

```
vagrant@homestead:~/code/api_rest_full$ artisan make:model Pokemon --all
Model created successfully.
Factory created successfully.
Created Migration: 2019_02_19_204742_create_pokemon_table
Controller created successfully.
vagrant@homestead:~/code/api_rest_full$
```

Creamos un nuevo modelo con la **opción --all** que permite crear automáticamente el **factory**, la **migration** y el **controlador**

4. En el fichero api.php de la carpeta routes añade las rutas mediante el método `resource: Route::resource('NombreModelo', 'NombreControlador');`

```
20 Route::resource('Pokemon', 'PokemonController');
```

5. ¿Por qué crees que se utiliza este fichero api.php en lugar de web.php, como hasta ahora?

La diferencia son la utilización de middleware y los prefijos de rutas, en web.php se aplican middleware para la sesión, cookies, bindings y protección CSRF, en cambio en api.php solo se aplica el numero de peticiones limites que se puede realizar.

A continuación, en el archivo **“App\Http\Kernel.php”**, vemos las aplicaciones que se realizan a cada uno:

```
protected $middlewareGroups = [
    'web' => [
        \App\Http\Middleware\EncryptCookies::class,
        \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
        \Illuminate\Session\Middleware\StartSession::class,
        // \Illuminate\Session\Middleware\AuthenticateSession::class,
        \Illuminate\View\Middleware\ShareErrorsFromSession::class,
        \App\Http\Middleware\VerifyCsrfToken::class,
        \Illuminate\Routing\Middleware\SubstituteBindings::class,
    ],

    'api' => [
        'throttle:60,1',
        'bindings',
    ],
];
```

La otra diferencia notable que encontramos, es el prefijo de las rutas, cuando indicamos en api.php la ruta por ejemplo “Pokemon”, crea las rutas con el prefijo `/api/Pokemon/..`

6. Consulta la tabla de rutas. Para una API restFul hay dos rutas de las que se muestran que no necesitamos. ¿Cuáles son? **GET CREATE y GET EDIT**

```
vagrant@homestead:~/code/api_rest_full$ artisan route:list
```

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	GET HEAD	api/Pokemon	Pokemon.index	App\Http\Controllers\PokemonController@index	api
	POST	api/Pokemon	Pokemon.store	App\Http\Controllers\PokemonController@store	api
	GET HEAD	api/Pokemon/create	Pokemon.create	App\Http\Controllers\PokemonController@create	api
	GET HEAD	api/Pokemon/{Pokemon}	Pokemon.show	App\Http\Controllers\PokemonController@show	api
	PUT PATCH	api/Pokemon/{Pokemon}	Pokemon.update	App\Http\Controllers\PokemonController@update	api
	DELETE	api/Pokemon/{Pokemon}	Pokemon.destroy	App\Http\Controllers\PokemonController@destroy	api
	GET HEAD	api/Pokemon/{Pokemon}/edit	Pokemon.edit	App\Http\Controllers\PokemonController@edit	api

7. ¿Para qué se utiliza Route::apiResource? Utilízala y vuelve a consultar la tabla de rutas. ¿Qué diferencias hay respecto al punto anterior? **No crea las rutas GET/CREATE GET/UPDATE, ya que no se puede actualizar la api por GET solo POST y PUT**

```
vagrant@homestead:~/code/api_rest_full$ artisan route:list
```

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

8. Elimina los métodos del controlador que no se usen en la API. ¿Para qué se utiliza el verbo HTTP PATCH?

PATCH es como el verbo HTTP PUT la diferencia según el [Estándar RFC 5789](#), el método PUT modifica por completo el documento, pero PATCH permite modificar parcialmente el documento.

¿Qué tienen de particular los parámetros de los métodos show, update y destroy?

Tienen como argumentos el modelo (Pokemon)

9. Crea un seed para la tabla pokemon: artisan make:seeder PokemonTableSeeder

```
PokemonController.php x PokemonTableSeeder.php x DatabaseSeeder.php

class PokemonTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        factory(App\Pokemon::class, 50)->create();
    }
}
```

En el método run especificamos que cree 50 registros mediante factory, (permite generar datos aleatorios.)

```
public function up()
{
    Schema::create( table: 'pokemon', function (Blueprint $table) {
        $table->increments( column: 'id');
        $table->string( column: 'name');
        $table->timestamps();
    });
}
```

En el método up creamos una columna tipo string que será el nombre.

```
UserFactory.php
migrations
  2014_10_12_000000_create_users_table.php
  2014_10_12_100000_create_password_resets_table.php
  2019_02_19_204742_create_pokemon_table.php
seeds
  DatabaseSeeder.php
  PokemonTableSeeder.php
.gitignore

DatabaseSeeder.php
8      * Seed the application's database.
9      *
10     * @return void
11     */
12     public function run()
13     {
14         // $this->call(UsersTableSeeder::class);
15         $this->call( class: PokemonTableSeeder::class);
16     }
17 }
```

Terminal: Local x +

```
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2019_02_19_204742_create_pokemon_table
Migrated: 2019_02_19_204742_create_pokemon_table
vagrant@homestead:~/code/api_rest_full$ artisan db:seed
Seeding: PokemonTableSeeder
Database seeding completed successfully.
vagrant@homestead:~/code/api_rest_full$
```

Por último, en el método run llamamos a la seeder.

11. En el archivo PokemonFactory añade el siguiente código: `return ['name' => $faker->name];` ¿Qué hace este código?

```
<?php

use Faker\Generator as Faker;

$factory->define(App\Pokemon::class, function (Faker $faker) {
    return [
        'name' => $faker->name,
    ];
});
```

Permite crear en la columna **nombre** de la tabla, **nombres** aleatorios

Completa todos los métodos de la API y compruébalos con postman o curl. Puedes ejecutar el comando `artisan serve --host=192.168.10.10 --port=8080` para ejecutar el servidor en una IP concreta y en un puerto determinado.

GET

GET 192.168.10.10:8080/api/Pokemon

Params Authorization Headers Body Pre-request Script Tests

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (9) Test Results

Status: 200 OK Time: 1137 ms Size: 5

Pretty Raw Preview JSON

```
1 {
2   [
3     {
4       "id": 1,
5       "name": "Alexander Olson IV",
6       "created_at": "2019-02-19 21:15:26",
7       "updated_at": "2019-02-19 21:15:26"
8     },
9     {
10      "id": 2,
11      "name": "Sonia Ward",
12      "created_at": "2019-02-19 21:15:26",
13      "updated_at": "2019-02-19 21:15:26"
14    },
15    {
16      "id": 3,
17      "name": "Brown Dach",
18      "created_at": "2019-02-19 21:15:26",
19      "updated_at": "2019-02-19 21:15:26"
20    }
21  ]
22 }
```

Realizamos un get a un pokemon en concreto

GET 192.168.10.10:8080/api/Pokemon/4

Params Authorization Headers Body Pre-request Script Tests

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (9) Test Results

Status: 200 OK

Pretty Raw Preview JSON

```
1 {
2   "id": 4,
3   "name": "Deion Cormier",
4   "created_at": "2019-02-19 21:15:26",
5   "updated_at": "2019-02-19 21:15:26"
6 }
```

Realizamos un POST

POST 192.168.10.10:8080/api/Pokemon/ Send

Params Authorization Headers (1) Body Pre-request Script Tests Cookies Code

none form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2
3   "name": "ivan"
4
5 }
```

Body Cookies Headers (9) Test Results Status: 200 OK Time: 558 ms Size: 358 B

Pretty Raw Preview HTML

i 1 Registro insertado correctamente: ivan

Realizamos un PUT

192.168.10.10:8080/api/Pokemon/53

PUT 192.168.10.10:8080/api/Pokemon/53 Send

Params Authorization Headers (1) Body Pre-request Script Tests Cookies C

none form-data x-www-form-urlencoded raw binary JSON (application/json)







```
1 {
2
3   "name": "ivancito"
4
5 }
```

Body Cookies Headers (9) Test Results Status: 200 OK Time: 650 ms Size: 364 B

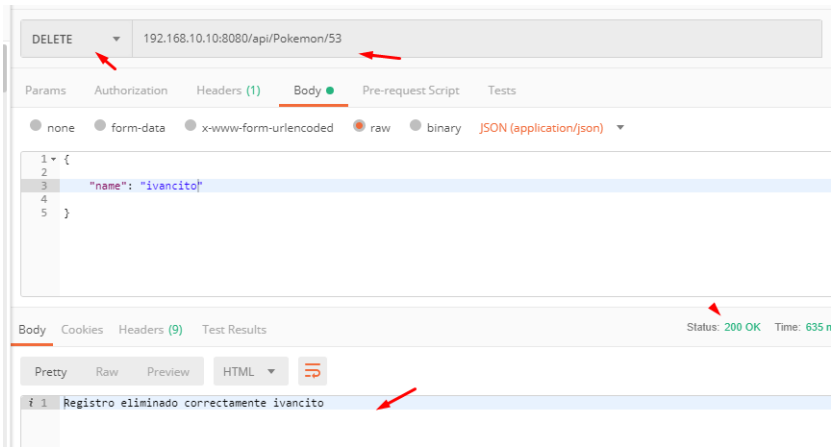
Pretty Raw Preview HTML

i 1 Registro ACTUALIZADO correctamente: ivancito

Comprobamos en la BBDD que los datos han sido modificados.

Result Grid			 Filter Rows:	<input type="text"/>	Edit:				Export/Import:	
id	name	created_at	updated_at							
37	Jason Torphy	2019-02-19 21:15:26	2019-02-19 21:15:26							
38	Dr. Kristofer Bosco	2019-02-19 21:15:26	2019-02-19 21:15:26							
39	Claude Muller	2019-02-19 21:15:26	2019-02-19 21:15:26							
40	Mrs. Audra O'Connell D...	2019-02-19 21:15:26	2019-02-19 21:15:26							
41	Isabella Schuppe	2019-02-19 21:15:26	2019-02-19 21:15:26							
42	Dr. Adolf Bogan MD	2019-02-19 21:15:26	2019-02-19 21:15:26							
43	Prof. Anais Green II	2019-02-19 21:15:26	2019-02-19 21:15:26							
44	Ona Satterfield	2019-02-19 21:15:26	2019-02-19 21:15:26							
45	Prof. Hanna Borer I	2019-02-19 21:15:26	2019-02-19 21:15:26							
46	Michale Leannon IV	2019-02-19 21:15:26	2019-02-19 21:15:26							
47	Kayden Kuphal	2019-02-19 21:15:26	2019-02-19 21:15:26							
48	Lorine Schneider	2019-02-19 21:15:26	2019-02-19 21:15:26							
49	Alford King	2019-02-19 21:15:26	2019-02-19 21:15:26							
50	Dr. Chris Abshire MD	2019-02-19 21:15:26	2019-02-19 21:15:26							
53	ivancito	2019-02-19 21:39:54	2019-02-19 21:45:09							
*	NULL	NULL	NULL							

Realizamos un DELETE



Comprobamos que el registro ya no se encuentra en la BBDD

Result Grid				
Filter Rows:				
	id	name	created_at	updated_at
	36	Zachariah Graham MD	2019-02-19 21:15:26	2019-02-19 21:15:26
	37	Jason Torphy	2019-02-19 21:15:26	2019-02-19 21:15:26
	38	Dr. Kristofer Bosco	2019-02-19 21:15:26	2019-02-19 21:15:26
	39	Claude Muller	2019-02-19 21:15:26	2019-02-19 21:15:26
	40	Mrs. Audra O'Connell D...	2019-02-19 21:15:26	2019-02-19 21:15:26
	41	Isabella Schuppe	2019-02-19 21:15:26	2019-02-19 21:15:26
	42	Dr. Adolf Bogan MD	2019-02-19 21:15:26	2019-02-19 21:15:26
	43	Prof. Anais Green II	2019-02-19 21:15:26	2019-02-19 21:15:26
	44	Ona Satterfield	2019-02-19 21:15:26	2019-02-19 21:15:26
	45	Prof. Hanna Borer I	2019-02-19 21:15:26	2019-02-19 21:15:26
	46	Michale Leannon IV	2019-02-19 21:15:26	2019-02-19 21:15:26
	47	Kayden Kuphal	2019-02-19 21:15:26	2019-02-19 21:15:26
	48	Lorine Schneider	2019-02-19 21:15:26	2019-02-19 21:15:26
	49	Alford King	2019-02-19 21:15:26	2019-02-19 21:15:26
	50	Dr. Chris Abshire MD	2019-02-19 21:15:26	2019-02-19 21:15:26
		NULL	NULL	NULL

14. En el siguiente [enlace](#) tienes una explicación sobre cómo proteger una ruta con autenticación básica y con autenticación básica sin estado. ¿Cuál es la diferencia entre ellas?

La autenticación básica almacena una cookie y crea una sesión, en cambio autenticación HTTP sin estado, no almacenaría la cookie ni sesión

Protege una ruta y comprueba que no puedes acceder, obteniendo un error 401.

Creamos el middleware AuthenticateOnceWithBasicAuth

```
<?php

namespace App\Http\Middleware;

use Illuminate\Support\Facades\Auth;
use Closure;

class AuthenticateOnceWithBasicAuth
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        return Auth::onceBasic() ?: $next($request);
    }
}
```

Registramos el middleware en app/Http/Kernel.php

```
nonController.php x AuthenticateOnceWithBasicAuth.php x PokemonTableSeeder.php x 2019_02_19_204742_create_pokemon_table.php x Kernel.php x

59      * These middleware may be assigned to groups or used individually.
60      *
61      * @var array
62      */
63      protected $routeMiddleware = [
64          'auth' => \App\Http\Middleware\Authenticate::class,
65          // 'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
66          'auth.basic.once' => \App\Http\Middleware\AuthenticateOnceWithBasicAuth::class,
67          'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,
68          'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
69          'can' => \Illuminate\Auth\Middleware\Authorize::class,
70          'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
71          'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
72          'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
73          'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
74      ];
75  }
```

Creamos un usuario nuevo utilizando el sistema de autenticación predeterminado de laravel. No se puede registrar el usuario “a pelo”, se debe realizar desde el formulario de registro.

Probamos primero con **estado**, y comprobamos que se almacena las cookies en el cliente.

← → ↺ 192.168.10.10:8080/api/Pokemon/

Aplicaciones Volver Aulas Virtuales CIFP... + Yomvi es Movist

[[{"id":1,"name":"Alexander Olson IV","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"},{"id":3,"name":"B. Cormier","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"},{"id":6,"name":"Baylee Price PhD","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"},{"id":8,"name":"E. Hill","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"},{"id":11,"name":"Keaton Wisoky","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"},{"id":13,"name":"Hammes","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"},{"id":16,"name":"Lila Douglas","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"},{"id":18,"name":"Hodkiewicz","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":21,"name":"Shakira Bosco V","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":23,"name":"Skiles I","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":26,"name":"Ms. Alana Kulas","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":28,"name":"Ankunding","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":31,"name":"Miss Dina Grimes DVM","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":33,"name":"I","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":36,"name":"Zachariah Graham MD","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":38,"name":"Muller","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":41,"name":"Isabella Schuppe","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":43,"name":"Satterfield","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":46,"name":"Michale Leannon IV","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":48,"name":"King","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}]]

Cookies en uso

Permitido Bloqueado

Se han permitido las siguientes cookies al visitar esta página

192.168.10.10

Cookies

XSRF-TOKEN

laravel_session

Nombre	no se ha seleccionado ninguna cookie
Contenido	no se ha seleccionado ninguna cookie
Dominio	no se ha seleccionado ninguna cookie
Ruta	no se ha seleccionado ninguna cookie
Enviar para	no se ha seleccionado ninguna cookie
Creada	no se ha seleccionado ninguna cookie
Caduca	no se ha seleccionado ninguna cookie

FuerteTienda > Pan...

Proceso de resoluci...

nia Ward","created_at":"2019-02-19 21:15:26"}, {"id":4,"name":"Deion Cornier","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":6,"name":"Mr. Fredy Satterfield","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":8,"name":"Jeff Hamill PhD","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":14,"name":"Sister Runolfsdottir","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":19,"name":"Reagan Larkin","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":24,"name":"Dr. Marilyn Labadie","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":29,"name":"Maudie Yost","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":34,"name":"Sally Will Torphy","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":39,"name":"Clau Bogdan MD","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":44,"name":"Ona Kuphal","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}, {"id":49,"name":"Alford Ward","created_at":"2019-02-19 21:15:26","updated_at":"2019-02-19 21:15:26"}]]

Probamos **sin estado**, y vemos como no se almacena cookies.

192.168.10.10:8080/api/Pokemon/ X

← → ↺ 192.168.10.10:8080/api/Pokemon/

Guardar Copiar Contraer todo Expandir todo Filtar JSON

0:

id: 1

name: "Alexander Olson IV"

created_at: "2019-02-19 21:15:26"

updated_at: "2019-02-19 21:15:26"

1:

id: 2

name: "Sonia Ward"

created_at: "2019-02-19 21:15:26"

updated_at: "2019-02-19 21:15:26"

2:

id: 3

name: "Brown Bach"

created_at: "2019-02-19 21:15:26"

updated_at: "2019-02-19 21:15:26"

3:

id: 4

name: "Deion Cornier"

created_at: "2019-02-19 21:15:26"

updated_at: "2019-02-19 21:15:26"

4:

id: 5

name: "Mr. Sanford Prohaska II"

created_at: "2019-02-19 21:15:26"

updated_at: "2019-02-19 21:15:26"

Inspector Consola Depurador {} Editor de estilos Almacenamiento

Almacenamiento local

Almacenamiento de sesión

Almacenamiento en caché

Cookies

http://192.168.10.10:8080

Indexed DB

Nombre	Dominio	Ruta	Caduca el	Último acceso el	Valor
No hay datos para el servidor seleccionado					