

Erply

March 28, 2019

Documentation

Erply Product API Assignment

Requirements

Write a small API that uses a database backend.

The main data entity in your API is a "product". A product must have the following fields/properties:

- ID (integer, assigned automatically)
 - Name (alphanumeric, at most 100 characters)
 - Price (decimal, with 2 decimal places)
1. Implement API calls for creating, updating and deleting these records.
 2. Implement API calls for retrieving a list of products, and for searching a product by name.
 3. Ensure that the API calls have proper error handling, and output sensible error messages.
 4. Document the API: explain the proper way to use it. Assume that the reader is not familiar with the API, and will be using it as a service (having no access to the source code).

Write the application in PHP. Use a MySQL database.

Please submit:

- API source code (including a README file with deployment instructions);
- SQL schema description;

- Documentation.

Getting Started

These instructions will get you a copy of the project up and running on your local machine for development and testing purposes. However this project was created using docker, you can install as a regular Laravel project in any server that support php/mysql

Requirements

```
PHP      7.1.27

Mysql    5.6

Docker   18.09.2 (build 6247962) (optional if using docker)

libmcrypt-dev mysql-client libmagickwand-dev git

php ext: imagick mcrypt pdo_mysql mbstring zip php-zip
```

If you will use docker run this instructions

```
cd product

docker-compose build

docker-compose up
```

Up and running with docker

In order to run the project using docker, after install git and docker you will need

1. clone this repository.

2. Build the app.
3. Start and runs the entire app.

```
git clone https://github.com/elminson/apiProject.git
```

```
cd apiProject
```

```
docker-compose build
```

```
docker-compose up
```

Now you can enter in the docker container

```
docker exec -it apiproject_app_1 bash
```

and run this

```
composer install
```

```
composer update
```

```
php artisan config:clear
```

```
php artisan config:cache
```

```
composer dump-autoload
```

```
php artisan key:generate
```

```
php artisan migrate
```

```
php artisan db:seed
```

The app will be accessible by the port 8080

```
http://127.0.0.1:8080/
```

In order to access the api you will need register in

```
http://127.0.0.1:8080/register
```

Install the project

(replace ***http://127.0.0.1:8080/*** with your server ip and port)

SQL schema **db_schema.sql**

The SQL Schema is included with the project, however this SQL Schema is loaded by running
`php artisan migrate`

Config file **.env**

If you are using docker not modification will be needed inside of .env file. If the project is loaded in a regular server (not execute by docker) you can modify the database configuration / APP name / APP Pagination / etc.

Install (no using docker)

In the folder ./product you should run

```
apt install php-dom  
  
composer install  
  
composer update  
  
php artisan config:clear  
  
php artisan config:cache  
  
composer dump-autoload  
  
php artisan key:generate  
  
php artisan migrate  
  
php artisan db:seed
```

Running PHPUnit (total Test Cases Created) (7 tests, 67 assertions) All test cases passed OK

You will need to replace the ip for your actual ip (domain name) from line 14

file **tests/Unit/ProductTest.php**

```
protected $server_ip = '172.20.0.4';
```

Also replace the line 15 with the full uri of the project

Ex.

```
protected $uri = 'product/public/api/products';
```

run inside *product* folder

```
php vendor/bin/phpunit
```

After register (<http://127.0.0.1:8080/register>) you can access your **API Token** from your dashboard

If you generate a new **API Token** this will invalidate the previous one.

Documentation

The documentation for the api use is available in

```
http://localhost:8080/docs/1.0
```

The documentation was created using SnowBoard project

```
$snowboard html -o resources/views/docs/index.blade.php  
apiDocs_snowboard.apib
```

<https://github.com/bukalapak/snowboard>

<https://apibluetooth.org/developers.html>

Errors or 500

If you receive any of this error please follow this steps

- First of all check your file permissions.
- To fix file permissions `sudo chmod 755 -R your_project`
- Then `chmod -R o+w your_project/storage` to write file to storage folder.
- `php artisan cache:clear`
- `composer dump-autoload`
- `php artisan key:generate`
- Then check server requirements as per the laravel requirement.
- Many times you got this error because of the php version. Try changing your php version in cpanel.
- Then configure your `.htaccess` file properly
- Check the properly user and group are assigned to the folder

If you receive and error (Uncaught ReflectionException: Class config does not exist in)

Mod_rewrite is required (more info)

https://www.digitalocean.com/community/tutorials/how-to-rewrite-urls-with-mod_rewrite-for-apache-on-ubuntu-18-04

Built With

- [Laravel](#) - The PHP framework used
- [Docker](#) - Used for containerization
- [Bukalapak/snowboard](#) - Documentation Generation
- [Api blueprint](#) - Documentation API Blueprint

Tested

Url: <http://157.230.233.149/erply/product/public/home>

Documentation: <http://157.230.233.149/erply/product/public/docs/1.0>

The project was tested in a new droplet in Digital Ocean
(<https://digitalocean.com>) following this document.

Droplet used Ubuntu Docker

1 GB Memory / 25 GB Disk / NYC1 - Ubuntu 18.04.2 x64



Erply

Web Application / Erply Assignment

Resources

Activity

Settings

DROPLETS (1)

●  **erply-ubuntu-s-1vcpu-1gb-nyc1-01** 157.230.233.149



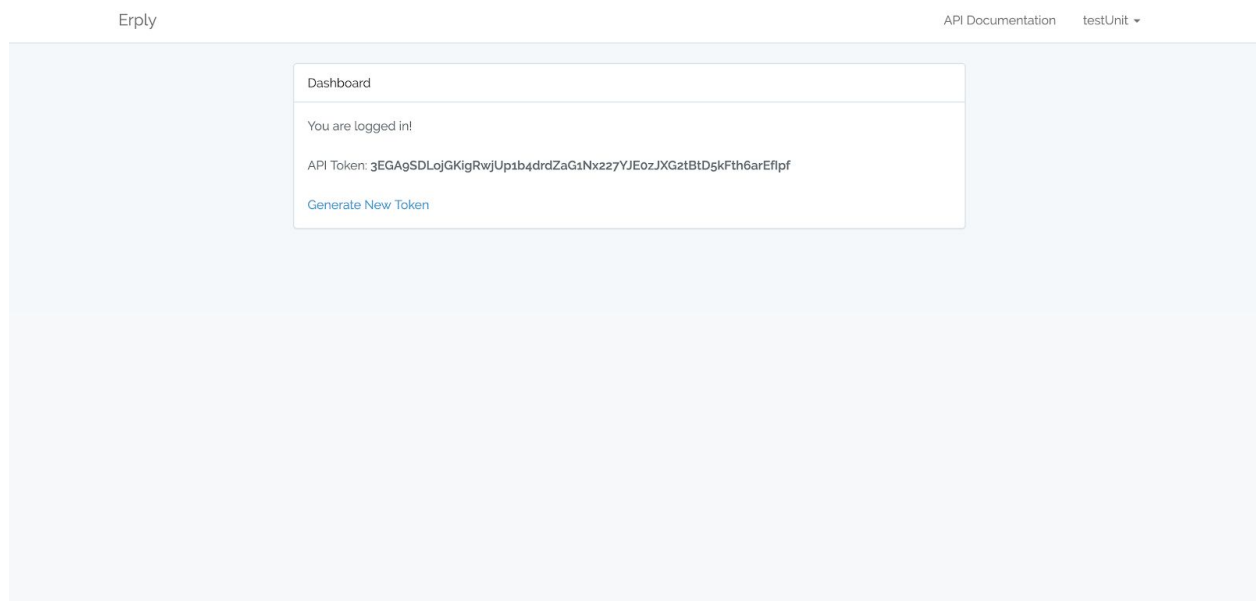
erply-ubuntu-s-1vcpu-1gb-nyc1-01

in  Erply / 1 GB Memory / 25 GB Disk / NYC1 - Ubuntu 18.04.2 x64

Screenshots

API DOCUMENTATION HOME

Erply Product API



Erply [API Documentation](#) [Login](#) [Register](#)

Login

E-Mail Address

Password

☐ Remember Me

[Login](#) [Forgot Your Password?](#)

Erply [API Documentation](#) [Login](#) [Register](#)

Register

Name

E-Mail Address

Password

Confirm Password

[Register](#)

Erply Product API

INTRODUCTION

Products API

▼ Products

List of Products

Retrieve Product

Retrieve Product (not found response)

Delete Product

Delete Product (not found response)

Create Product

Update a Product

Update a Product (not found response)

Search Products by name

Search Products by name (not found response)

Search Products by exact name

Search Products by exact name (not found response)

Complete request example:

Products API

Assignment by Erply Inc.

Products

In order to get an **API Token** you have to [Login](#) and Getting from your dashboard or you can generate a new **API Token** invalidating the previews one.

List of Products

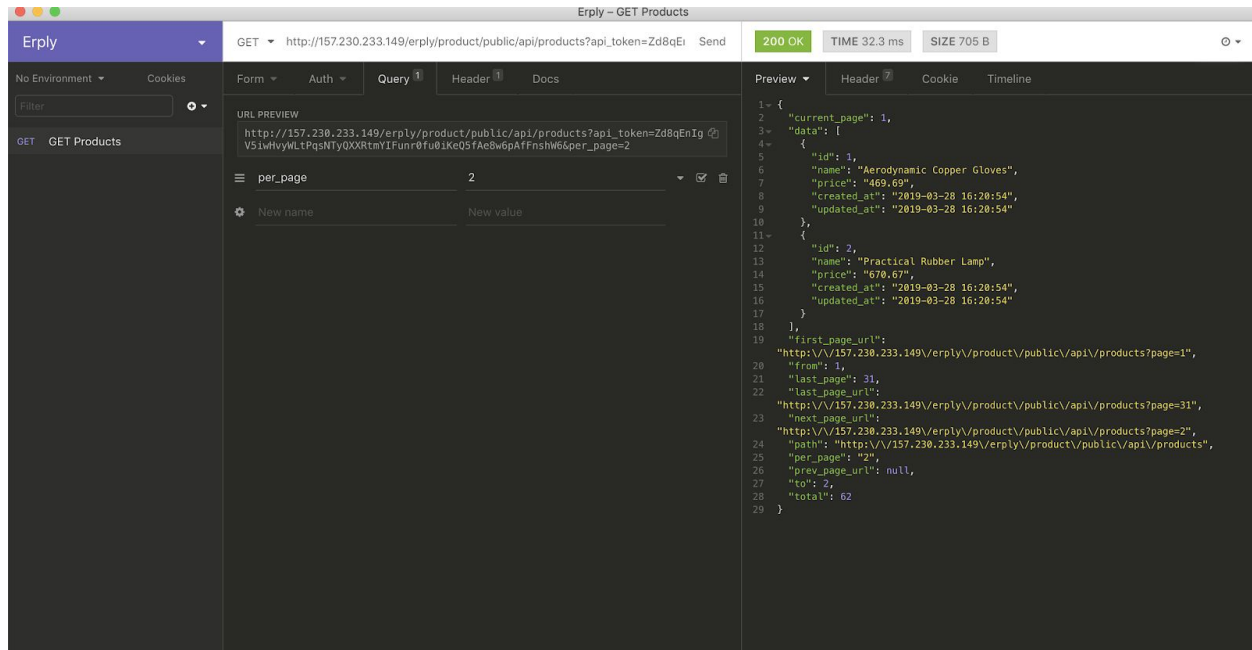
REQUEST

GET `/api/products/{?api_token,page,per_page}`

Parameters

●	page	number	1	-
●	per_page	number	5	Number of item per page (default 10) [Optional]
		string		

Request GET [products] API Screenshots



The screenshot shows a web browser window with the address bar displaying the URL: `http://157.230.233.149/erply/product/public/api/products?api_token=Zd8qEnIgV51wHyvMLtPqkTyQXXRtm1IFunr0fu01kcQ5fAe0w6pAffnshW66per_page=2`. The browser status bar indicates a 200 OK response, a time of 32.3 ms, and a size of 705 B. The main content area displays the JSON response:

```

1- {
2-   "current_page": 1,
3-   "data": [
4-     {
5-       "id": 1,
6-       "name": "Aerodynamic Copper Gloves",
7-       "price": "469.69",
8-       "created_at": "2019-03-28 16:28:54",
9-       "updated_at": "2019-03-28 16:28:54"
10-    },
11-    {
12-      "id": 2,
13-      "name": "Practical Rubber Lamp",
14-      "price": "670.67",
15-      "created_at": "2019-03-28 16:28:54",
16-      "updated_at": "2019-03-28 16:28:54"
17-    }
18-  ],
19-   "first_page_url": "http://157.230.233.149/erply/product/public/api/products?page=1",
20-   "from": 1,
21-   "last_page": 31,
22-   "last_page_url": "http://157.230.233.149/erply/product/public/api/products?page=31",
23-   "next_page_url": "http://157.230.233.149/erply/product/public/api/products?page=2",
24-   "path": "http://157.230.233.149/erply/product/public/api/products",
25-   "per_page": "2",
26-   "prev_page_url": null,
27-   "to": 2,
28-   "total": 62
29- }

```

PHPUnit result

```

root@erply-ubuntu-s-1vcpu-1gb-nyc1-01:/var/www/html/erply/product# php vendor/bin/phpunit
PHPUnit 6.5.14 by Sebastian Bergmann and contributors.

```

```

.....

```

```

7 / 7 (100%)

```

```

Time: 521 ms, Memory: 16.00MB

```

```

OK (7 tests, 67 assertions)

```

```

root@erply-ubuntu-s-1vcpu-1gb-nyc1-01:/var/www/html/erply/product#

```