|  |  |
| --- | --- |
|  | **UNIVERSIDAD DON BOSCO**  **FACULTAD DE ESTUDIOS TECNOLÓGICOS**  **COORDINACIÓN DE COMPUTACIÓN Y MÓVILES** |
| **Ciclo II** | **Desarrollo de Avanzado de Software para móviles** Guía de Laboratorio No. 4 Laravel |

* **I Objetivos**
* Configurar Laravel para crear aplicaciones php y crear webservices.
* Utilizar motor de plantillas blade para generar backends.
* Utilizar el ORM (Object-Relational mapping) Eloquent para acceso de bases de datos.
* **II Desarrollo**

**Introducción a Laravel.**

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

**Instalar composer.**

Composer es un gestor de dependencias de PHP, independiente de la plataforma. Realizaremos la descarga para MacOS o bien para sus sistema operativo respectivo.

<https://www.abeautifulsite.net/installing-composer-on-os-x>

**Realizar QuickStar de Laravel**

No hay mejor ejemplo que la guía rápida de Laravel. Procederá a realizarla, tomando en cuenta que la versión de su proyecto es Laravel 5.6 y además el archivo de rutas se encuentra en routes/web.php o routes/api.php

<https://laravel.com/docs/5.1/quickstart>

Configure su base de datos. Puede verificar si tiene instalado MySQL, puede instalarlo para MacOS o solicitar al docente que le facilite una base de datos remota.

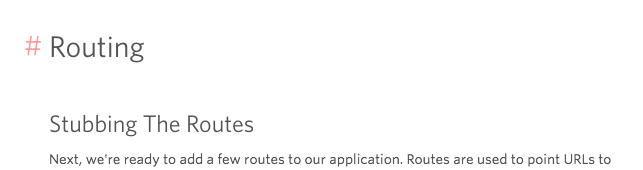
Localmente procederá a editar el archivo para la configuración de la base de datos el cuál es **.env**

**Consideraciones del ejemplo:**

Una vez iniciado el proceso del quickstar debe crear un controlador denominado TaskController:

php artisan make:controller TaskController --resource

Deberá hacer redirecciones hacia /tasks de manera correcta. **Consulte a su docente cómo asociar el controlador a las rutas.** Usted encontrará el proceso en la sección Routing, teniendo en cuenta que el único cambio que realizará es que la ruta no se encuentra en app/Http/routes, si no que en routes/web.php.



**Creando un WebService Simple.**

Crear un nuevo Controller denominado TaskApiController:

|  |
| --- |
| *php artisan make:controller TaskApiController --resource* |

Agregar la ruta del endpoint en routes/api.php

|  |
| --- |
| *Route::resource('api/v1/tasks', 'TaskApiController');* |

El funcionamiento es similar, pero no retornará vistas si no que arrays y estos se transformarán automáticamente en formato json.

Ejemplo:

|  |
| --- |
| public function index()  {  return $tasks = Task::orderBy('created\_at', 'asc')->get();  } |

Para probar nuestra api editaremos:

App\Http\Middleware\VerifyCsrf.php



Ahora agregue el método post:

|  |
| --- |
| public function store(Request $request)  {  $validator = Validator::make($request->all(), [  'name' => 'required|max:255',  ]);  if ($validator->fails()) {  return array("error"=>"Fields Required", "fields" => $validator->errors()->all());  } |

* **III Ejercicios Complementarios**

Crear un proyecto en Laravel, para registrar un control de calorías personal. Debe agregar el nombre y descripción del alimento que consume, el aproximado de calorías y la fecha y hora de consumo. Debe crear el webservice para agregar, listar,modificar y eliminar el registro. Para manejar la información, puede ser utilizando únicamente el webservice o bien creando un backend con laravel.

Investigue el funcionamiento de:

csrf\_field()