Práctica UD-8 Sesión-1

|  |  |  |  |
| --- | --- | --- | --- |
| Objetivos | * Aprender a crear una aplicación web con Laravel que utilice **jQuery** y haga uso de llamadas **AJAX**. | | |
|
| Instrucciones de entrega | * No hay que entregar nada. | | |

* Abre una línea de comandos desde **VS Code** y ve a la ruta donde está el directorio ***htdocs*** de Xampp (**c:\xampp\htdocs**) después crea un nuevo proyecto con nombre ***PHPPracticaUD8Sesion1***:

|  |
| --- |
| C:\xampp\htdocs>laravel new PHPPracticaUD8Sesion1 |

* O como alternativa usando *composer*:

|  |
| --- |
| C:\xampp\htdocs>composer create-project --prefer-dist laravel/laravel PHPPracticaUD8Sesion1 |

* Una vez creado carga el proyecto desde ***VS Code*** simplemente abriendo la carpeta.

## Laravel y JS + Ajax (jQuery)

* Tutorial de referencia:
  + <https://xpertphp.com/laravel-7-crud-operation-with-ajax-example/>
* Lo primero que haremos será crear un modelo con nombre ***Student*** junto con sus ficheros asociados del ***controlador de tipo resource, la migración, la siembra y la factoria*** todo desde una sóla línea de comandos:
  + *php artisan make:model Student -a --resource*
* Comenzamos editando el fichero de **migración** situado en ***database/migrations/xxx\_\_create\_students\_table.php*** con el siguiente contenido:

public function up()

{

Schema::create('students', function (Blueprint $table) {

$table->id();

$table->string('first\_name');

$table->string('last\_name');

$table->text('address');

$table->timestamps();

});

}

* Ahora editamos el fichero del **modelo** para señalar los campos que serán rellenables que se corresponden casi siempre a los que hemos definido como columnas en el fichero de migración para la base de datos:

/\*\*

\* The attributes that are mass assignable.

\* @var array

\*/

protected $fillable = [

'first\_name','last\_name', 'address'

];

* Continuamos editando el fichero de la **factoría** situado en ***database/factories/StudentFactory.php***:

/\*\*

\* Define the model's default state.

\*

\* @return array

\*/

public function definition()

{

return [

'first\_name' => $this->faker->firstName,

'last\_name' => $this->faker->lastName,

'address' => $this->faker->address

];

}

* Seguidamente editamos el fichero de la **siembra** para el modelo Book situado en ***database/seeders/StudentSeeder*** :

public function run()

{

// Truncamos la tabla para comenzar con una vacía.

Student::truncate();

// Creamos 10 registros para pruebas en base de datos.

Student::factory()->count(10)->create();

}

* También editamos el fichero de la **siembra** general ***DatabaseSeeder.php*** para no tener que hacer la siembra de forma individual:

public function run()

{

$this->call([

StudentSeeder::class

]);

}

* En este punto creamos la base de datos con nombre ***phppracticaud8sesion1*** ejecutamos la **migración** junto con la **siembra** y comprobamos el resultado en la base de datos:
  + - *php artisan migrate --seed*
* Ahora vamos a completar la lógica del controlador:

class StudentController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

$data['students'] = Student::orderBy('id','desc')->paginate(5);

return view('student.list', $data);

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function create()

{

//

}

/\*\*

\* Store a newly created resource in storage.

\*

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

\* @return \Illuminate\Http\Response

\*/

public function store(Request $request)

{

$student = new Student([

'first\_name' => $request->post('txtFirstName'),

'last\_name'=> $request->post('txtLastName'),

'address'=> $request->post('txtAddress')

]);

$student->save();

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

}

/\*\*

\* Display the specified resource.

\*

\* @param \App\Models\Student $student

\* @return \Illuminate\Http\Response

\*/

public function show(Student $student)

{

//

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param \App\Models\Student $student

\* @return \Illuminate\Http\Response

\*/

public function edit($id)

{

$where = array('id' => $id);

$student = Student::where($where)->first();

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

}

/\*\*

\* Update the specified resource in storage.

\*

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

\* @param \App\Models\Student $student

\* @return \Illuminate\Http\Response

\*/

public function update(Request $request, Student $student)

{

$student = Student::find($request->post('hdnStudentId'));

$student->first\_name = $request->post('txtFirstName');

$student->last\_name = $request->post('txtLastName');

$student->address = $request->post('txtAddress');

$student->update();

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

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param \App\Models\Student $student

\* @return \Illuminate\Http\Response

\*/

public function destroy(Student $student)

{

$student->delete();

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

}

}

* Por último editaremos el fichero de rutas (***routes/web.php***) para poder llamar a los métodos en el controlador:

Route::get('student',[StudentController::class, 'index']);

Route::post('student', [StudentController::class, 'store'])->name('student.store');

Route::get('student/{id}/edit', [StudentController::class, 'edit'])->name('student.edit');

Route::post('student/update', [StudentController::class, 'update'])->name('student.update');

Route::get('student/{id}/delete', [StudentController::class, 'destroy'])->name('student.delete');

* En este punto ya solo falta crear las vistas, para ello crea una nueva carpeta en **views** con nombre **layouts** y dentro de él el fichero ***app.blade.php***:

<!DOCTYPE html>

<html lang="en">

<head>

<title>Laravel 8 Ajax CRUD Example</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.0/jquery.validate.js"></script>

</head>

<body>

<div class="container">

@yield('content')

</div>

</body>

</html>

* Por último crea una nueva carpeta en ***views*** con nombre ***student*** y dentro de él el fichero ***list.blade.php***:

@extends('layouts.app')

@section('content')

<div class="row">

<div class="col-lg-11">

<h2>Laravel 8 Ajax CRUD Example</h2>

</div>

<div class="col-lg-1">

<a class="btn btn-success" href="#" data-toggle="modal" data-target="#addModal">Add</a>

</div>

</div>

@if ($message = Session::get('success'))

<div class="alert alert-success">

<p>{{ $message }}</p>

</div>

@endif

<table class="table table-bordered" id="studentTable">

<thead>

<tr>

<th>id</th>

<th>First Name</th>

<th>Last Name</th>

<th>Address</th>

<th width="280px">Action</th>

</tr>

</thead>

<tbody>

@foreach ($students as $student)

<tr id="{{ $student->id }}">

<td>{{ $student->id }}</td>

<td>{{ $student->first\_name }}</td>

<td>{{ $student->last\_name }}</td>

<td>{{ $student->address }}</td>

<td>

<a data-id="{{ $student->id }}" class="btn btn-primary btnEdit">Edit</a>

<a data-id="{{ $student->id }}" class="btn btn-danger btnDelete">Delete</button>

</td>

</tr>

@endforeach

</tbody>

</table>

<!-- Add Student Modal -->

<div id="addModal" class="modal fade" role="dialog">

<div class="modal-dialog">

<!-- Student Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title">Add New Student</h4>

</div>

<div class="modal-body">

<form id="addStudent" name="addStudent" action="{{ route('student.store') }}" method="post">

@csrf

<div class="form-group">

<label for="txtFirstName">First Name:</label>

<input type="text" class="form-control" id="txtFirstName" placeholder="Enter First Name" name="txtFirstName">

</div>

<div class="form-group">

<label for="txtLastName">Last Name:</label>

<input type="text" class="form-control" id="txtLastName" placeholder="Enter Last Name" name="txtLastName">

</div>

<div class="form-group">

<label for="txtAddress">Address:</label>

<textarea class="form-control" id="txtAddress" name="txtAddress" rows="10" placeholder="Enter Address"></textarea>

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

<!-- Update Student Modal -->

<div id="updateModal" class="modal fade" role="dialog">

<div class="modal-dialog">

<!-- Student Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title">Update Student</h4>

</div>

<div class="modal-body">

<form id="updateStudent" name="updateStudent" action="{{ route('student.update') }}" method="post">

<input type="hidden" name="hdnStudentId" id="hdnStudentId"/>

@csrf

<div class="form-group">

<label for="txtFirstName">First Name:</label>

<input type="text" class="form-control" id="txtFirstName" placeholder="Enter First Name" name="txtFirstName">

</div>

<div class="form-group">

<label for="txtLastName">Last Name:</label>

<input type="text" class="form-control" id="txtLastName" placeholder="Enter Last Name" name="txtLastName">

</div>

<div class="form-group">

<label for="txtAddress">Address:</label>

<textarea class="form-control" id="txtAddress" name="txtAddress" rows="10" placeholder="Enter Address"></textarea>

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

<script>

$(document).ready(function () {

//Add the Student

$("#addStudent").validate({

rules: {

txtFirstName: "required",

txtLastName: "required",

txtAddress: "required"

},

messages: {

},

submitHandler: function(form) {

var form\_action = $("#addStudent").attr("action");

$.ajax({

data: $('#addStudent').serialize(),

url: form\_action,

type: "POST",

dataType: 'json',

success: function (data) {

var student = '<tr id="'+data.id+'">';

student += '<td>' + data.id + '</td>';

student += '<td>' + data.first\_name + '</td>';

student += '<td>' + data.last\_name + '</td>';

student += '<td>' + data.address + '</td>';

student += '<td><a data-id="' + data.id + '" class="btn btn-primary btnEdit">Edit</a>&nbsp;&nbsp;<a data-id="' + data.id + '" class="btn btn-danger btnDelete">Delete</a></td>';

student += '</tr>';

$('#studentTable tbody').prepend(student);

$('#addStudent')[0].reset();

$('#addModal').modal('hide');

},

error: function (data) {

}

});

}

});

//When click edit student

$('body').on('click', '.btnEdit', function () {

var student\_id = $(this).attr('data-id');

$.get('student/' + student\_id +'/edit', function (data) {

$('#updateModal').modal('show');

$('#updateStudent #hdnStudentId').val(data.id);

$('#updateStudent #txtFirstName').val(data.first\_name);

$('#updateStudent #txtLastName').val(data.last\_name);

$('#updateStudent #txtAddress').val(data.address);

})

});

// Update the student

$("#updateStudent").validate({

rules: {

txtFirstName: "required",

txtLastName: "required",

txtAddress: "required"

},

messages: {

},

submitHandler: function(form) {

var form\_action = $("#updateStudent").attr("action");

$.ajax({

data: $('#updateStudent').serialize(),

url: form\_action,

type: "POST",

dataType: 'json',

success: function (data) {

var student = '<td>' + data.id + '</td>';

student += '<td>' + data.first\_name + '</td>';

student += '<td>' + data.last\_name + '</td>';

student += '<td>' + data.address + '</td>';

student += '<td><a data-id="' + data.id + '" class="btn btn-primary btnEdit">Edit</a>&nbsp;&nbsp;<a data-id="' + data.id + '" class="btn btn-danger btnDelete">Delete</a></td>';

$('#studentTable tbody #'+ data.id).html(student);

$('#updateStudent')[0].reset();

$('#updateModal').modal('hide');

},

error: function (data) {

}

});

}

});

//delete student

$('body').on('click', '.btnDelete', function () {

var student\_id = $(this).attr('data-id');

$.get('student/' + student\_id +'/delete', function (data) {

$('#studentTable tbody #'+ student\_id).remove();

})

});

});

</script>

@endsection

* Arranca el servidor y prueba el resultado en:
  + <http://localhost:8000/student>

## 

## Ejercicios Extra

* Modifica el ejemplo para que el *backend* sea totalmente un **API Resource** como el visto en la unidad anterior.
* Modificar la plantilla ***blade*** para utilizar los ficheros JS y CSS de nuestro directorio local en lugar de llamadas a URLs públicas.