

**OBJETIVO:**

# **Hacer una API RESTful en Lumen tomando componentes de la API RESTful que fue hecha en Laravel**

## **Contenido**

Preparar entorno.....	2
Generar y configurar Api-key.....	2
Copiar archivos de modelos.....	2
Copiar archivos de factories y seeders.....	2
Copiar archivos de migraciones.....	3
Habilitar Eloquent.....	3
Ejecutar migraciones y limpiar.....	3
Copiar los controladores.....	3
Cambiar método validate en controladores.....	3
Modificaar controladores para llamada de clase Lang.....	3
Verificar que nombres de los métodos de los controladores coinciden con lo que está en el ruteador.....	4
Habilitar el multilenguaje.....	4
Copiar middleware de localización.....	4
Registrar el middleware.....	4
Copiar carpetas de lenguaje.....	4
Agregar variable por defecto al archivo .env.....	4
Configurar el idioma por defecto.....	4
Agregar rutas de los controladores.....	5
Testear la api.....	5
Documentación pública de la API.....	7
Referencias web adicionales.....	7



## Preparar entorno

Información en <https://github.com/riv4wi/document-and-notes/blob/master/nuevo-proyecto-con-laravel-composer-apache.pdf>

## Generar y configurar Api-key

Puesto que lumen viene con lo minimo necesario para trabajar, a diferencia de Laravel, hay que hacer esta pequeña operación manualmente

```
alias rands='cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w ${1:-32} | head -n 1'
```

Tengo este alias en el bash, pero el objetivo es crear una clave y copiarla en el archivo de configuración .env.

## Configurar la base de datos en archivo .env

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=lar_apirestful  
DB_USERNAME=riv4wi  
DB_PASSWORD=1234
```

## Copiar archivos de modelos

```
cd /var/www/laravel/apirf.lumen/  
cp -v /var/www/laravel/apirf.lar/app/Vehicle.php ./app/  
cp -v /var/www/laravel/apirf.lar/app/Manufacturer.php ./app/
```

Se copió el contenido del modelo User de **apirf.lar** en **apirf.lumen**

## Copiar archivos de factories y seeders

```
cp -v /var/www/laravel/apirf.lar/database/factories/* ./database/factories/  
cp -v /var/www/laravel/apirf.lar/database/seeds/* ./database/seeds/  
Cambiar en los seeders y en los factories el bcrypt por el hash en la generación de usuarios  
/var/www/laravel/apirf.lumen/database/seeds/UserTableSeeder.php  
/var/www/laravel/apirf.lumen/database/factories/ModelFactory.php  
Cambiar 'password' => bcrypt(str_random(10)), por  
'password' => \Illuminate\Support\Facades\Hash::make($faker->password()),
```



## Copiar archivos de migraciones

```
cp -v /var/www/laravel/apirf.lar/database/migrations/* ./database/migrations/
```

## Habilitar Eloquent

En el archivo </var/www/laravel/apirf.lumen/bootstrap/app.php> descomentar las siguientes líneas:

```
$app->withFacades();  
$app->withEloquent();
```

## Ejecutar migraciones y limpiar

```
php artisan migrate:install  
composer dump-autoload
```

## Copiar los controladores

```
cp -v /var/www/laravel/apirf.lar/app/Http/Controllers/ManufacturerController.php  
.app/Http/Controllers/  
cp -v /var/www/laravel/apirf.lar/app/Http/Controllers/VehicleController.php  
.app/Http/Controllers/
```

## Cambiar método validate en controladores

Cambiar el método validate, la forma de referenciarlo

Eliminar `use Illuminate\Support\Facades\Validator;`

cambiar esto `$validatedData = $request->validate($rules);` por `$this->validate($request, $rules);`

## Modificaar controladores para llamada de clase Lang

Estoy cambiando la forma de llamar la clase Lang

cambio de `use Lang;` por `use Illuminate\Support\Facades\Lang as Lang;`



# Verificar que los nombres de los métodos de los controladores coinciden con lo que está en el ruteador

## Habilitar el multilenguaje

### Copiar middleware de localización

```
cp -v /var/www/laravel/apirf.lar/app/Http/Middleware/localization.php  
./app/Http/Middleware/
```

### Registrar el middleware

Global Middleware – ya que se va a usar siempre  
\$app->middleware() method in your bootstrap/app.php file

### Copiar carpetas de lenguaje

```
cp -Rv /var/www/laravel/apirf.lar/resources/lang ./resources/lang
```

### Agregar variable por defecto al archivo .env

```
APP_LOCALE=ar
```

### Ejecutar la configuración del archivo .env

En /var/www/laravel/apirf.lumen/bootstrap/app.php  
hacer llamada \$app->configure('app'); para que tome los cambios del archivo de configuracion

### Configurar el idioma por defecto

En el middleware /var/www/laravel/apirf.lumen/app/Http/Middleware/localization.php  
cambiar la sintaxis **s App::setLocale(\$locale);** por **app('translator')->setLocale(\$locale);**  
Eliminar use Config



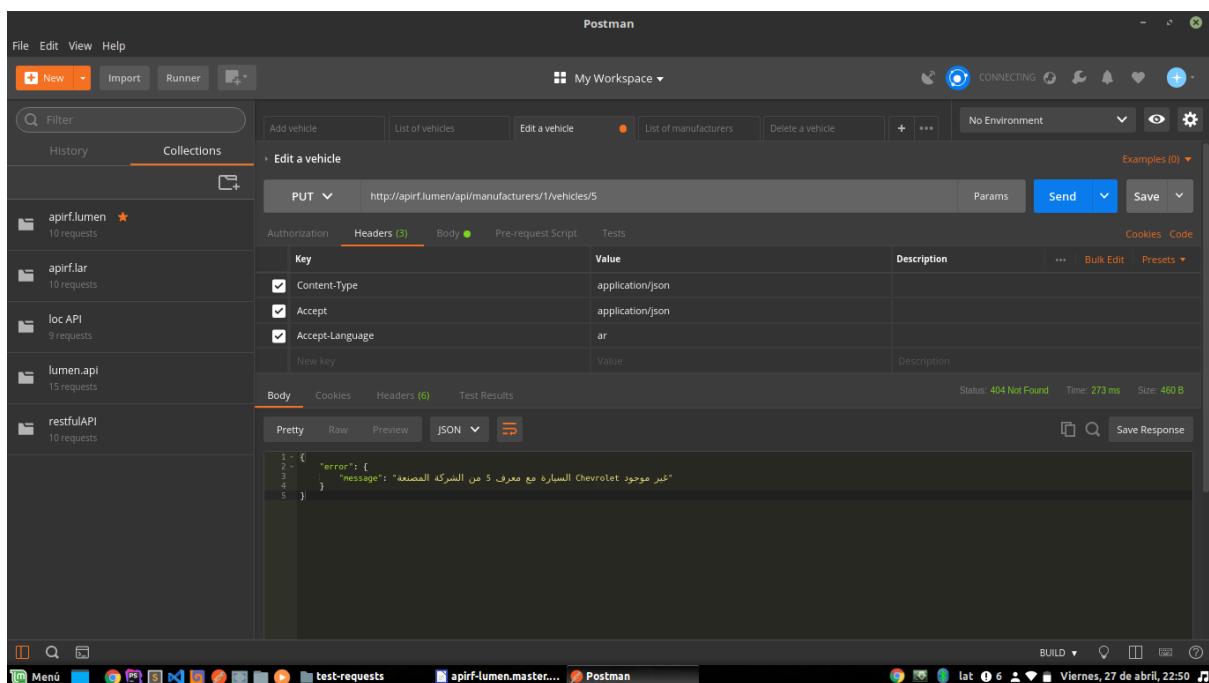
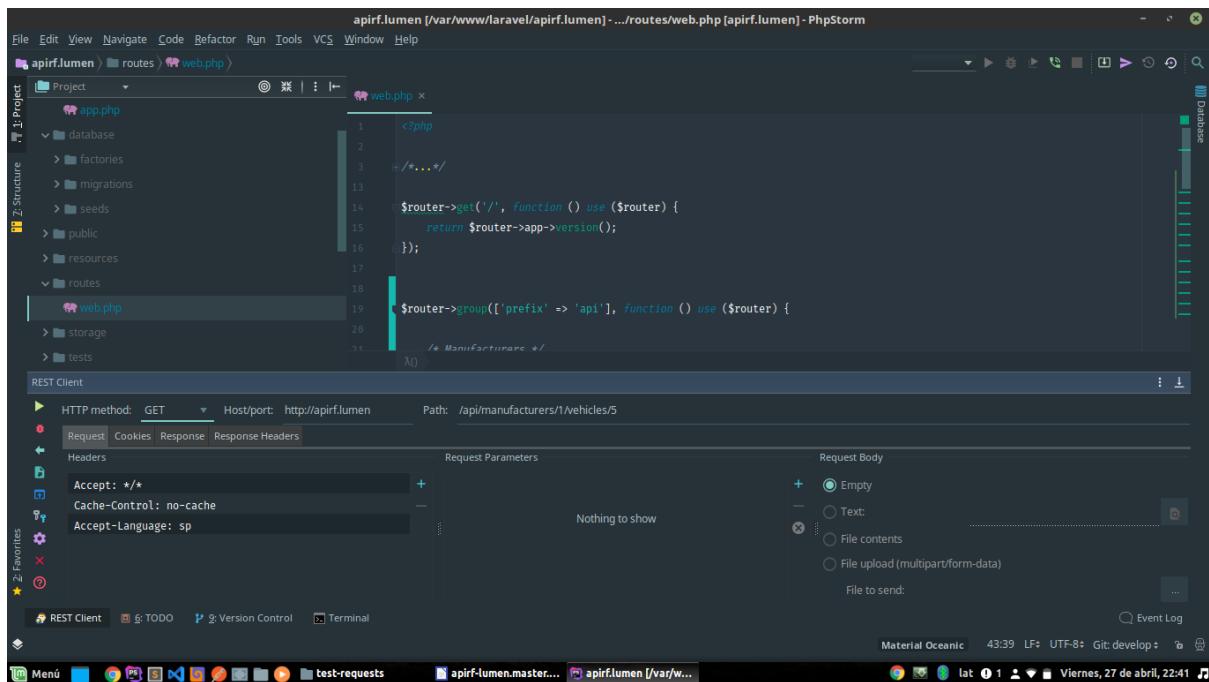
# Agregar rutas de los controladores

```
$router->group(['prefix' => 'api'], function () use ($router) {  
    /* Manufacturers */  
    $router->get('manufacturers', ['uses' => 'ManufacturerController@index']);  
    $router->get('manufacturers/{id}', ['uses' => 'ManufacturerController@show']);  
    $router->post('manufacturers', ['uses' => 'ManufacturerController@store']);  
    $router->delete('manufacturers/{id}', ['uses' => 'ManufacturerController@destroy']);  
    $router->put('manufacturers/{id}', ['uses' => 'ManufacturerController@update']);  
    $router->patch('manufacturers/{id}', ['uses' => 'ManufacturerController@update']);  
    /* Vehicles */  
    $router->get('manufacturers/{manufacturer_id}/vehicles', ['uses' => 'VehicleController@index']);  
    $router->get('manufacturers/{manufacturer_id}/vehicles/{id}', ['uses' => 'VehicleController@show']);  
    $router->post('manufacturers/{manufacturer_id}/vehicles', ['uses' => 'VehicleController@store']);  
    $router->delete('manufacturers/{manufacturer_id}/vehicles/{id}', ['uses' => 'VehicleController@destroy']);  
    $router->put('manufacturers/{manufacturer_id}/vehicles/{id}', ['uses' => 'VehicleController@update']);  
    $router->patch('manufacturers/{manufacturer_id}/vehicles/{id}', ['uses' => 'VehicleController@update']);  
});
```

## Testear la api

Con herramientas con Postman o similar, o con la que viene integrada en phpstorm. Hay archivos incluidos ya elaborados en el directorio **test-requests**. Sólo queda importar las peticiones desde Postman o el Client REST Tool de phpstorm





Repositorio: <https://github.com/riv4wi/apirf-laravel>

Rama: Master

Nombre-documento: apirf.lumen.master

# Documentación pública de la API

Adicionalmente se puede ver la documentación de la API en

<https://documenter.getpostman.com/view/3996143/apirflumen/RW1bkYQ9>

No se podrá probar a menos que se monte como local, ya que no está disponible en un servidor público en la web.

## Referencias web adicionales

<https://seegatesite.com/restful-api-tutorial-with-lumen-laravel-5-5-for-beginners/>

<https://autho.com/blog/developing-restful-apis-with-lumen/>

<https://coderexample.com/restful-api-in-lumen-a-laravel-micro-framework/>

<https://lumen.laravel.com/docs/5.6/middleware>

<https://stackoverflow.com/questions/32630039/how-to-change-locale-language-on-lumen>

<https://www.cloudways.com/blog/lumen-rest-api-authentication/>

<https://medium.com/@jpedraza/autenticaci%C3%B3n-por-token-en-lumen-f0558ed4961>

<https://autho.com/blog/developing-restful-apis-with-lumen/>



Repositorio: <https://github.com/riv4wi/apirf-laravel>

Rama: **Master**

Nombre documento: [apirf.lumen.master](#)