

Documents

- 1) How to run project:
Configuration Requirements:
PHP >=7.4

We need to run:

composer install

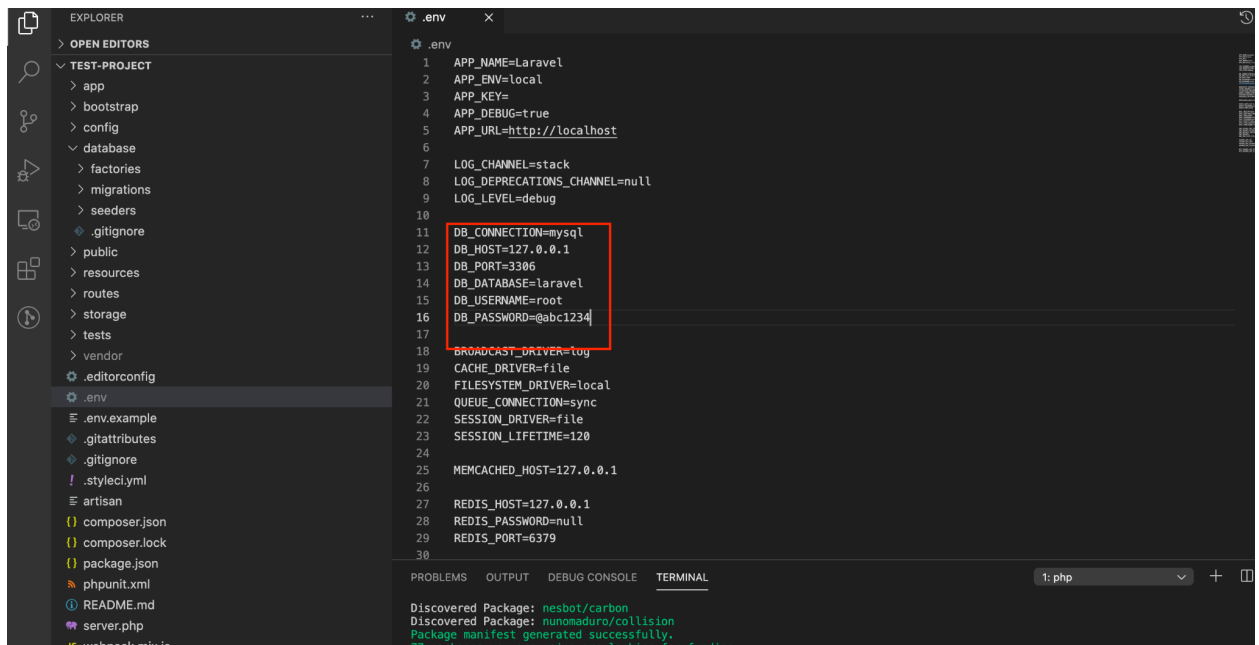
Copy ***.env.example*** file to ***.env*** file

Run ***php artisan jwt:secret***

This will update your ***.env*** file with something like ***JWT_SECRET=foobar***

Create a new Mysql database.

Then configure database in ***.env file***



The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar displays the project structure, with the `.env` file selected. The main editor area shows the contents of the `.env` file. The database configuration section is highlighted with a red box, showing the following lines:

```
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=laravel
15 DB_USERNAME=root
16 DB_PASSWORD=abc1234
```

Below the database configuration, other environment variables are visible, including `BROADCAST_DRIVER=log`, `CACHE_DRIVER=file`, `FILESYSTEM_DRIVER=local`, `QUEUE_CONNECTION=sync`, `SESSION_DRIVER=file`, `SESSION_LIFETIME=120`, `MEMCACHED_HOST=127.0.0.1`, `REDIS_HOST=127.0.0.1`, `REDIS_PASSWORD=null`, and `REDIS_PORT=6379`. At the bottom, the TERMINAL panel shows output from a command, indicating that package manifests were generated successfully.

Generate application key:

php artisan key:generate

Run migration and seeder:

php artisan migrate

php artisan db:seed

So now you can run Unit test:

php artisan test

Run code in Testing server:

php artisan serve

Note: when call any API, you should add to Header: **Accept: application/json**

The screenshot shows the Postman interface. On the left, there's a sidebar with 'My Workspace' and a collection named 'My first collection'. The main area displays a POST request to 'http://127.0.0.1:8000/api/register'. The 'Headers' tab is selected, showing a table with one header: 'Accept' with value 'application/json'. A red circle highlights this header. Below the headers, the 'Body' tab is selected, showing a JSON response in 'Pretty' format. The response is: { "message": "The given data was invalid.", "errors": { "email": ["The email has already been taken."] } }. The status bar at the bottom indicates 'Status: 422 Unprocessable Content', 'Time: 41 ms', and 'Size: 425 B'.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Accept	application/json	
Key	Value	Description

```
1 {
2   "message": "The given data was invalid.",
3   "errors": {
4     "email": [
5       "The email has already been taken."
6     ]
7   }
8 }
```

2) Questions

a) Register API



109092	Bajo Sate Meriah	Edit	Delete
--------	------------------	------	--------

Email

Password

Login

Note :

CRUD can only happen if you have logged in.

1. Register User

Method : Post
Url : hoodwink.medkomtek.net/api/register
Body : formdata
email : john.doe@gmail.com
Password : T3stKl1kd0kt3r

2. Login

Method : Post
Url : hoodwink.medkomtek.net/api/auth/login

In my opinion, I have to add **re-password** field to Body Request
But I still implemented it as your requirement.

b) Update Product API

```
Body : formdata
Sku : OBT-001
Product_name : OBT -Sehat01
Qty : 100
Price : 100000
unit: Carton
Status : 1
```

4. Update Products

```
Method Post
Url hoodwink.medkomtek.net/api/item/update
Header
Authorization : Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJpZ
WFyZXIiLCJzdz
WlI0IjIsImhhdCI6MTYwNTc2NzEyOCwiZXhwIjoxNjA1N
zcwNzI4fQ.wytEMA
eAzUUeZDGHx5opFwaf97r2slzzBTdvHqp6__l
Body formdata
Sku : OBT-001
Product_name : OBT -Sehat01
Qty : 100
Price : 100000
unit: Carton
Status : 1
```

5. Delete Products

```
Method : Post
Url hoodwink.medkomtek.net/api/item/add
Header
Authorization : Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJpZ
WFyZXIiLCJzdz
WlI0IjIsImhhdCI6MTYwNTc2NzEvOCwiZXhwIjoxNjA1Nz
```

This api I'm not quite clear about. It seems you want to update all fields of table. (I still implemented it as you want).

It should be able to update some pieces of fields (or all fields) **with PUT method**.

c) Get product list by SKU

With product list by sku -> I will use “%LIKE%” to query database, and return data array (As you want)

But I think Sku is unique. So we just search with a sku and return an Object (use “sku = 134HAAD” in query)