**Les API REST**

REST (Representational State Transfer) est un style d’architecture pour constituer des API. Pour Laravel une API REST est :

* organisée autours de ressources
* sans persistance (aucune session)
* orientée client-serveur
* pouvant être mis en cache (donc une requête doit retourner toujours les mêmes résultats)
* retourner du JSON…

### **Le contrôleur**

On va créer le contrôleur :

php artisan make:controller CarApiController --api

Avec évidemment le même code que celui qu’on a eu quand on avait déjà créé une ressource (et Laravel se débrouille tout seul pour les espaces de nom) mais sans les méthodes **create** et **edit** qui n’ont pas lieu d’être pour une API :

<?php

namespace App\Http\Controllers;

use App\Models\Car;

use Illuminate\Http\Request;

use Validator;

class CarApiController extends Controller

{

    /\*\*

     \* Display a listing of the resource.

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function index()

    {

        //

    }

    /\*\*

     \* Store a newly created resource in storage.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @return \Illuminate\Http\Response

     \*/

    public function store(Request $request)

    {

//

    }

    /\*\*

     \* Display the specified resource.

     \*

     \* @param  \App\Models\Car  $car

     \* @return \Illuminate\Http\Response

     \*/

    public function show(Car $car)

    {

        //

    }

    /\*\*

     \* Update the specified resource in storage.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @param  \App\Models\Car  $car

     \* @return \Illuminate\Http\Response

     \*/

    public function update(Request $request, Car $car)

    {

        //

    }

    /\*\*

     \* Remove the specified resource from storage.

     \*

     \* @param  \App\Models\Car  $car

     \* @return \Illuminate\Http\Response

     \*/

    public function destroy(Car $car)

    {

        //

    }

}

Et on va coder les méthodes :

<?php

namespace App\Http\Controllers;

use App\Models\Car;

use Illuminate\Http\Request;

use Validator;

class CarApiController extends Controller

{

    /\*\*

     \* Display a listing of the resource.

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function index()

    {

        $cars = Car::orderBy('created\_at')->get();

        return $cars->toJson();

    }

    /\*\*

     \* Store a newly created resource in storage.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @return \Illuminate\Http\Response

     \*/

    public function store(Request $request)

    {

        $rules = [

            'marque' => 'required',

            'prix' => 'required|numeric'

        ];

        $Message=[

            'marque.required' => 'حقل اجباري',

            'prix.required' => 'حقل اجباري',

            'prix.numeric' => 'حقل يكون في شكل عدد'

        ];

        $validator = Validator::make($request->all(),$rules,$Message);

        if($validator->fails()){

            return response()->json($validator->errors(), 400);

        }else {

        if(Car::create($request->all())){

            return response()->json([

               "message" => "Voiture enregistré avec succes"

           ],200);

       }

    }

    }

    /\*\*

     \* Display the specified resource.

     \*

     \* @param  \App\Models\Car  $car

     \* @return \Illuminate\Http\Response

     \*/

    public function show(Car $car)

    {

        return $car->toJson();

    }

    /\*\*

     \* Update the specified resource in storage.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @param  \App\Models\Car  $car

     \* @return \Illuminate\Http\Response

     \*/

    public function update(Request $request, Car $car)

    {

        if($car->update($request->all())){

            return response()->json(["message"=>"Voiture modifié avec success"], 200);

        };

    }

    /\*\*

     \* Remove the specified resource from storage.

     \*

     \* @param  \App\Models\Car  $car

     \* @return \Illuminate\Http\Response

     \*/

    public function destroy(Car $car)

    {

        if($car->delete()){

            return response()->json(["message"=>"Voiture supprimée"], 200);

        }

    }

}

Évidemment ça ressemble énormément à ce qu’on a fait précédemment et c’est normal puisqu’il s’agit du même genre de traitement sur les mêmes données.

### **Les routes**

Pour les routes on adopte aussi une ressource dans le fichier **routes/api.php** :

**use** App\Http\Controllers\CarApiController;

Route::apiResource(Cars, CarApiController::**class**);