Partie 2 : Mise en place d’une application Gestion Bibliothèque : Controllers

Auteur



<?php

namespace App\Http\Controllers;

use App\Models\Auteur;

use Illuminate\Http\Request;

class AuteurController extends Controller

{

    /\*\*

     \* Display a listing of the resource.

     \*/

    public function index()

    {

        $auteurs = Auteur::all()->toArray();

        return array\_reverse($auteurs);

    }

    /\*\*

     \* Store a newly created resource in storage.

     \*/

    public function store(Request $request)

    {

        $auteur = new Auteur([

            'nomauteur' => $request->input('nomauteur'),

            'email' => $request->input('email'),

            'numtel' => $request->input('numtel')

            ]);

        $auteur->save();

        return response()->json('Auteur créé !');

    }

    /\*\*

     \* Display the specified resource.

     \*/

    public function show($id)

    {

        $auteur= Auteur::find($id);

        return response()->json($auteur);

    }

    /\*\*

     \* Update the specified resource in storage.

     \*/

    public function update($id, Request $request)

    {

        $auteur = Auteur::find($id);

        $auteur->update($request->all());

        return response()->json($auteur);

    }

    /\*\*

     \* Remove the specified resource from storage.

     \*/

    public function destroy($id)

    {

        $auteur = Auteur::find($id);

        $auteur->delete();

        return response()->json(['message' => 'Auteur deleted successfully']);

    }

}



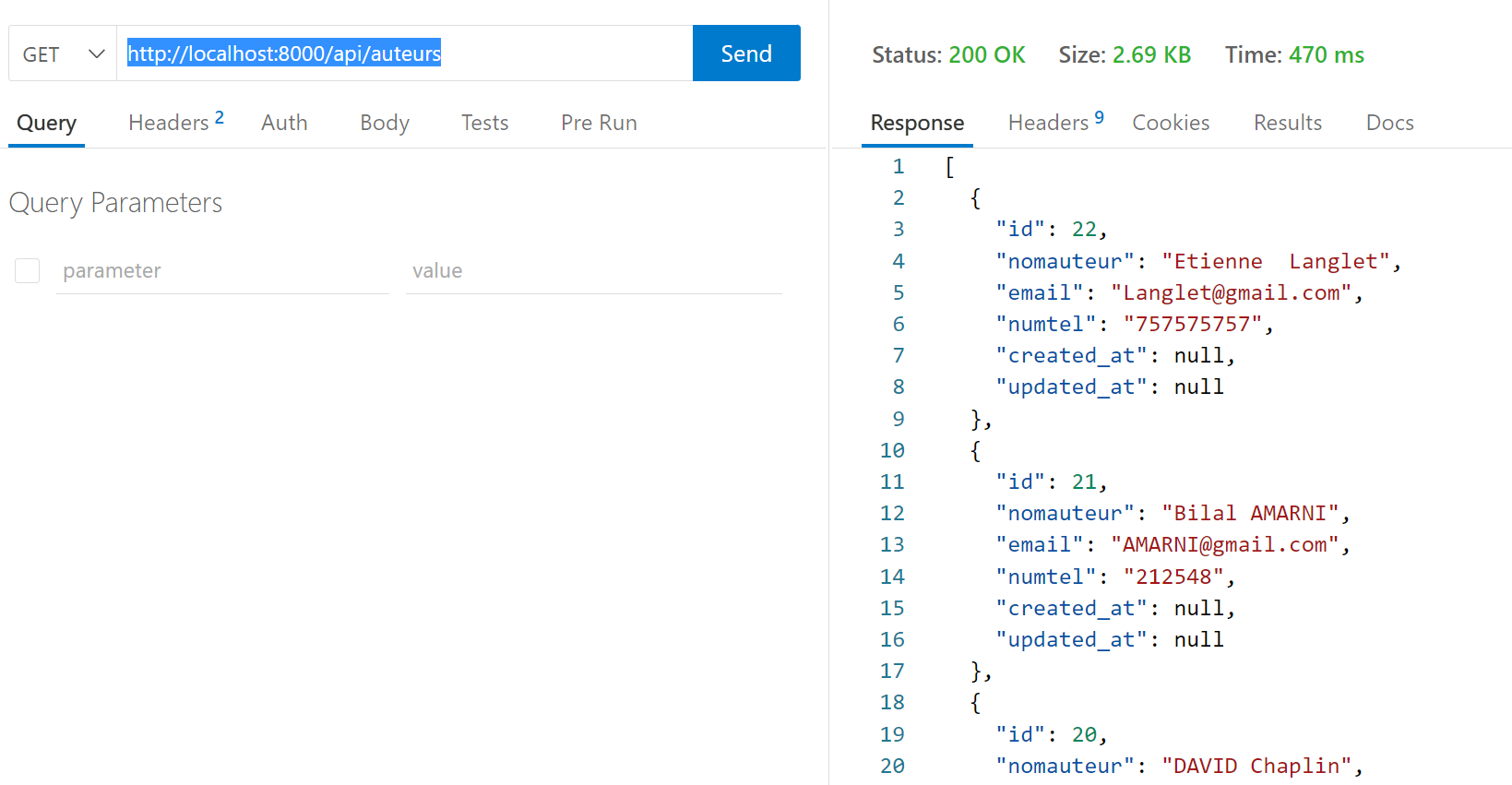
use App\Http\Controllers\AuteurController;

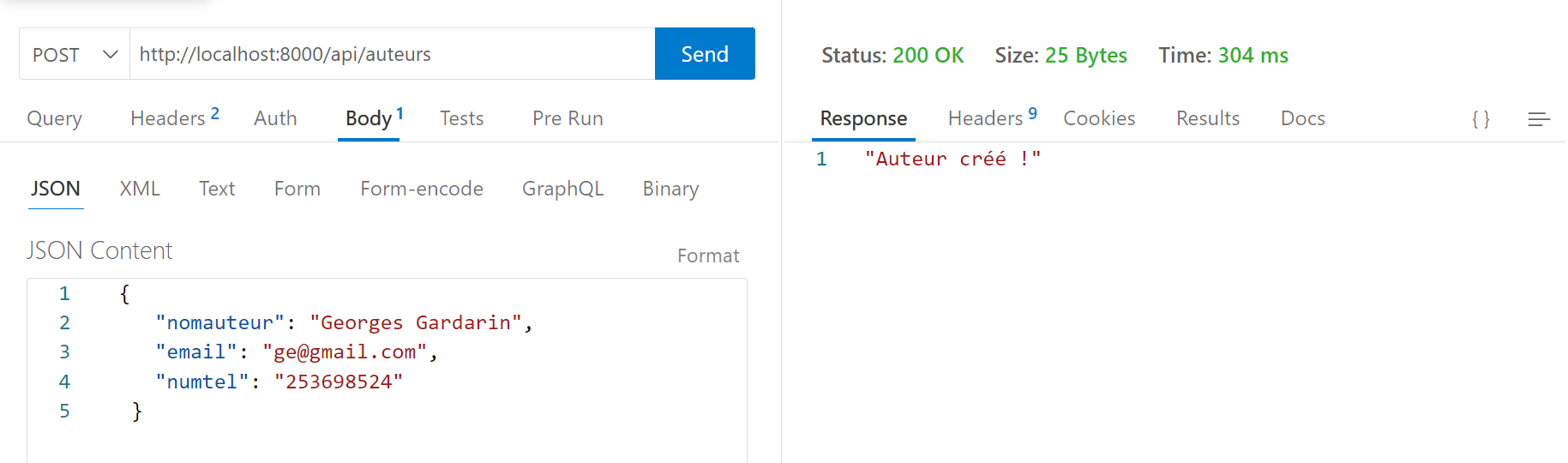
Route::middleware('api')->group(function () {

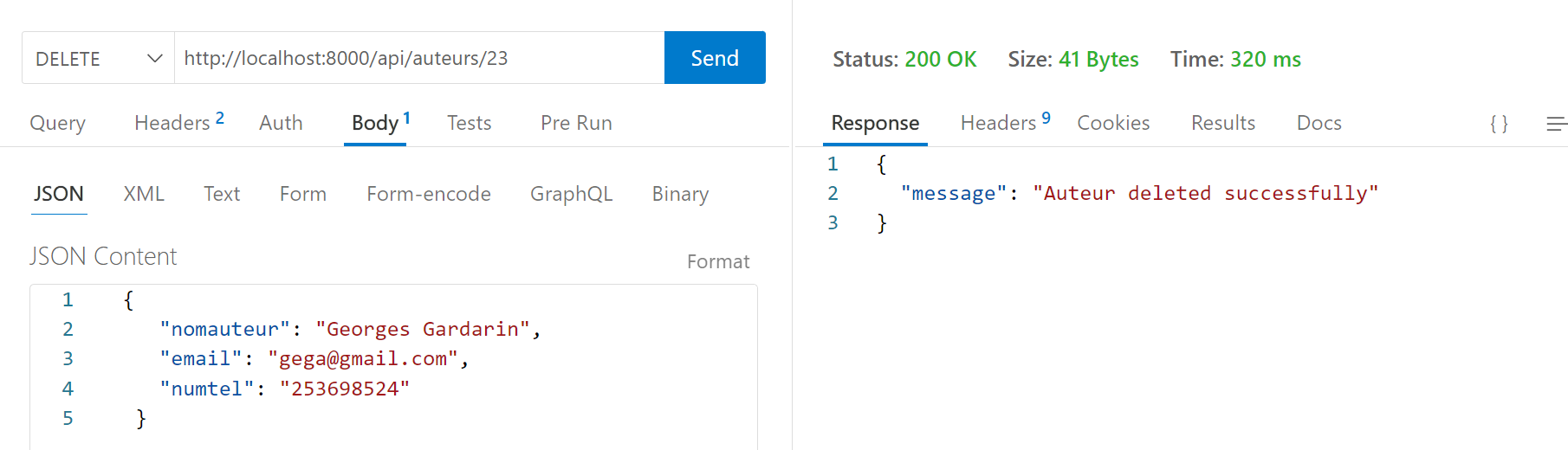
    Route::resource('auteurs', AuteurController::class);

});

***Tests API***







Spécialité



<?php

namespace App\Http\Controllers;

use App\Models\Specialite;

use Illuminate\Http\Request;

class SpecialiteController extends Controller

{

   /\*\*

     \* Display a listing of the resource.

     \*/

    public function index()

    {

        $specialites = Specialite::all()->toArray();

        return array\_reverse($specialites);

    }

    /\*\*

     \* Store a newly created resource in storage.

     \*/

    public function store(Request $request)

    {

        $specialite = new Specialite([

            'nomspecialite' => $request->input('nomspecialite')

            ]);

        $specialite->save();

        return response()->json('Specialite créée !');

    }

    /\*\*

     \* Display the specified resource.

     \*/

    public function show($id)

    {

        $specialite= Specialite::find($id);

        return response()->json($specialite);

    }

    /\*\*

     \* Update the specified resource in storage.

     \*/

    public function update($id, Request $request)

    {

        $specialite = Specialite::find($id);

        $specialite->update($request->all());

        return response()->json($specialite);

    }

    /\*\*

     \* Remove the specified resource from storage.

     \*/

    public function destroy($id)

    {

        $specialite = Specialite::find($id);

        $specialite->delete();

        return response()->json(['message' => 'Specialite deleted successfully']);

    }

}



use App\Http\Controllers\SpecialiteController;

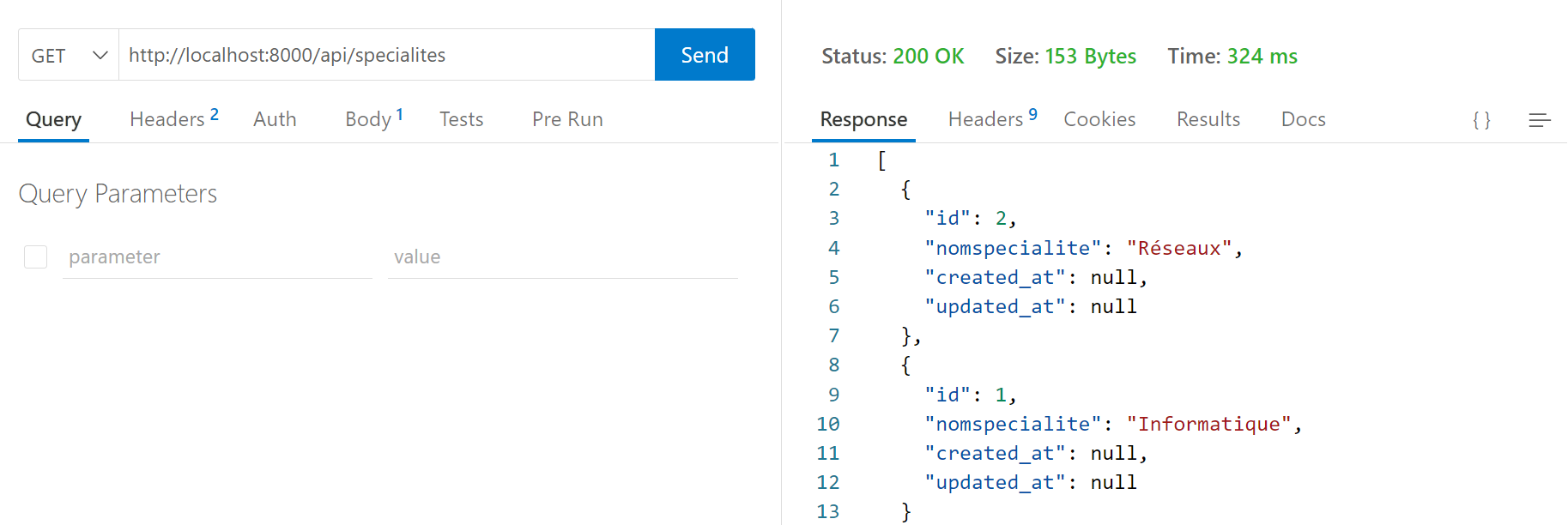
Route::middleware('api')->group(function () {

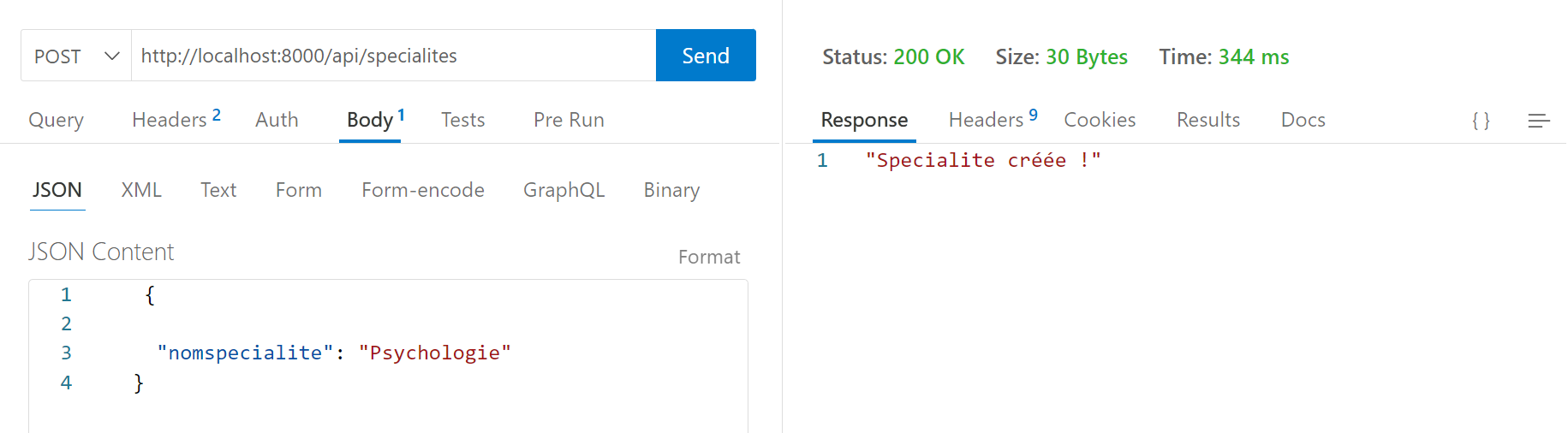
    Route::resource('specialites', SpecialiteController::class);

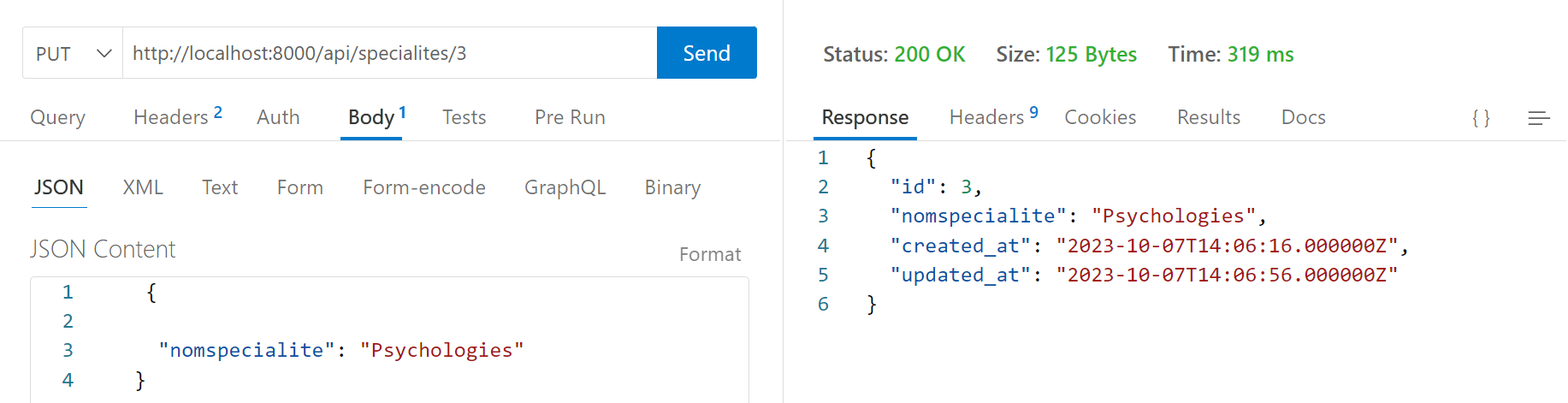
});

***Tests API***

http://localhost:8000/api/specialites







Editeur



<?php

namespace App\Http\Controllers;

use App\Models\Editeur;

use Illuminate\Http\Request;

class EditeurController extends Controller

{

   /\*\*

     \* Display a listing of the resource.

     \*/

    public function index()

    {

        $editeurs = Editeur::all()->toArray();

        return array\_reverse($editeurs);

    }

    /\*\*

     \* Store a newly created resource in storage.

     \*/

    public function store(Request $request)

    {

        $editeur = new Editeur([

            'maisonedit' => $request->input('maisonedit'),

            'siteweb' => $request->input('siteweb'),

            'email' => $request->input('email')

            ]);

        $editeur->save();

        return response()->json('Editeur créé !');

    }

    /\*\*

     \* Display the specified resource.

     \*/

    public function show($id)

    {

        $editeur= Editeur::find($id);

        return response()->json($editeur);

    }

    /\*\*

     \* Update the specified resource in storage.

     \*/

    public function update($id, Request $request)

    {

        $editeur = Editeur::find($id);

        $editeur->update($request->all());

        return response()->json($editeur);

    }

    /\*\*

     \* Remove the specified resource from storage.

     \*/

    public function destroy($id)

    {

        $editeur = Editeur::find($id);

        $editeur->delete();

        return response()->json(['message' => 'Editeur deleted successfully']);

    }

}



use App\Http\Controllers\EditeurController;

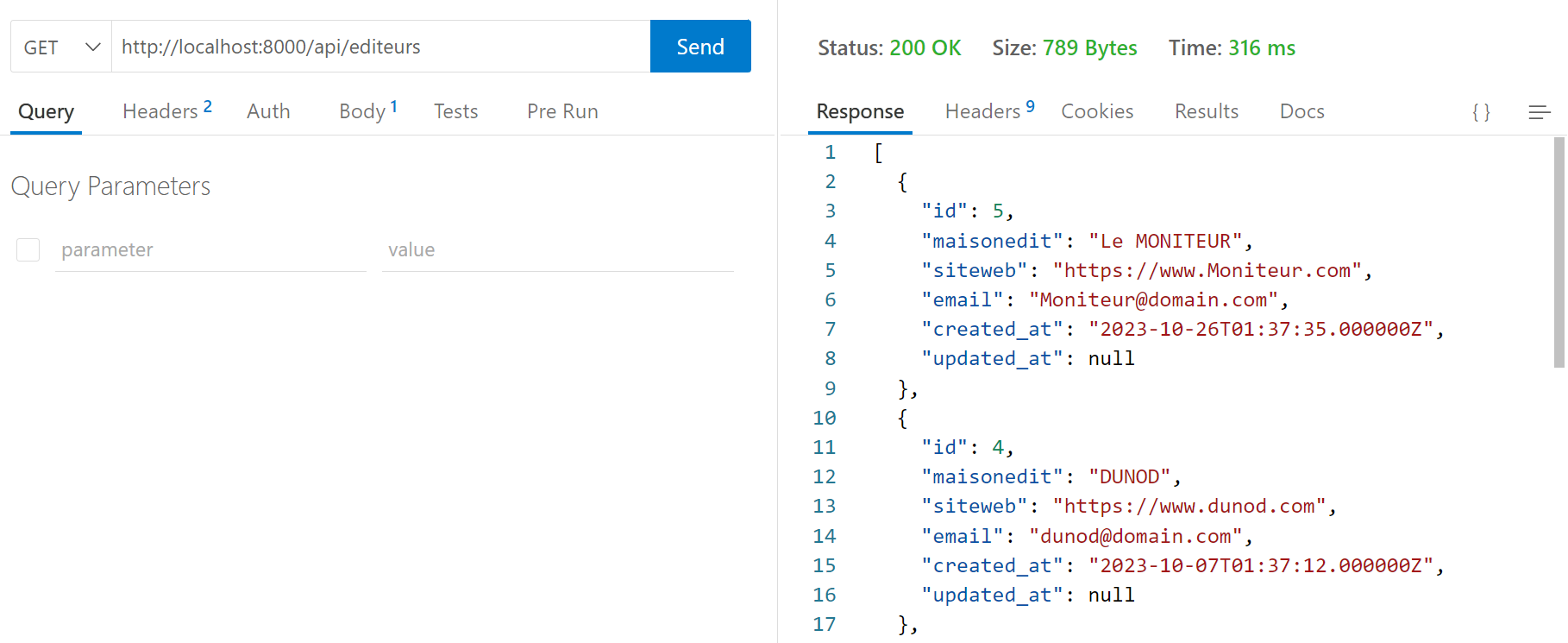
Route::middleware('api')->group(function () {

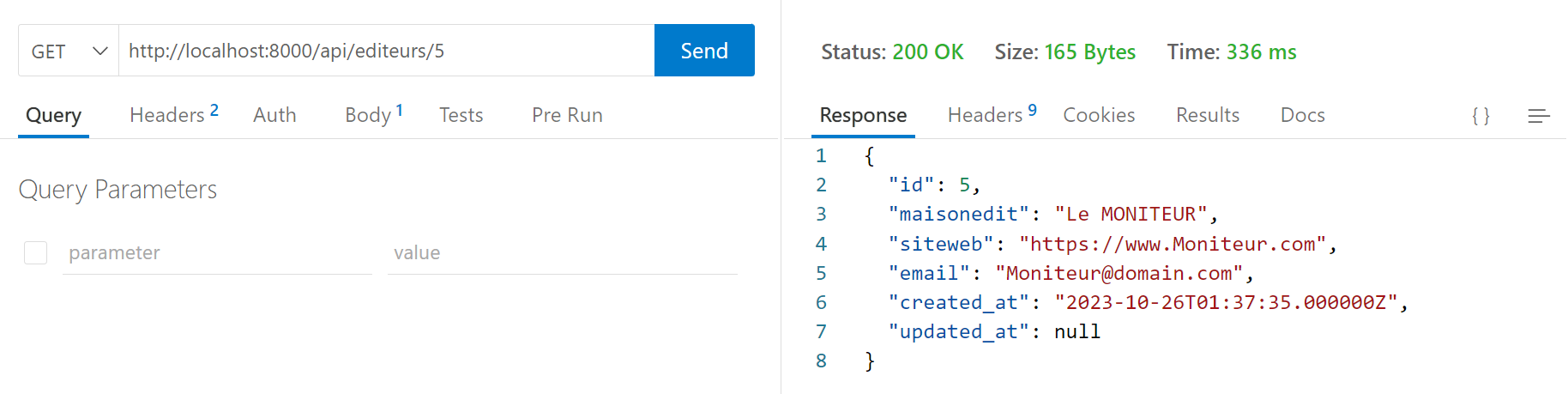
    Route::resource('editeurs', EditeurController::class);

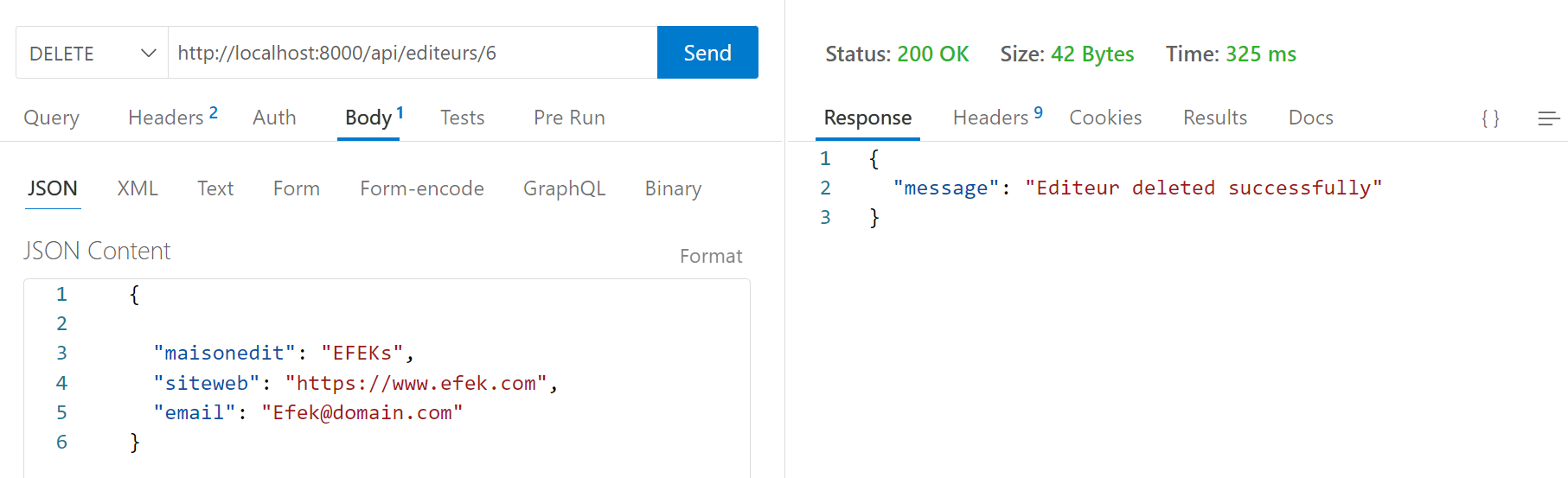
});

***Tests API***

<http://localhost:8000/api/editeurs>







Livre



<?php

namespace App\Http\Controllers;

use App\Models\Livre;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\DB;

class LivreController extends Controller

{

    /\*\*

     \* Display a listing of the resource.

     \*/

    public function index()

    {

        $livres = Livre::with('editeur','specialite','auteurs')->get()->toArray();

        /\*

 (&) pour modifier directement chaque élément du tableau "$livres".

 Cela permet d'accéder à la relation "auteurs" pour chaque livre à l'intérieur de la boucle foreach.

        \*/

        foreach ($livres as &$livre) {

               $auteurs = $livre['auteurs'];

        }

        return array\_reverse($livres);

    }

    /\*\*

     \* Store a newly created resource in storage.

     \*/

    public function store(Request $request)

    {

        $livre = new Livre([

            'isbn' => $request->input('isbn'),

            'titre' => $request->input('titre'),

            'annedition' => $request->input('annedition'),

            'prix' => $request->input('prix'),

            'qtestock' => $request->input('qtestock'),

            'couverture' => $request->input('couverture'),

            'specialite\_id' => $request->input('specialite\_id'),

            'editeur\_id' => $request->input('editeur\_id')

            ]);

        $livre->save();

                // On ajoute les auteurs associés au livre en utilisant attach

                $auteurIds = $request->input('auteur\_ids'); // On aura un champ "auteur\_ids" dans le formulaire

                $livre->auteurs()->attach($auteurIds);

                return response()->json('Livre créé !');

    }

    /\*\*

     \* Display the specified resource.

     \*/

    public function show($id)

    {

        $livre = Livre::with('editeur', 'specialite', 'auteurs')->find($id);

        if (!$livre) {

            return response()->json(['message' => 'Livre introuvable'], 404);

        }

        return response()->json($livre);

    }

    /\*\*

     \* Update the specified resource in storage.

     \*/

    public function update($id, Request $request)

    {

        $livre = Livre::find($id);

        $livre->update($request->all());

         // supprimer l'enregistrement correspondant dans la table "livre\_auteur"

        DB::table('livre\_auteur')->where('livre\_id', $livre->id)->delete();

        //puis le recréer

        $auteurIds = $request->input('auteur\_ids'); // On aura un champ "auteur\_ids" dans le formulaire

        $livre->auteurs()->attach($auteurIds);

        return response()->json($livre);

    }

    /\*\*

     \* Remove the specified resource from storage.

     \*/

    public function destroy($id)

    {

   $livre = Livre::find($id);

   if (!$livre) {

       return response()->json(['message' => 'Livre not found'], 404);

   }

   // Supprimer l'enregistrement correspondant dans la table "livre\_auteur"

   DB::table('livre\_auteur')->where('livre\_id', $livre->id)->delete();

   $livre->delete();

   return response()->json(['message' => 'Livre deleted successfully']);

    }

}

***Tests API***

<http://localhost:8000/api/livres>

