MongoDB and Laravel with Docker

Docker Compose

```yaml
version: '3'

services:
    nginx:
        build:
            context: .
            dockerfile: nginx.dockerfile
        ports:
            - 80:80
        volumes:
            - ./:/var/www/html
        depends_on:
            - php
            - mongo
        networks:
            - app-network

    php:
        build:
            context: .
            dockerfile: php.dockerfile
        volumes:
            - ./:/var/www/html
        networks:
            - app-network

    mongo-express:
        image: mongo-express
        ports:
            - 8081:8081
        environment:
            ME_CONFIG_BASICAUTH_USERNAME: gustavo
            ME_CONFIG_BASICAUTH_PASSWORD: gustavo
            ME_CONFIG_MONGODB_PORT: 27017
            ME_CONFIG_MONGODB_ADMINUSERNAME: root
            ME_CONFIG_MONGODB_ADMINPASSWORD: gustavo
        links:
```

```yaml
        - mongo
    networks:
        - app-network

  mongo:
    image: mongo
    environment:
        MONGO_INITDB_ROOT_USERNAME: root
        MONGO_INITDB_ROOT_PASSWORD: gustavo
    ports:
        - "27017:27017"
    volumes:
        - .docker/dbdata:/data/db
    networks:
        - app-network

networks:
  app-network:
    driver: bridge
```

Docker file PHP

```dockerfile
FROM php:7.4-fpm-alpine

# ADD ./php/www.conf /usr/local/etc/php-fpm.d/www.conf

RUN addgroup -g 1000 wp && adduser -G wp -g wp -s /bin/sh -D wp

RUN mkdir -p /var/www/html

RUN chown wp:wp /var/www/html

WORKDIR /var/www/html

RUN docker-php-ext-install mysqli pdo pdo_mysql && docker-php-ext-enable pdo_mysql \
   && apk --update add --virtual build-dependencies build-base openssl-dev autoconf \
 && pecl install mongodb \
 && docker-php-ext-enable mongodb \
 && apk del build-dependencies build-base openssl-dev autoconf \
```

```
    && rm -rf /var/cache/apk/*
```

Docker file NGINX

```dockerfile
FROM nginx:stable-alpine
ADD ./nginx/default.conf /etc/nginx/conf.d/default.conf
```

NGINX default.conf

```nginx
upstream php {
    server unix:/tmp/php-cgi.socket;
    server php:9000;
}

server {
    listen 80;
    server_name wordpress-docker.test;

    root /var/www/html/public;

    index index.php;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        include fastcgi.conf;
        fastcgi_intercept_errors on;
        fastcgi_pass php;
    }

    location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
        expires max;
        log_not_found off;
    }
}
```

Laravel config/database.php

```php
'mongodb' => [
            'driver'   => 'mongodb',
```

```
        'host'     => env('MONGO_DB_HOST'),
        'port'     => env('MONGO_DB_PORT', 27017),
        'database' => env('MONGO_DB_DATABASE'),
        'username' => env('MONGO_DB_USERNAME'),
        'password' => env('MONGO_DB_PASSWORD'),
        'options'  => []
    ],
```

Laravel app providers

```
Jenssegers\Mongodb\MongodbServiceProvider::class,
```

Laravel Model Example

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Jenssegers\Mongodb\Eloquent\Model as Model;

class Post extends Model
{
    use HasFactory;
    protected $connection = 'mongodb';
    protected $collection = 'posts';
    protected $fillable = [
        'title','content'
    ];
}
```

Disable auth on API routes
Script: app/Http/Middleware/VerifyCsrfToken.php
Add the route in the except array

```php
protected $except = [
        'posts/'
    ];
```

Docker commands

```
docker-compose up -d --build
docker exec -it mongolaravel_php_1 sh
apk update
```

```
apk add composer
composer require jenssegers/mongodb
docker exec -it mongolaravel_nginx_1 sh
chmod -R 777 /var/www/html/storage/
```

Result GET