Restful API with Laravel 9

Getting Started

Objective

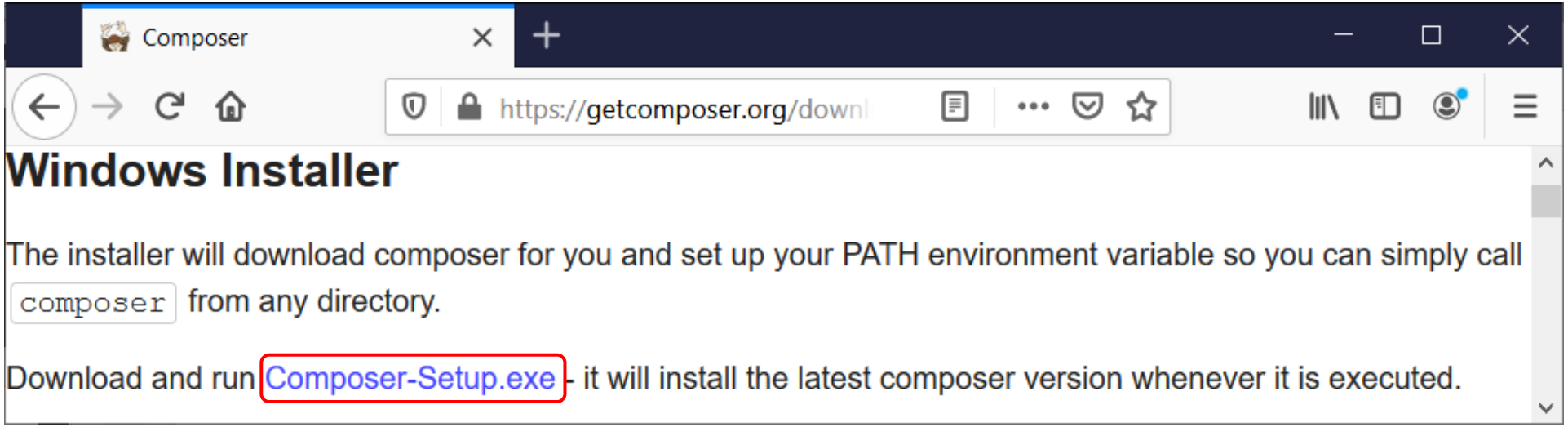
1. Laravel Introduction
2. Installing composer
3. Creating Laravel project
4. Connecting to MySQL Database
5. Creating DB Migration
6. Seeding Data
7. Working with Eloquent ORM
8. Creating Simple Restful APIs

Laravel Introduction

- What Laravel?
- Why Laravel?
- Web Server
- IDE

1. Installing composer

- Visit url: https://getcomposer.org/download/
- Download and run Composer-Setup.exe
- To test composer version
  - Run Command Prompt (cmd) window
  - Type: composer --version



2. Creating Laravel project

- In command prompt windows, move to "C:\xampp\htdocs" folder:

```
composer create-project --prefer-dist laravel/laravel laravelapi
```

- Wait for a few minutes to download and create the project

3. Connecting to MySQL Database

- Create a database named “laravelapi”.
- Open the .env file the project folder, then modify the following fields:
  - DB_DATABASE=laravelapi
  - DB_USERNAME=root
  - DB_PASSWORD=
- In command prompt window, move to project folder (ex: c:\xampp\htdocs\laravelapi), then type:

```
php artisan migrate:install
```

4. Creating DB Migration

- To create a DB Migration file, in command prompt type:

php artisan make:migration create_employees_table

- The file will be created in "database/migrations" folder (https://laravel.com/docs/9.x/migrations)
- Add the highlighted codes as follow

```php
return new class extends Migration
{
    public function up()
    {
        Schema::create('employees', function (Blueprint $table) {
            $table->bigIncrements('EMP_ID');
            $table->string('LastName', 100);
            $table->string('FirstName', 100);
            $table->string('Email', 255)->nullable();
            $table->string('Phone', 255)->nullable();
            $table->string('Image', 255)->nullable();
        });
    }
    public function down()
    { ...
    }
};
```

- Modify …_create_users_table migration file

```php
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->string('username')->unique();
        $table->string('password');
        $table->string('api_token')->nullable();
        $table->timestamps();
    });
}
public function down()
{
    Schema::dropIfExists('users');
}
```

- To run all of your outstanding migrations, execute the migrate Artisan command:

  ```
  php artisan migrate
  ```

- To roll back migrations:
  - 1 step, use

    ```
    php artisan migrate:rollback
    ```

  - All steps, use

    ```
    php artisan migrate:reset
    ```

- To roll back & migrate

  ```
  php artisan migrate:refresh --seed
  ```

5. Seeding Data

- To create a sample data for a table, in command prompt type:

  php artisan make:seeder EmployeesTableSeeder

- The file will be created in "database/seeders" folder (https://laravel.com/docs/9.x/seeding)
- Open the file named "EmployeesTableSeeder", then add the highlighted codes as follow

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class EmployeesTableSeeder extends Seeder
{
    public function run()
    {
        $employees = [
            ['LastName' => 'Trần', 'FirstName' => 'Chân', 'Email' => 'chant@yahoo.com',
                'Phone' => '0919937563'],
            ['LastName' => 'Lê Công', 'FirstName' => 'Định', 'Email' => 'dinhlc@yahoo.com',
                'Phone' => '0984633523'],
        ];
        DB::table('employees')->insert($employees);
    }
}
```

- Type: php artisan make:seeder UsersTableSeeder
- Open the file named "UsersTableSeeder", then modify

```php
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class UsersTableSeeder extends Seeder
{
    public function run()
    {
        DB::table('users')->insert([
            'name'=>'web admin',
            'email'=>'admin@yahoo.com',
            'username'=>'admin',
            'password'=>bcrypt('123456789'),
            'created_at'=>now()
        ]);
    }
}
```

- Open the file named “DatabaseSeeder”, then add the highlighted codes as below

```php
class DatabaseSeeder extends Seeder
{
    /**...
    public function run()
    {
        $this->call([
            UsersTableSeeder::class,
            EmployeesTableSeeder::class
        ]);
    }
}
```

- To execute all seeder files, type: php artisan db:seed
- To executed the specific seeder class, type:

php artisan db:seed --class=UsersTableSeeder

6. Working with Eloquent ORM

- To create a model named “Employee”, in command prompt type: php artisan make:model Employee
- The file will be created in “app/Models” folder (https://laravel.com/docs/9.x/eloquent)
- Add the highlighted codes as below

```php
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;

class Employee extends Model
{
    protected $table = 'employees';
    protected $primaryKey = 'EMP_ID';

    public $timestamps = false;
}
```

7. Creating Simple Restful APIs

7.1 Creating a new controller

7.2 Displaying data from resource (GET)

7.3 Storing a newly created resource (POST)

7.4 Updating the specified resource (PUT)

7.5 Removing the specified resource (DELETE)

7.1 Creating a new controller

- To create a controller, in command prompt type:

  php artisan make:controller EmployeesController

- The file will be created in "app/Http/Controllers" folder (https://laravel.com/docs/9.x/controllers)

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class EmployeesController extends Controller
{
    //
}
```

7.2 Displaying data from resource

- Open the file named "EmployeesController.php", then add highlighted codes as below

```php
use App\Models\Employee;
use Illuminate\Http\Request;

class EmployeesController extends Controller
{
    public function index($id = null) {
        if ($id == null) {
            return Employee::orderBy('FirstName', 'asc')->get();
        } else {
            $data = Employee::find($id);
            if ($data) {
                return $data;
            } else {
                return response()->json(['message' => 'Data not found.'], 404);
            }
        }
    }
}
```

- Open the file named “api.php” in “routes” folder, then add the highlighted codes below for define a new route.

```php
<?php

use App\Http\Controllers\EmployeesController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/
Route::get('employees/{id?}',[EmployeesController::class, 'index']);
```

Testing

- GET: http://localhost/laravelapi/public/api/employees


```
[
    {
        "EMP_ID": 1,
        "LastName": "Trần",
        "FirstName": "Chân",
        "Email": "chant@yahoo.com",
        "Phone": "0919937563",
        "Image": null
    },
    {
        "EMP_ID": 2,
        "LastName": "Lê Công",
        "FirstName": "Định",
        "Email": "dinhlc@yahoo.com",
        "Phone": "0984633523",
        "Image": null
    }
]
```

- GET: http://localhost/laravelapi/public/api/employees/1


```
{
    "EMP_ID": 1,
    "LastName": "Trần",
    "FirstName": "Chân",
    "Email": "chant@yahoo.com",
    "Phone": "0919937563",
    "Image": null
}
```

7.3 Storing a newly created resource

- In "EmployeesController", add "create" method with highlighted codes

```php
use Throwable;

class EmployeesController extends Controller
{
    public function index($id = null) {…
    }
    public function create(Request $req) {
        try {
            $data = new Employee();
            $data->LastName = $req->input('lastName');
            $data->FirstName = $req->input('firstName');
            $data->Phone = $req->input('phone');
            $data->Email = $req->input('email');
            $data->save();
            return $data;
        } catch(Throwable $e) {
            return response()->json(['message' => $e->getMessage()], 500);
        }
    }
}
```

- In file named “routes/api.php”, add a new route

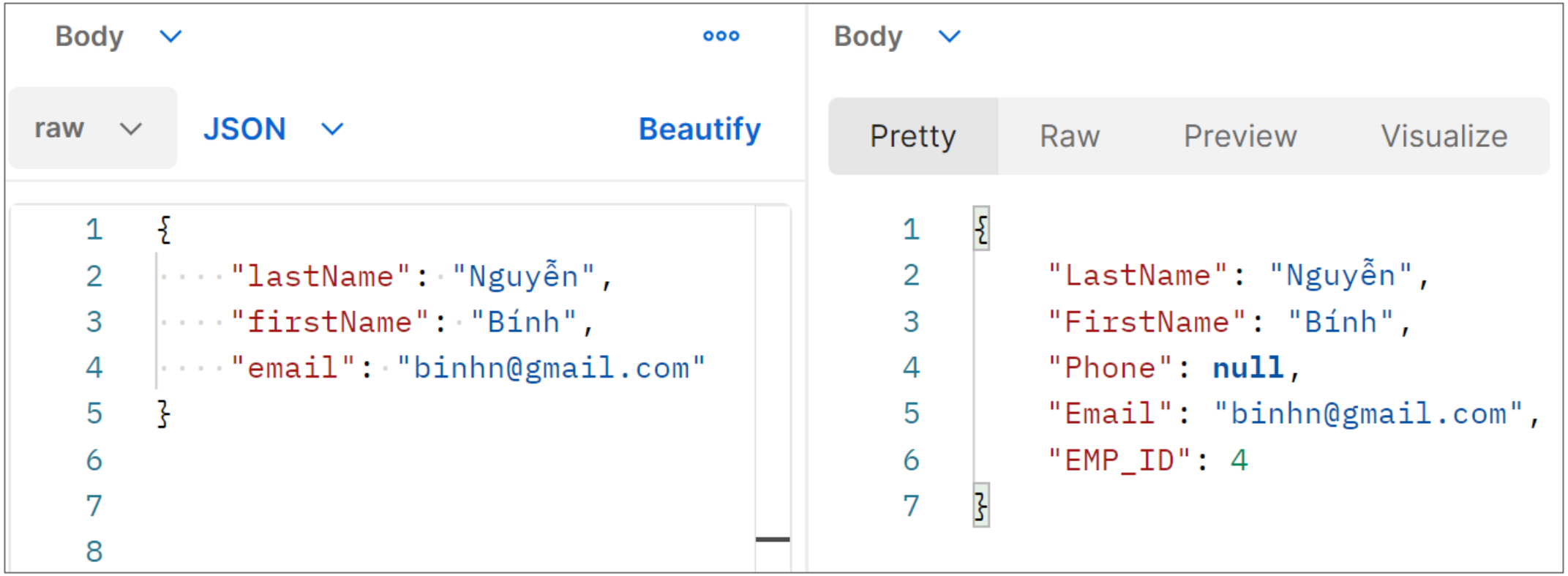
```php
<?php

use App\Http\Controllers\EmployeesController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/
Route::get('employees/{id?}',[EmployeesController::class, 'index']);
Route::post('employees',[EmployeesController::class, 'create']);
```

Testing

- POST: http://localhost/laravelapi/public/api/employees



7.4 Updating the specified resource

- In "EmployeesController", add "update" method with highlighted codes

```php
public function update($id, Request $req) {
    $data = Employee::find($id);
    if ($data) {
        try {
            $data->LastName = $req->input('lastName');
            $data->FirstName = $req->input('firstName');
            $data->Phone = $req->input('phone');
            $data->Email = $req->input('email');
            $data->save();
            return $data;
        } catch(Throwable $e) {
            return response()->json(['message' => $e->getMessage()], 500);
        }
    } else {
        return response()->json(['message' => 'Data not found'], 404);
    }
}
```

- In file named “routes/api.php”, add a new route

```php
<?php

use App\Http\Controllers\EmployeesController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/
Route::get('employees/{id?}',[EmployeesController::class, 'index']);
Route::post('employees',[EmployeesController::class, 'create']);
Route::put('employees/{id}',[EmployeesController::class, 'update']);
```

Testing

- PUT: http://localhost/laravelapi/public/api/employees/4

raw JSON Beautify

```
{
    "lastName": "Nguyễn",
    "firstName": "Bính",
    "email": "binhn@yahoo.com",
    "phone": "038447432"
}
```

Pretty Raw Preview Visualize

```
{
    "EMP_ID": 4,
    "LastName": "Nguyễn",
    "FirstName": "Bính",
    "Email": "binhn@yahoo.com",
    "Phone": "038447432",
    "Image": null
}
```

7.5 Removing the specified resource

- In "EmployeesController", add "delete" method with highlighted codes

```php
public function delete($id) {
    $data = Employee::find($id);
    if ($data) {
        try {
            $data->delete();
            return $data;
        } catch(Throwable $e) {
            return response()->json(['message' => $e->getMessage()], 500);
        }
    } else {
        return response()->json(['message' => 'Data not found'], 404);
    }
}
```

- In file named “routes/api.php”, add a new route

```php
<?php

use App\Http\Controllers\EmployeesController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/
Route::get('employees/{id?}',[EmployeesController::class, 'index']);
Route::post('employees',[EmployeesController::class, 'create']);
Route::put('employees/{id}',[EmployeesController::class, 'update']);
Route::delete('employees/{id}',[EmployeesController::class, 'delete']);
```

Testing

- DELETE: http://localhost/laravelapi/public/api/employees/4