

Desarrollo de aplicaciones web con Laravel 5

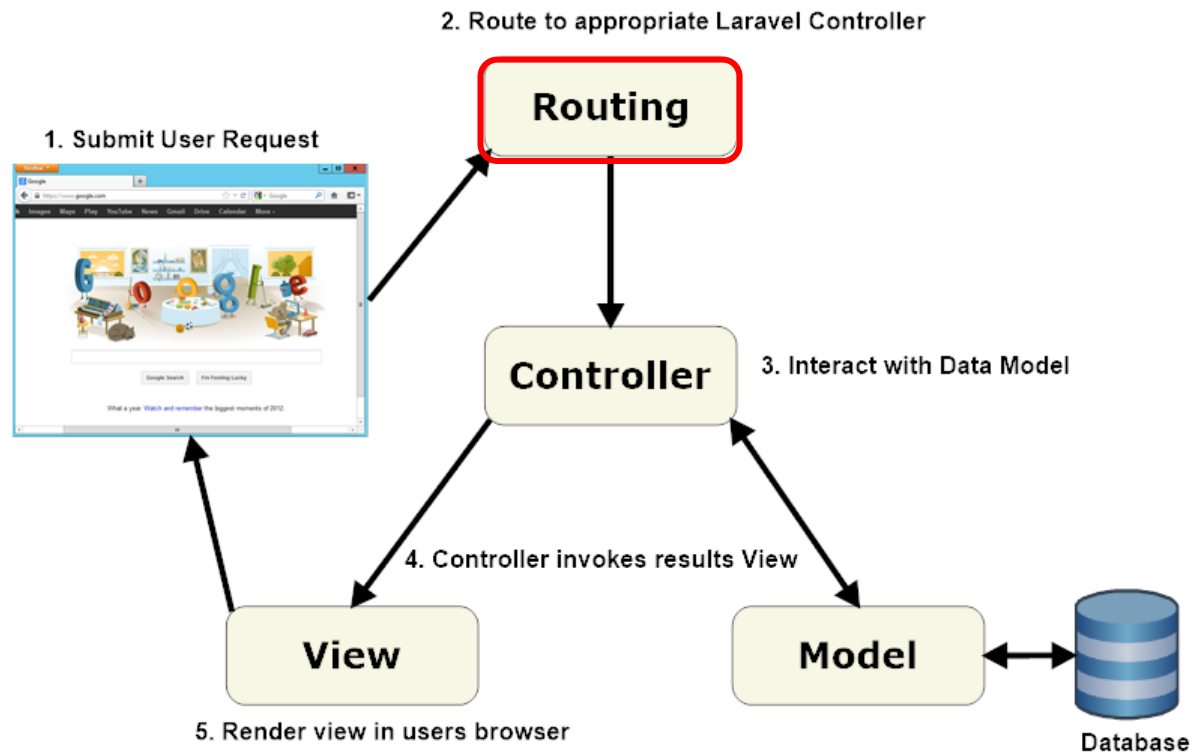
Clase 2. Rutas y Controladores
Carlos Ruiz Ruso • @micromante

Rutas

Routing

<http://laravel.com/docs/5.0/routing>

MVC con Ruteo



Rutas

- Potente clase de ruteo
- Gestiona las rutas y acciones de nuestra aplicación `app/routes.php`
- Podemos conocer las rutas definidas con `php artisan routes`
- Se encuentra en `app/Http/routes.php`
- La operación más básica se realiza con un Closure y una url
- Operaciones con verbos HTTP
 - GET, PUT, DELETE, POST,....

GET • Verbos HTTP

- Lo utilizamos para leer un recurso
- Operaciones
 - OK 200
 - BAD 404 or 400
- Ejemplo
 - <http://www.dominio.com/usuarios>
 - <http://www.dominio.com/usuarios/1>
 - <http://www.dominio.com/productos/paginate>

POST • Verbos HTTP

- Lo utilizamos para crear un recurso
- Operaciones
 - CREATED 201
 - BAD 404 or 400
- Ejemplo
 - <http://www.dominio.com/usuarios>
 - <http://www.dominio.com/productos>
 -

PUT • Verbos HTTP

- Lo utilizamos para actualizar un recurso
- Operaciones
 - UPDATED 200
 - BAD 404 or 400
- Ejemplo
 - <http://www.dominio.com/usuarios/20>
 - <http://www.dominio.com/usuarios/20/edit>
 - ...

DELETE • Verbos HTTP

- Lo utilizamos para eliminar un recurso
- Operaciones
 - OK 200
 - BAD 404 or 400
- Ejemplo
 - <http://www.dominio.com/usuarios/20>
 - <http://www.dominio.com/usuarios/20/delete>
 - ...

Tutorial sobre REST API

<http://www.restapitutorial.com/lessons/httpmethods.html>

Más información complementaria

Debemos tener en cuenta...

Cuando trabajamos con una API Rest las pruebas de la aplicación no podemos hacerlas desde el navegador porque solo podríamos probar las de tipo GET y POST

Usaremos POSTMAN - Rest client para chrome.

<https://chrome.google.com/webstore/detail/postman-rest-client/fdmmgilgnpjigdojojpjoooidkmcomcm?hl=en>

Creación rutas controladores

Route::verb(path,action)

- verb
 - get, put, post, delete, any(comodin para todos los verbos)
- path
 - dirección de la ruta, ejemplo www.dominio.com/producto
- action
 - Controlador que realiza las operaciones
 - Función anónima o closure que gestiona la operación

Creación rutas controladores

Ejemplos básicos ruteo de controladores

```
Route::get('/productos', function(){
```

```
...
```

```
});
```

```
Route::get('/productos', 'NameController@action');
```

Creación rutas controladores

Ejemplos con parámetros

```
Route::get('/productos/{name}/{description?}', function(){...});
```

```
Route::get('/productos/{name}/{description?}', function($name, $description = null ){...});
```

Paso al controlador

```
Route::get('/productos/{nameparam}/{nameother?}', 'NameController@action');
```

***Podemos definir parámetros opcionales con {variable?}, si son obligatorios la url no responderá si no encuentra el parámetro. La url debe cumplirse al completo para funcionar**

Creación rutas controladores

Rutas con agrupación bajo prefijo

```
Route::group(array('prefix' => 'v1/admin'), function(){  
    Route::get('usuarios',...);  
    Route::post(...);    ... más rutas agrupadas  
});
```

```
Route::group(['prefix' => 'v1/admin'], function(){ ... }); ← Nueva forma posible
```

Ahora la ruta sería: <http://www.dominio.com/v1/admin/usuarios>

Creación rutas controladores

Namespace y Middleware en Rutas (Cuando veamos los middleware podéis usar este método)

```
Route::group(['middleware' => 'auth', function() { <--orden según definición
```

```
    Route::get(perfil...)
```

```
    Route::post(...);    ... más rutas agrupadas
```

```
});
```

```
Route::group(['namespace' => 'Admin', function() {
```

```
    Route::get(perfil...) ← Controladores bajo App\Http\Controllers\Admin
```

```
});
```

Creación rutas controladores

Trabajando con un subdominio

```
Route::group(['domain' => {nombre}.miejemplo.com'], function(){ <--orden según definición
    Route::get(user/{id}, function(){...})
    Route::post(...);    ... más rutas agrupadas
});
```

Creación rutas controladores

Ejemplos con alias en la url

```
Route::get('/usuario/perfil', array('as' => profile, function(){ ... }));
```

```
Route::get('/usuario/perfil', array('as' => profile, 'uses' => 'UserController@showProfileInfo');
```

`redirect()->route('profile');` ← en un método de un controlador por ejemplo

```
$url = route('profile',['id' => 1]);
```

Ahora la redirección se podrá hacer con el alias

Muy útil cuando tengamos que crear un link desde la vista a la ruta o desde un controlador queremos redireccionar

Creación rutas controladores

Varios verbos a la vez por ruta o todos los verbos

```
Route::match(['get','post'], '/', function(){  
    ...  
});
```

```
Route::any('/', function(){ ... });
```

Creación rutas controladores

Puedes crear URLs a tus rutas:

```
$url = url('profile'); ← www.midominio.com/profile
```

Creación rutas controladores

Ejemplos con patrones en la url y patrones globales

Ejemplo para una sola ruta

`Route::post('usuarios/{id}/{other}', action)`

`->where(array('id' => '[0-9]+', 'other' => '[a-z]+'));`

Creación rutas controladores

Patrones globales en fichero RouteServiceProvider

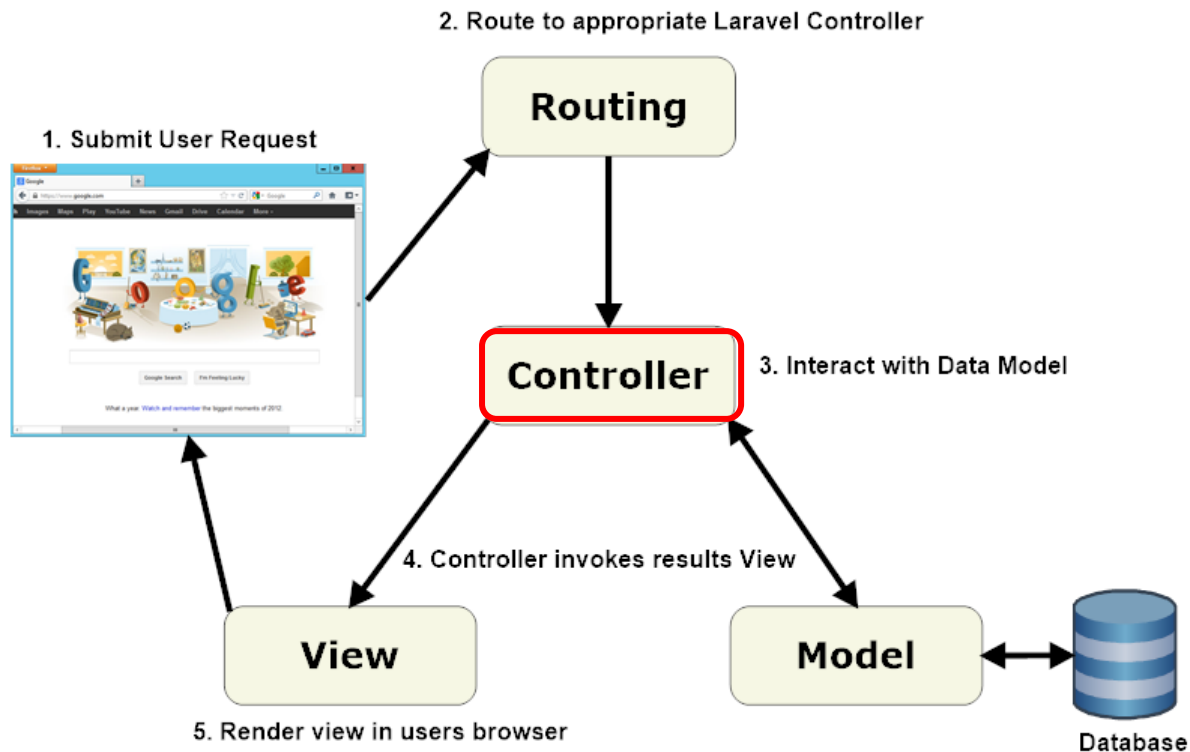
```
/**
 * Define your route model bindings, pattern filters, etc.
 *
 * @param \Illuminate\Routing\Router $router
 * @return void
 */
public function boot(Router $router)
{
    $router->pattern('id', '[0-9]+');

    parent::boot($router);
}
```

Controllers

Controladores en App\Http\Controllers

MVC con Ruteo

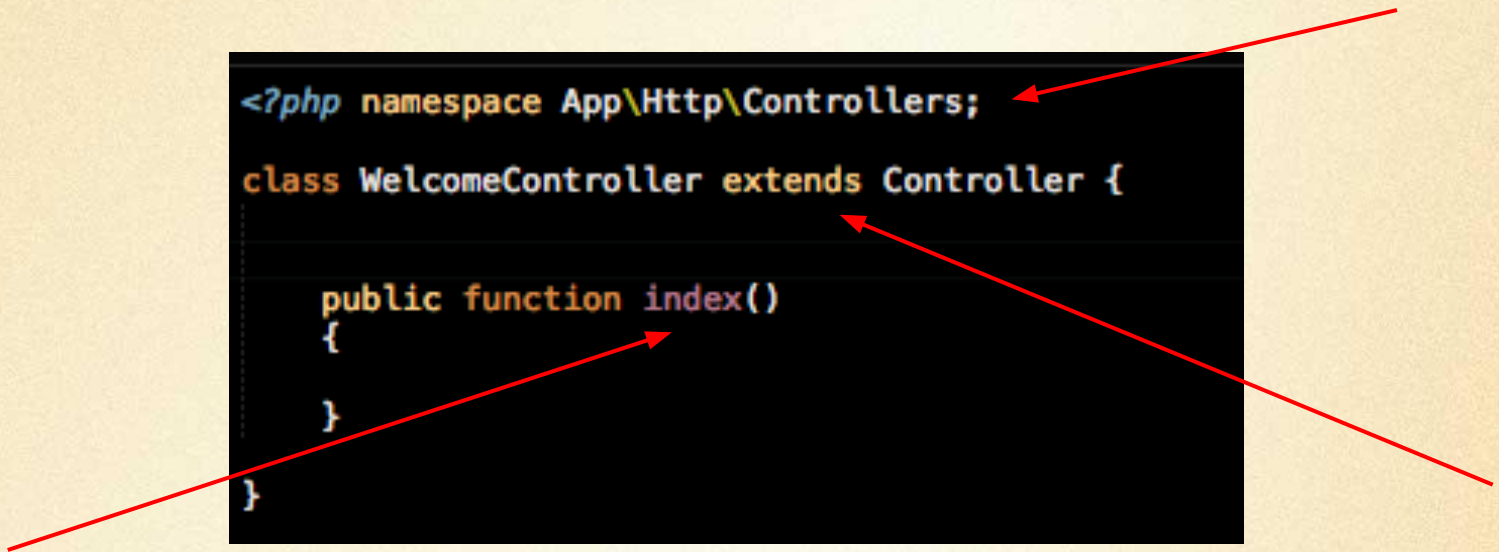


Controladores

- Una vez definidas las rutas debemos crear los controladores
- Permiten inyección de dependencias
 - Ejemplo: `new Suscriber(User $user);`
- Estan en `app/controllers`
- Registro del classmap en `composer.json`, podemos auto generarlos con `composer dump-autoload` pero con PSR-4 no es necesario normalmente.

Estructura de un controlador

```
<?php namespace App\Http\Controllers;  
  
class WelcomeController extends Controller {  
  
    public function index()  
    {  
  
    }  
  
}
```

A diagram illustrating the structure of a Laravel controller. It features a black rectangular box containing PHP code. Three red arrows originate from the right side of the slide and point to specific parts of the code: the top arrow points to the namespace declaration 'namespace App\Http\Controllers;', the middle arrow points to the class declaration 'class WelcomeController extends Controller {', and the bottom arrow points to the 'index()' method definition.

Generar un controlador RESTful desde Artisan

```
php artisan make:controller TestController
```


Ahora las rutas seran de tipo RESTful

Tabla de ejemplo y documentación completa de parámetros

<http://laravel.com/docs/master/controllers#restful-resource-controllers>

Verb	Path	Action	Route Name
GET	/resource	index	resource.index
GET	/resource/create	create	resource.create
POST	/resource	store	resource.store
GET	/resource/{resource}	show	resource.show
GET	/resource/{resource}/edit	edit	resource.edit
PUT/PATCH	/resource/{resource}	update	resource.update
DELETE	/resource/{resource}	destroy	resource.destroy

Generar un controlador RESTful desde Artisan


Con el service container podemos hacer uso de la inyección de dependencias. ¿y esto para que sirve?

```
<?php namespace App\Http\Controllers;

use Illuminate\Routing\Controller;
use App\Repositories\UserRepository;

class UserController extends Controller
{
    /**
     * The user repository instance.
     */
    protected $users;

    /**
     * Create a new controller instance.
     *
     * @param UserRepository $users
     * @return void
     */
    public function __construct(UserRepository $users)
    {
        $this->users = $users;
    }
}
```



Siguiente paso con los controladores...

Aprender a rellenarlos en los próximos temas

Por hoy ya está...una cosa más...

¡Ahora os toca a vosotros!

Ejercicio para casa

Crear una pequeña API Rest con los controladores y rutas para un supuesto sistema con productos y usuarios
Probarlo todo en POSTMAN con Chrome

¡MUCHAS GRACIAS A TODOS POR VUESTRA PARTICIPACIÓN!

Aquí me teneis para lo que os pueda ayudar.
Estais invitados a seguirme en la redes sociales
como @micromante o Carlos Ruiz Ruso
www.micromante.com