

Test Laravel

La aplicación permitirá a los gestores de una discográfica mantener su histórico de producciones, con la lista de los artistas que han participado en cada una. En particular, los usuarios deben poder:

1. Crear, modificar, borrar y listar (con todos los campos relacionados) álbumes
2. Crear, modificar, borrar y listar artistas
3. Añadir y quitar artistas de un álbum existente.

Notas adicionales:

- Un álbum tiene un título y una fecha de publicación.
- De los artistas se conoce el nombre y el instrumento o rol.
- Los artistas se presentarán en el orden en que han sido añadidos al álbum.
- La aplicación se ejecutará en un servidor accesible solo desde la oficina del promotor, y por lo tanto no necesita ningún tipo de control de acceso.

Descripción: La aplicación está implementada en Laravel 5.1 y Angular 1.4.8, haciendo uso del método resource de la clase Route que permite crear las rutas por defecto de una API Rest según las convenciones.

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	
	POST	album	album.store	App\Http\Controllers\AlbumController@store	
	GET HEAD	album	album.index	App\Http\Controllers\AlbumController@index	
	DELETE	album/{album}	album.destroy	App\Http\Controllers\AlbumController@destroy	
	GET HEAD	album/{album}	album.show	App\Http\Controllers\AlbumController@show	
	PUT PATCH	album/{album}	album.update	App\Http\Controllers\AlbumController@update	

Con angular se usó el módulo \$resource, el mismo que permite consumir API's Rest a través de métodos más naturales como lo son query(), get(), save() remove().

Instalación:

En el repositorio están incluidos todos los archivos necesarios para que funcione la aplicación, a excepción de la información de base de datos. En los siguientes pasos podrás realizar la instalación.

1- Descargar repositorio:

Puede ser a través de git haciendo uso de la terminal

```
$ git clone git@github.com:manuelm3z/test-laravel.git
```

o descargando el archivo zip desde el sitio web de github y descomprimirlo.

[Download ZIP](#)

2 - Creando vhost:

Ya sea usando MAMPP, XAMPP, WAMPP o tengas una instalación nativa de Apache + PHP y MySQL .

puedes crear un vhost para correr la aplicación con los siguientes directivas.

```
<VirtualHost *:80>
    DocumentRoot "/ruta_absoluta_a_carpeta_proyecto/public"
    ServerName laravel.local
    ErrorLog "/private/var/log/apache2/laravel.local-error_log"
    CustomLog "/private/var/log/apache2/laravel.local-access_log" common

    <Directory "/ruta_absoluta_a_carpeta_proyecto/public">
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Las rutas varían de acuerdo a tu sistema operativo en el caso de Windows debes incluir el alias del disco duro donde está almacenado por ejemplo C:/

3- Editar el archivo hosts.

En el caso de Windows se encuentra la siguiente ruta:

C:\Windows\System32\driver\etc\hosts

Esto se debe hacer ejecutando block de notas como administrador y te diriges a esa ruta.

y Sistemas Operativos Unix-like, ya sea OS X o Linux, lo encontrarás en la ruta:

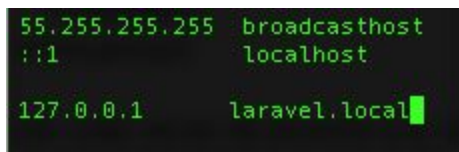
/etc/hosts

Esto es más cómodo hacer en la terminal, con el editor de tu preferencia.

Ejemplo: sudo nano /etc/hosts

Agregar el siguiente texto:

127.0.0.1 laravel.local



```
55.255.255.255 broadcasthost
::1 localhost
127.0.0.1 laravel.local
```

4- Editar archivo .env.example y guardar como .env

```
APP_ENV=local
APP_DEBUG=true
APP_KEY=SomeRandomString

DB_HOST=localhost
DB_DATABASE=homestead
DB_USERNAME=homestead
DB_PASSWORD=secret

CACHE_DRIVER=file
SESSION_DRIVER=file
QUEUE_DRIVER=sync

REDIS_HOST=localhost
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
```

Editarlo agregando la información de tu servidor de base de datos, en mi caso quedó de la siguiente forma:

```
APP_ENV=local
APP_DEBUG=true
APP_KEY=ye1jLLgUlll0kSSJCIGd1xNemRxEWflj

DB_HOST=localhost
DB_DATABASE=api
DB_USERNAME=root
DB_PASSWORD=1234

CACHE_DRIVER=file
SESSION_DRIVER=file
QUEUE_DRIVER=sync

REDIS_HOST=localhost
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
```

Consideraciones finales.

En el caso de estar usando Windows no tendrás problemas con respecto a permisos, pero en el caso de SO Unix-like, aun queda un paso por completar, debemos cambiar permisos en algunos directorios de nuestra aplicación.

/public
/storage

Desde la terminal, ejecutar el comando “chmod” para cambiar permisos y “chown” para cambiar de grupo. ejecutarlos desde la carpeta raíz del proyecto.

En el caso de public

```
sudo chmod -Rf 755 public
```

```
sudo chown -Rf nombre_usuario:_www public (en el caso de OS X)
```

```
sudo chown -Rf nombre_usuario:www-data public (en la mayoría de los SO Linux)
```

En el caso de storage, el servidor web debe tener permiso de escribir en este directorio.

```
sudo chmod -Rf 775 storage
```

```
sudo chown -Rf nombre_usuario:_www storage OS X
```

```
sudo chown -Rf nombre_usuario:www-data SO Linux
```

Después de todo esto ya podrás colocar en el navegador el alias escogido para en este caso laravel.local

Super Discs

Albums

Album Information

Title

Publication

Name

Artists

RoI

Album List

Title	Publication	Actions
Weathered	1999-09-28	<input type="button" value="Update"/> <input type="button" value="Delete"/>
Follow the Leader	1999-09-28	<input type="button" value="Update"/> <input type="button" value="Delete"/>