1-fpm
 => [app stage-0 1/8] FROM docker.io/library/php:8.1-fpm@sha256:bde482789a6d1fea323e3d29c5f2e01c19f0d84c7eb897661e991ce8b8757b9e
 => [app internal] load build context
 => => transferring context: 1.13MB
 => [app] FROM docker.io/library/composer:latest@sha256:7c03aa544494973299998db9e51f3c4ca880f2d51475b78a4bca4214a8a4fe82
 => CACHED [app stage-0 2/8] RUN apt-get update && apt-get upgrade -y     && apt-get install -y libpng-dev libjpeg-dev libfreetype6-dev zip git unzip curl nodejs npm default-mysql-client     && do
 => CACHED [app stage-0 3/8] WORKDIR /var/www
 => CACHED [app stage-0 4/8] COPY --from=composer:latest /usr/bin/composer /usr/bin/composer
 => [app stage-0 5/8] COPY . .
 => [app stage-0 6/8] RUN composer install
 => [app stage-0 7/8] RUN npm install
 => [app stage-0 8/8] RUN chmod +x /var/www/entrypoint.sh
 => [app] exporting to image
 => => exporting layers
 => => writing image sha256:67ceaea3e34ec3d442db8416f113281f69fedc16fc49877dd56b27761780e167
 => => naming to docker.io/library/todo-list-app-app
[+] Running 4/4
 ✔ Network todo-list-app_default      Created
 ✔ Container todo-list-app-mysql-1    Started
 ✔ Container todo-list-app-adminer-1  Started
 ✔ Container todo-list-app-app-1      Started
todo-list-app-app-1  | Waiting for the Database to Start (it might take a minute)...
```

Image 1. launch the script command .

```
todo-list-app-app-1  | MySQL DB is up - executing commands (migrations)
todo-list-app-app-1  |
todo-list-app-app-1  |
todo-list-app-app-1  |    INFO  Preparing database.
todo-list-app-app-1  |
todo-list-app-app-1  |
todo-list-app-app-1  |   Creating migration table ..................................... 248ms DONE
todo-list-app-app-1  |
todo-list-app-app-1  |
todo-list-app-app-1  |    INFO  Running migrations.
todo-list-app-app-1  |
todo-list-app-app-1  |
todo-list-app-app-1  |   2014_10_12_000000_create_users_table ......................... 401ms DONE
todo-list-app-app-1  |   2014_10_12_100000_create_password_reset_tokens_table ......... 438ms DONE
todo-list-app-app-1  |   2019_08_19_000000_create_failed_jobs_table ................... 321ms DONE
todo-list-app-app-1  |   2019_12_14_000001_create_personal_access_tokens_table ....... 555ms DONE
todo-list-app-app-1  |   2023_08_27_203909_create_tasks_table ......................... 123ms DONE
todo-list-app-app-1  |
todo-list-app-app-1  |
todo-list-app-app-1  | [29-Aug-2023 19:44:01] NOTICE: fpm is running, pid 92
todo-list-app-app-1  | [29-Aug-2023 19:44:01] NOTICE: ready to handle connections
todo-list-app-app-1  |
todo-list-app-app-1  |
todo-list-app-app-1  |    INFO  Server running on [http://0.0.0.0:8000].
todo-list-app-app-1  |
todo-list-app-app-1  |
todo-list-app-app-1  |   Press Ctrl+C to stop the server
todo-list-app-app-1  |
todo-list-app-app-1  |
todo-list-app-app-1  |   2023-08-29 19:44:05                                       ~ ls
```

Image 2: Logs that show that migrations are ready and the app is ready to go.

4) Results

In http://localhost:8000/ you can find the TODO List app (frontend). The UI is pretty basic (Image 3) and straightforward, you can add a new task (a form will appear), but also edit, and delete. The list is updated instantly based on the seleted operation.
It shows the title and the status (completed tasks are in green).

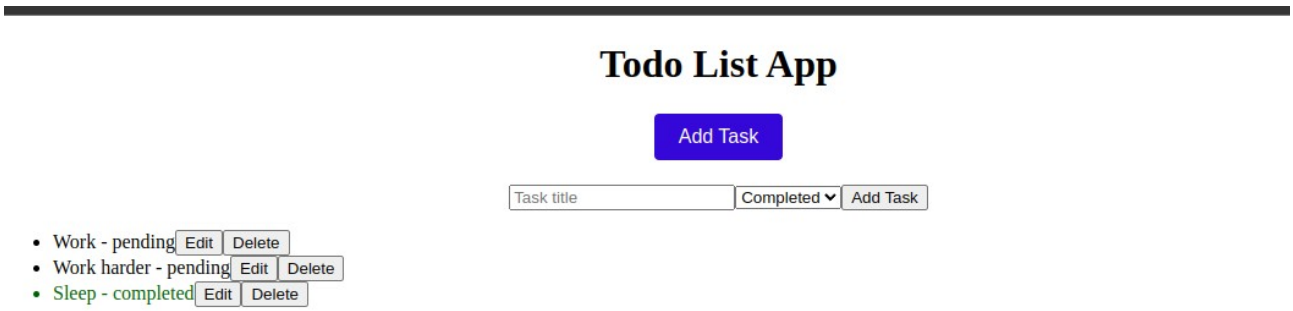

Image 3: Frontend TODO List App.

Also, there is the adminer page (bonus) which allows to check the database, in the url: http://localhost:8080/ , enter the config then login.

System: MySQL;
Server: mysql;
Username: root
Password: root

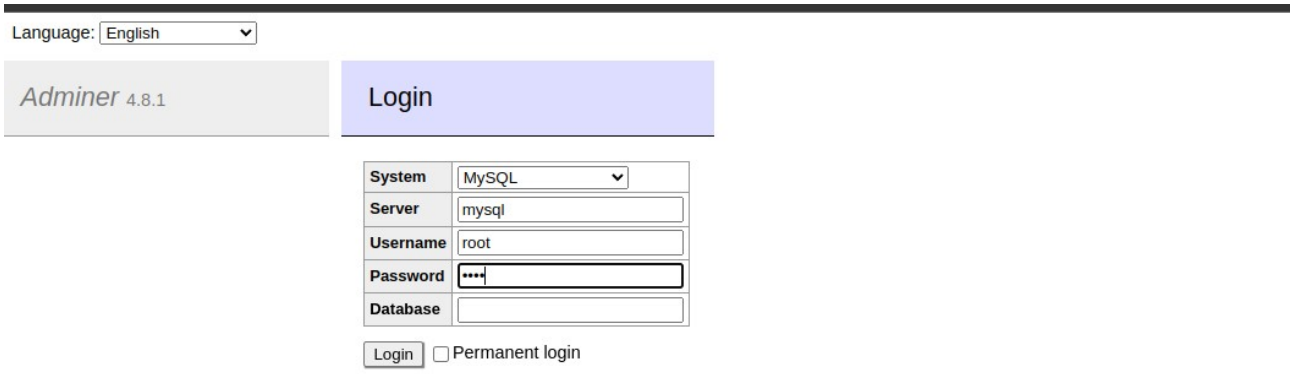

Image 4: Adminer page (check database tables). Password: "root"

You can check the tables and the database (Tasks model) if you follow the path
Database: laravel » Table: tasks. Like in the image 5.

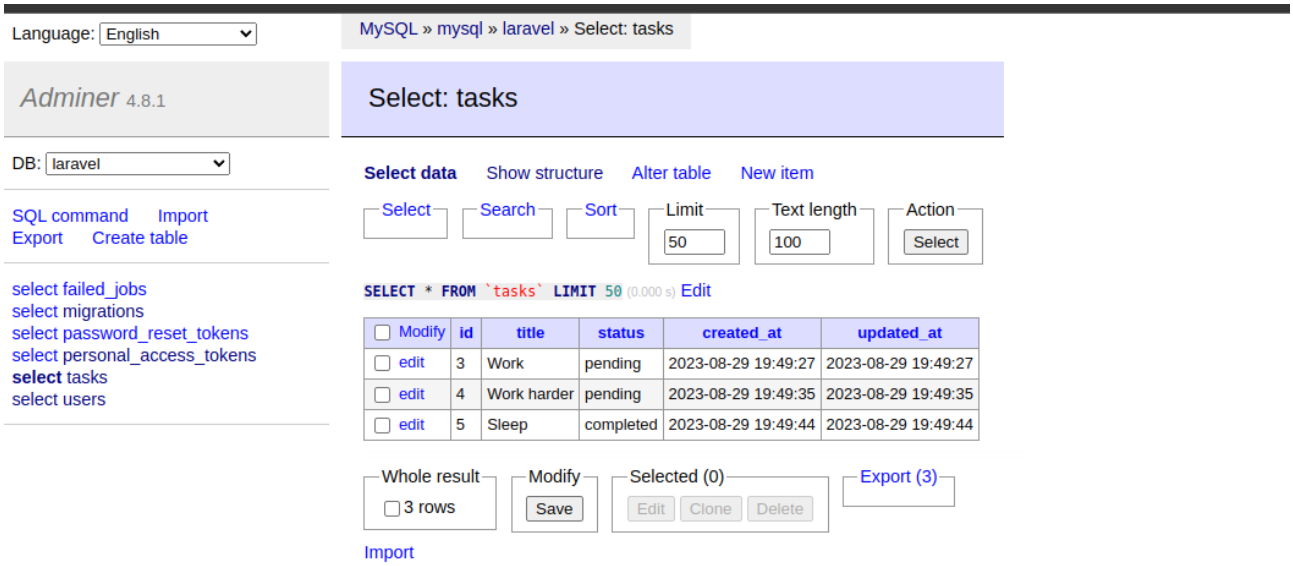

Image 5: Database. Table Tasks data.

5) Tests

After your containers are up and running, you can manually execute the tests inside the app container. Open a new terminal and run:

> docker compose exec app php artisan test

Example in the image 6.

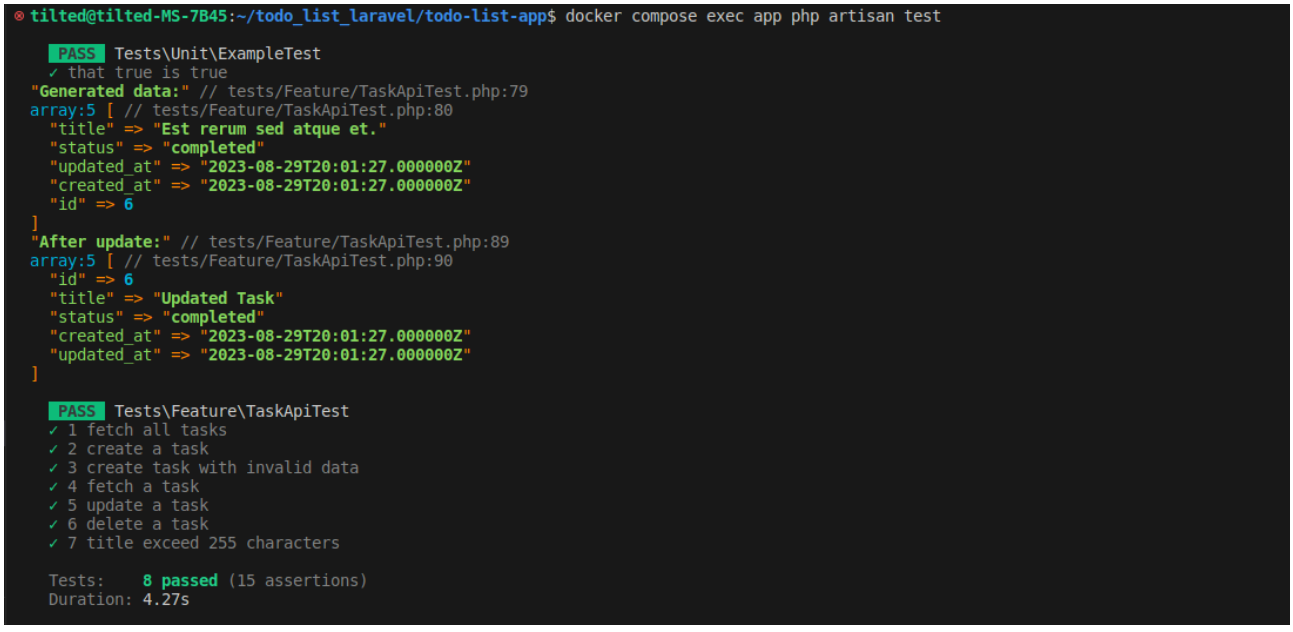

Image 6: Tests execution