Laravel

Outlines of Discussion

- Accessors in Laravel
- Mutators in Laravel
- Relationship in Laravel
- API in laravel
- Types of APIs in Laravel
- API Methods is Laravel
- API Validation

Accessors in Laravel

- In Laravel, accessors are special methods used to transform attributes in your model whenever they are accessed. This feature is particularly useful when you want to customize the way certain data is presented or format it before it is retrieved.

How Accessors Work

How Accessors Work

Accessors follow a specific naming convention:

- The method name should be `get{Attribute}Attribute`.
- `{Attribute}` is the name of the attribute you want to modify or format.

For example, if you have a `name` column in your database, the accessor to modify it would be named `getNameAttribute`.

Defining Accessors

Defining Accessors

To define an accessor, add it to your Eloquent model. For instance, let's say you have a `User` model with `first_name` and `last_name` attributes, and you want to create a `full_name` attribute.

Example: Full Name Accessor

php

```php
class User extends Model
{
    public function getFullNameAttribute()
    {
        return "{$this->first_name} {$this->last_name}";
    }
}
```

Now, you can access `full_name` as if it were a property:

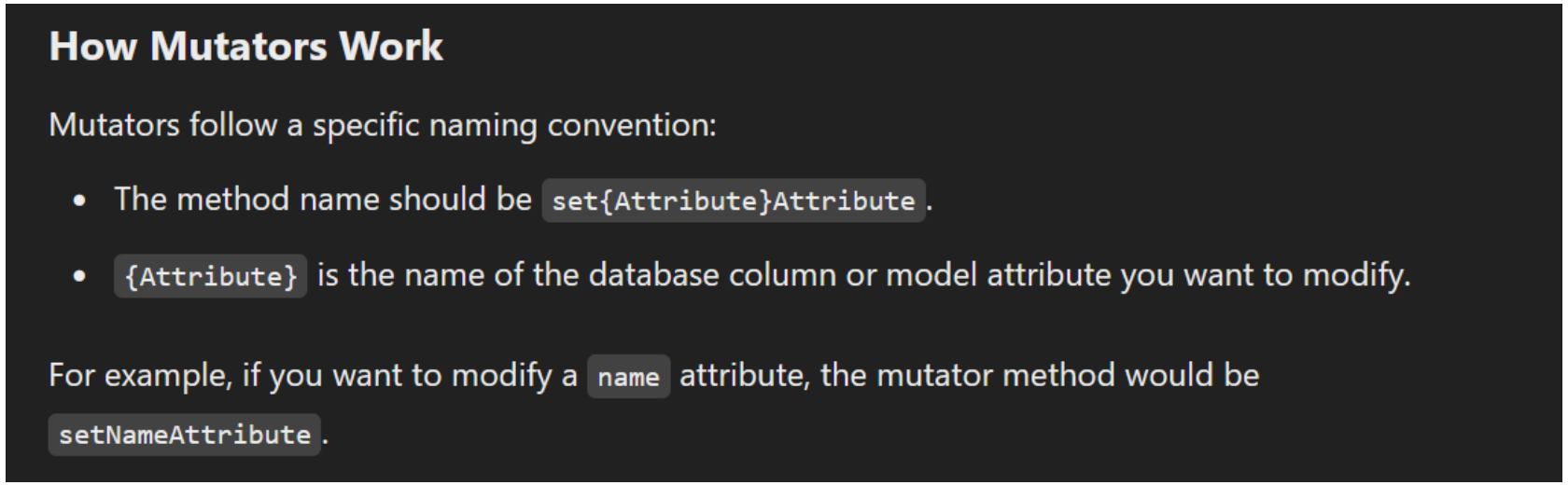
Copy code

php

```php
$user = User::find(1);
echo $user->full_name; // Outputs: "John Doe" (if first_name is John and last_name is Doe)
```

Mutators in Laravel

- In Laravel, mutators are methods that allow you to modify the value of an attribute before it is saved to the database. They are the counterpart to accessors, which format data when retrieving it.



How Mutators Work

Mutators follow a specific naming convention:

- The method name should be `set{Attribute}Attribute`.
- `{Attribute}` is the name of the database column or model attribute you want to modify.

For example, if you want to modify a `name` attribute, the mutator method would be `setNameAttribute`.

Defining Mutators

Defining Mutators

You define a mutator in your model class. The value you set in the mutator method will be automatically applied whenever the attribute is assigned.

Example: Storing Name in Lowercase

php Copy code

```php
class User extends Model
{
    public function setNameAttribute($value)
    {
        $this->attributes['name'] = strtolower($value);
    }
}
```

Cont. ...

Now, whenever you assign a value to `name`, it will be converted to lowercase before being saved:

php

Copy code

```php
$user = new User();
$user->name = 'John DOE';
echo $user->name; // Outputs: "john doe"
```

Cont. ...

Key Points to Remember

1. **Naming Convention**: Mutators use `set{Attribute}Attribute`.
2. **Attribute Transformation**: Mutators modify values before they are stored in the database.
3. **No Impact on Retrieval**: Mutators do not affect how attributes are retrieved. For retrieval transformations, use **accessors**.
4. **Used with Input Validation**: Mutators are often paired with Laravel's validation rules to ensure data integrity.
5. **Automatic Application**: Mutators are automatically applied whenever you assign a value to the corresponding attribute.

Relationship in Laravel

- In Laravel, a **relationship** refers to the connection between database tables, enabling you to interact with related data efficiently.
- Relationships are defined in **Eloquent models**, which map your database tables to PHP classes. By defining relationships, you can fetch related data without writing complex SQL queries.

Types of Relationships in Laravel

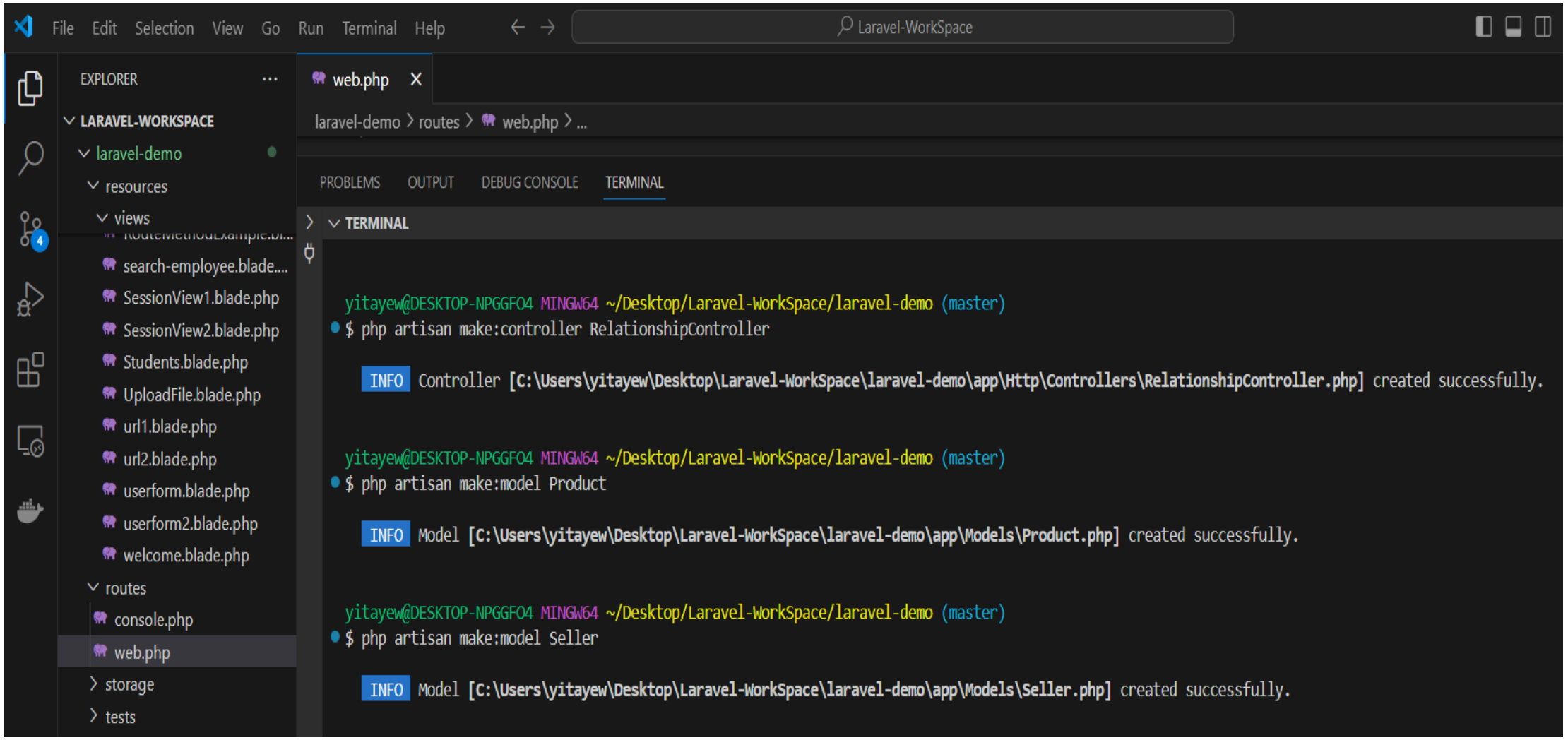
1. One-to-One Relationship

- **Description**: One record in a table is associated with one record in another table.
- **Example**: A user has one profile.

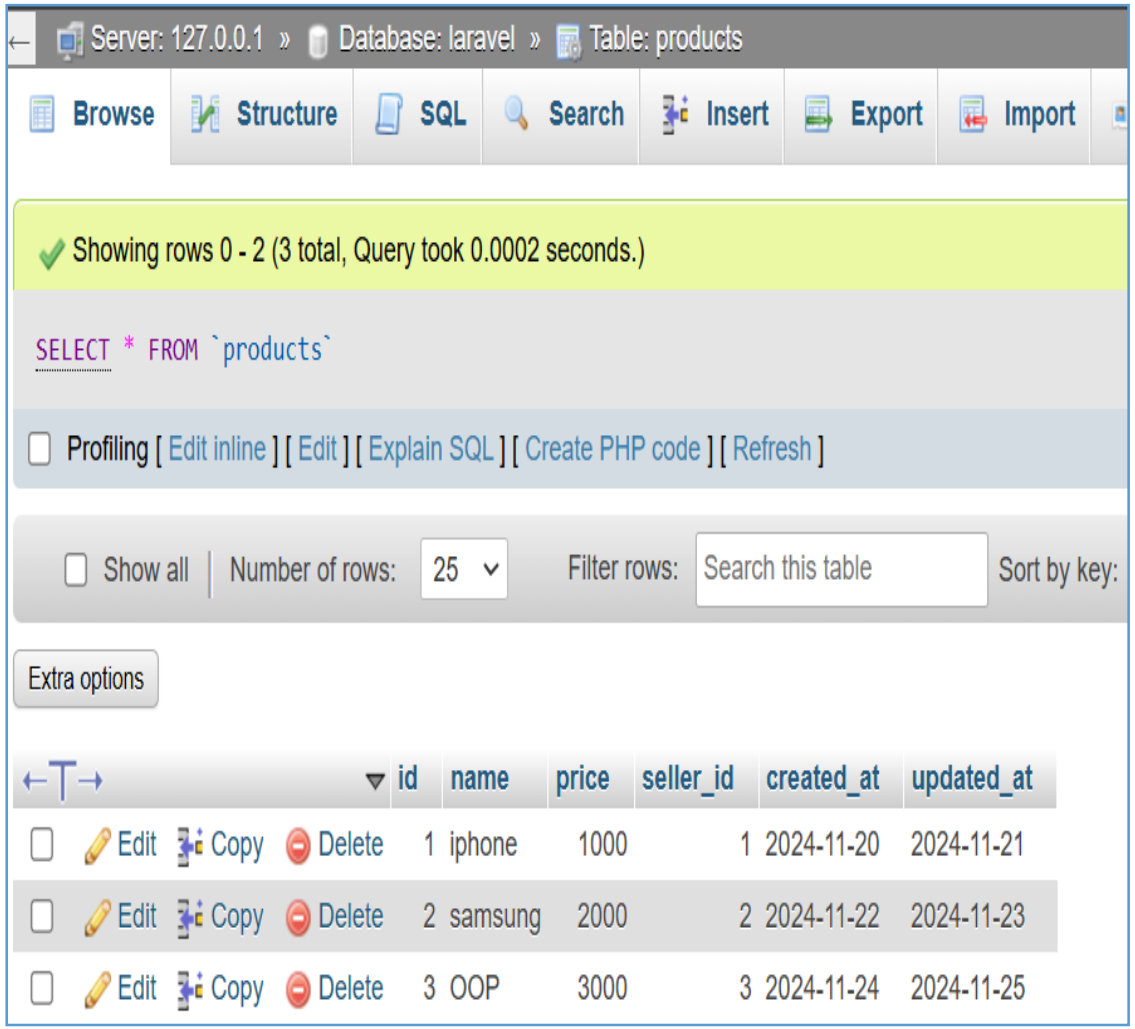
Table Structure:

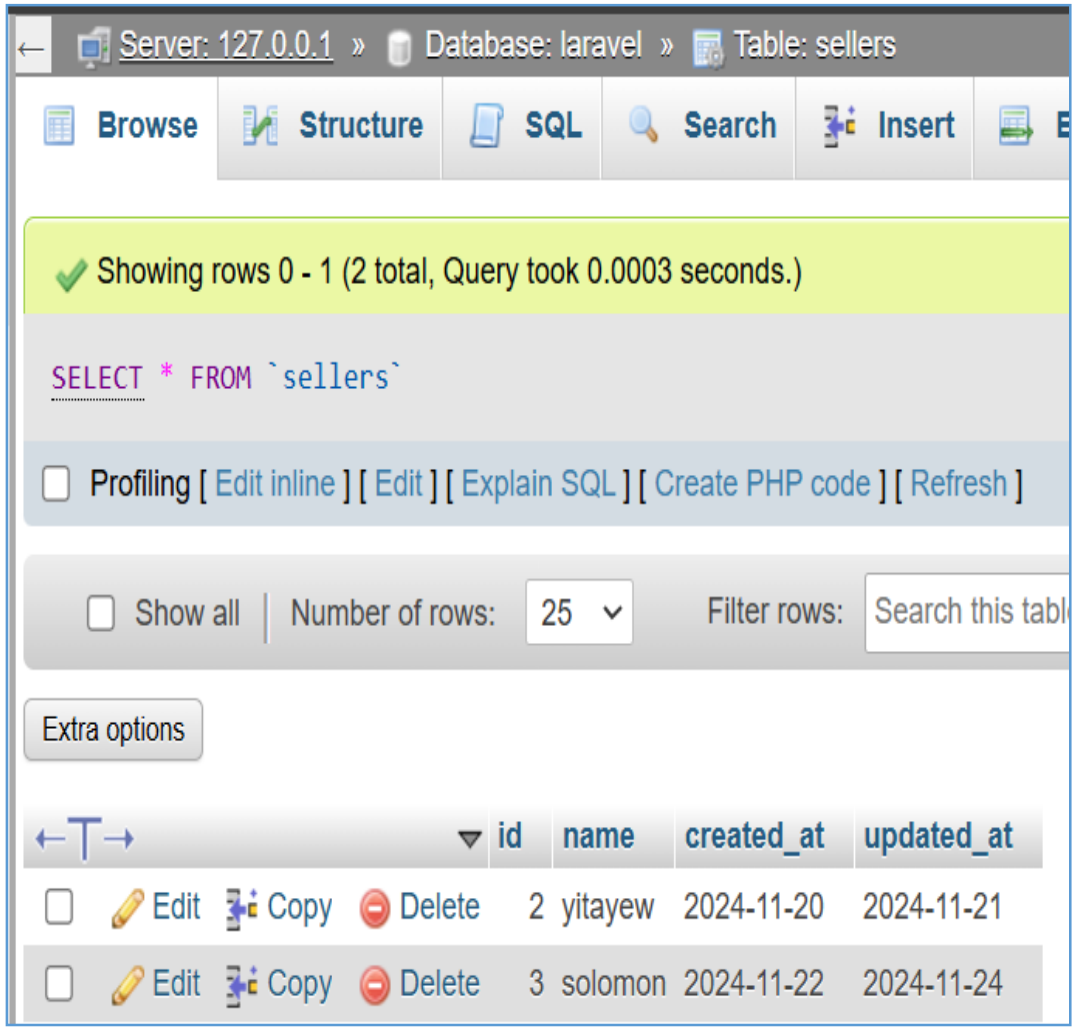
Users Table	Profiles Table
`id` (PK)	`id` (PK)
`name`	`user_id` (FK)
	`bio`

Example (One-to-One Relationship)

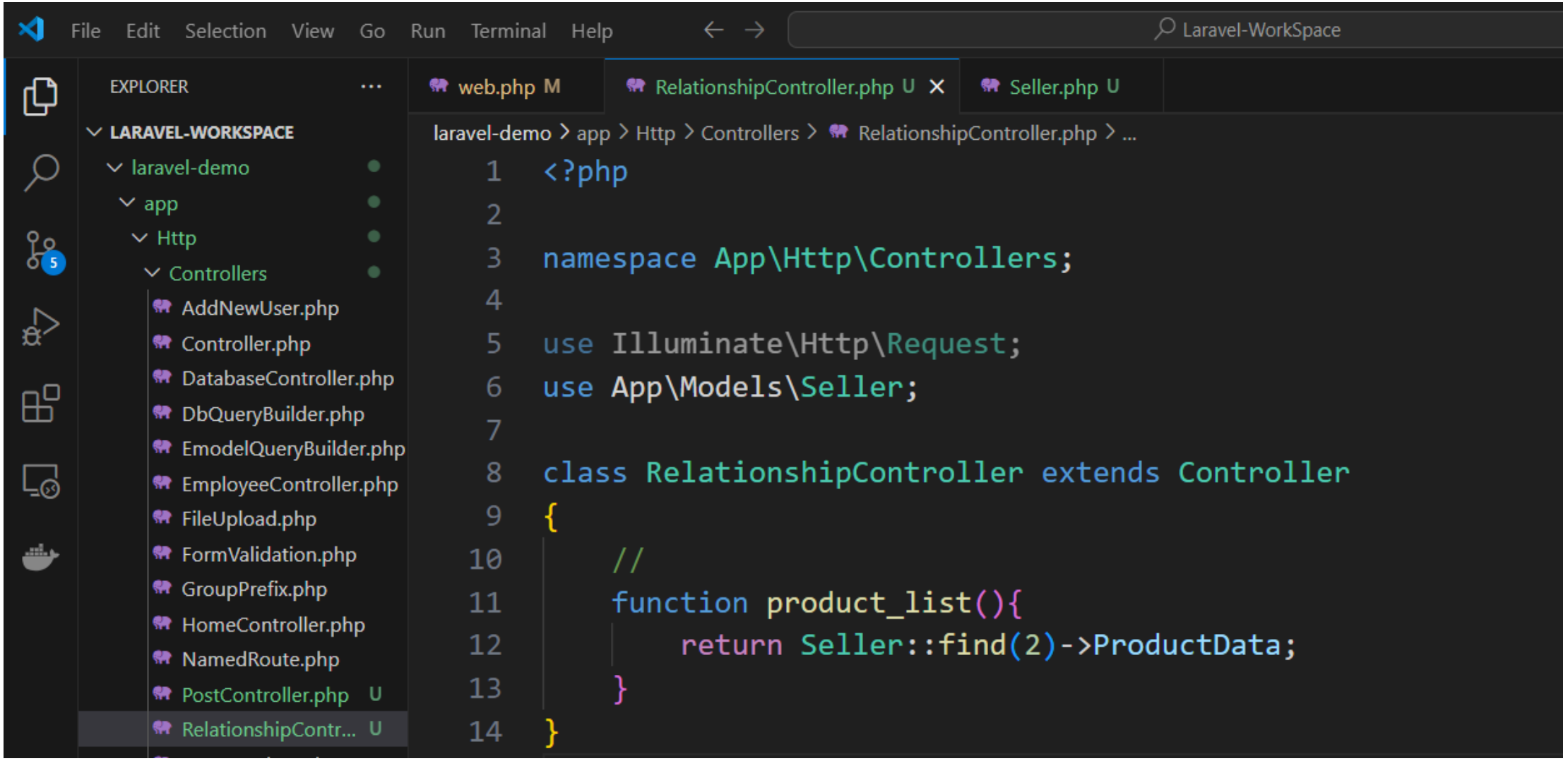


Databases

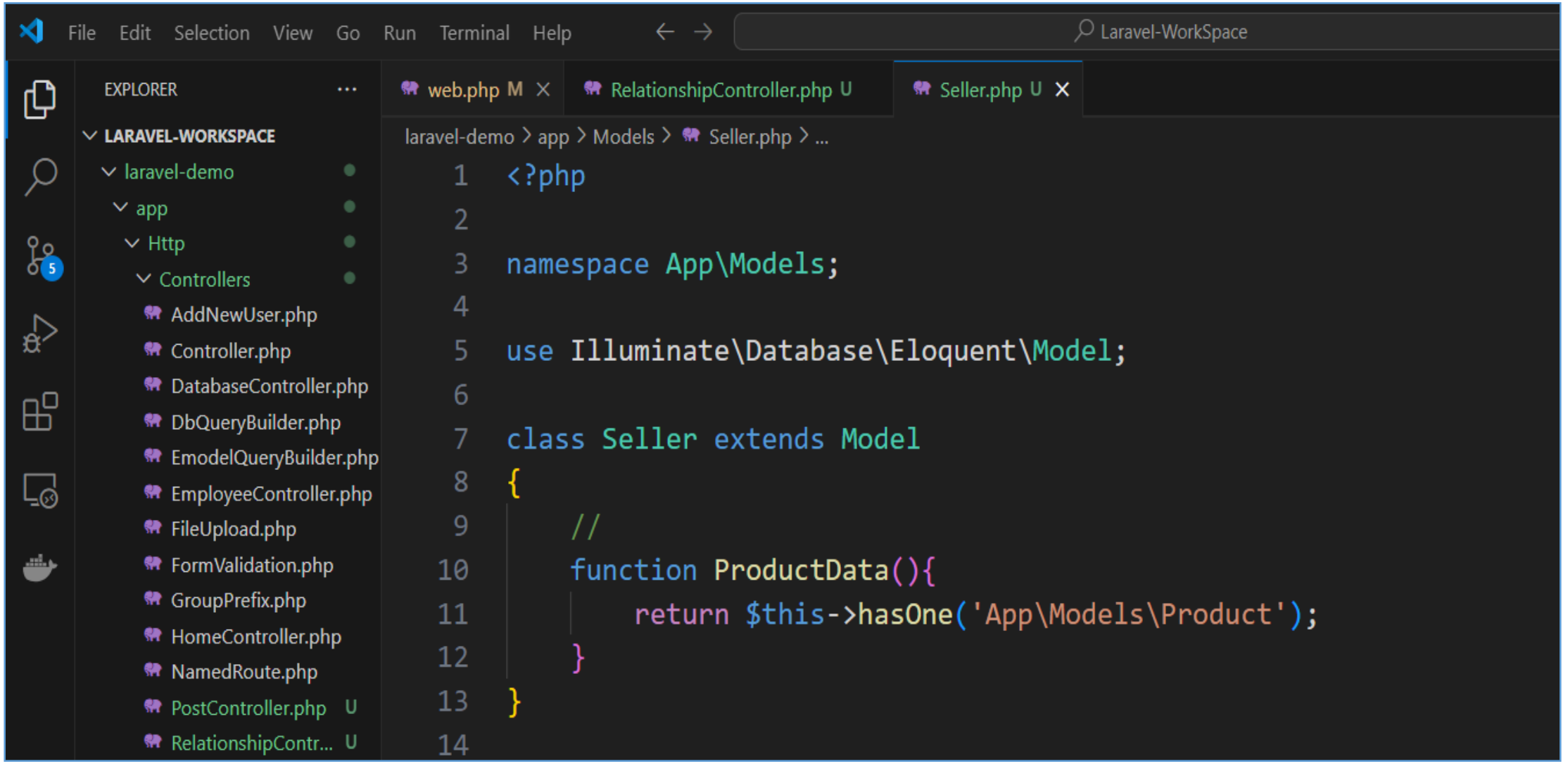

Server: 127.0.0.1 » Database: laravel » Table: products
Browse
Structure
SQL
Search
Insert
Export
Import
Showing rows 0 - 2 (3 total, Query took 0.0002 seconds.)
SELECT * FROM `products`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key:
Extra options
id name price seller_id created_at updated_at
Edit Copy Delete 1 iphone 1000 1 2024-11-20 2024-11-21
Edit Copy Delete 2 samsung 2000 2 2024-11-22 2024-11-23
Edit Copy Delete 3 OOP 3000 3 2024-11-24 2024-11-25


Server: 127.0.0.1 » Database: laravel » Table: sellers
Browse
Structure
SQL
Search
Insert
Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)
SELECT * FROM `sellers`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows:
Extra options
id name created_at updated_at
Edit Copy Delete 2 yitayew 2024-11-20 2024-11-21
Edit Copy Delete 3 solomon 2024-11-22 2024-11-24

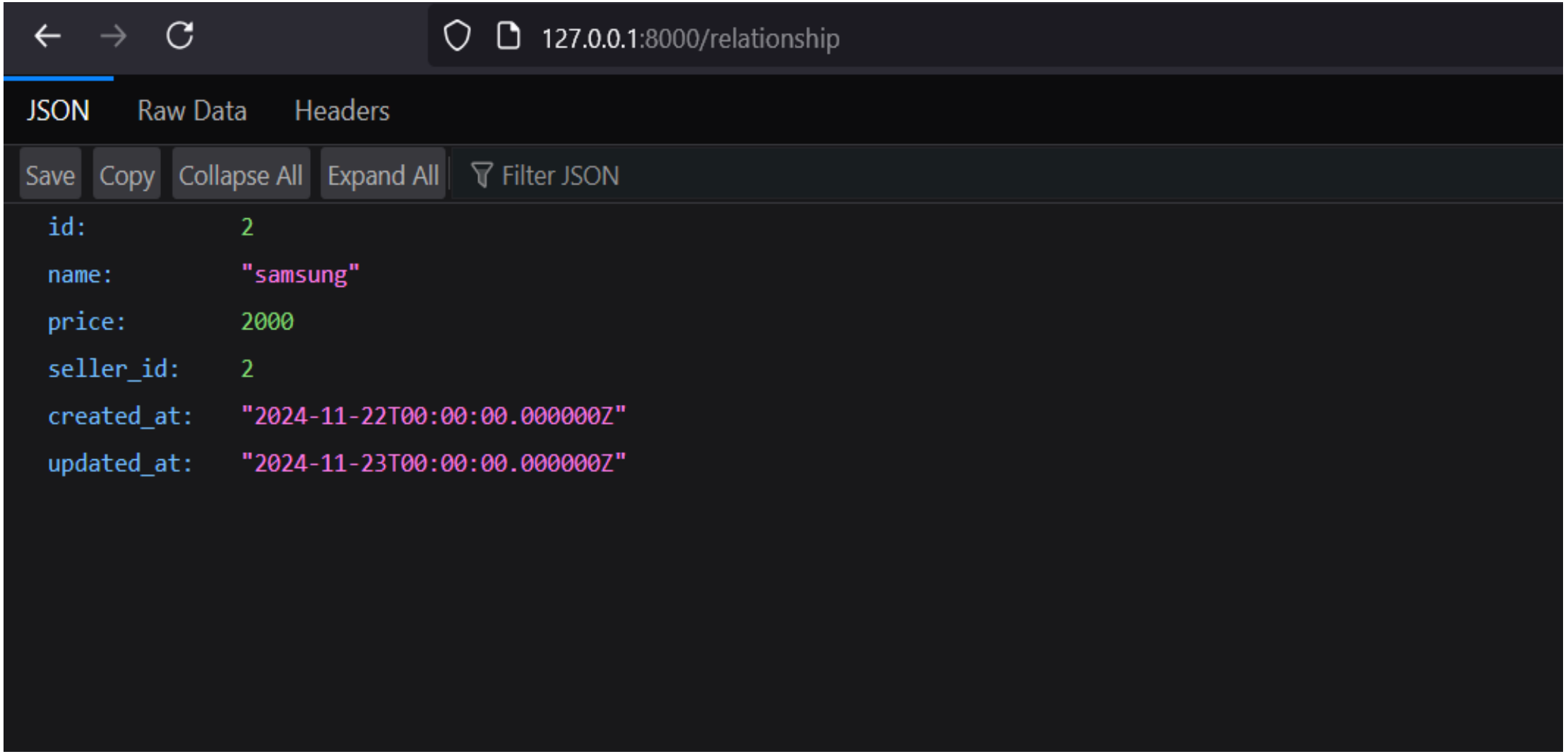
Controller (RelationshipController)



Model (Seller)



Output


127.0.0.1:8000/relationship
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON
id: 2
name: "samsung"
price: 2000
seller_id: 2
created_at: "2024-11-22T00:00:00.000000Z"
updated_at: "2024-11-23T00:00:00.000000Z"

One to Many Relationship

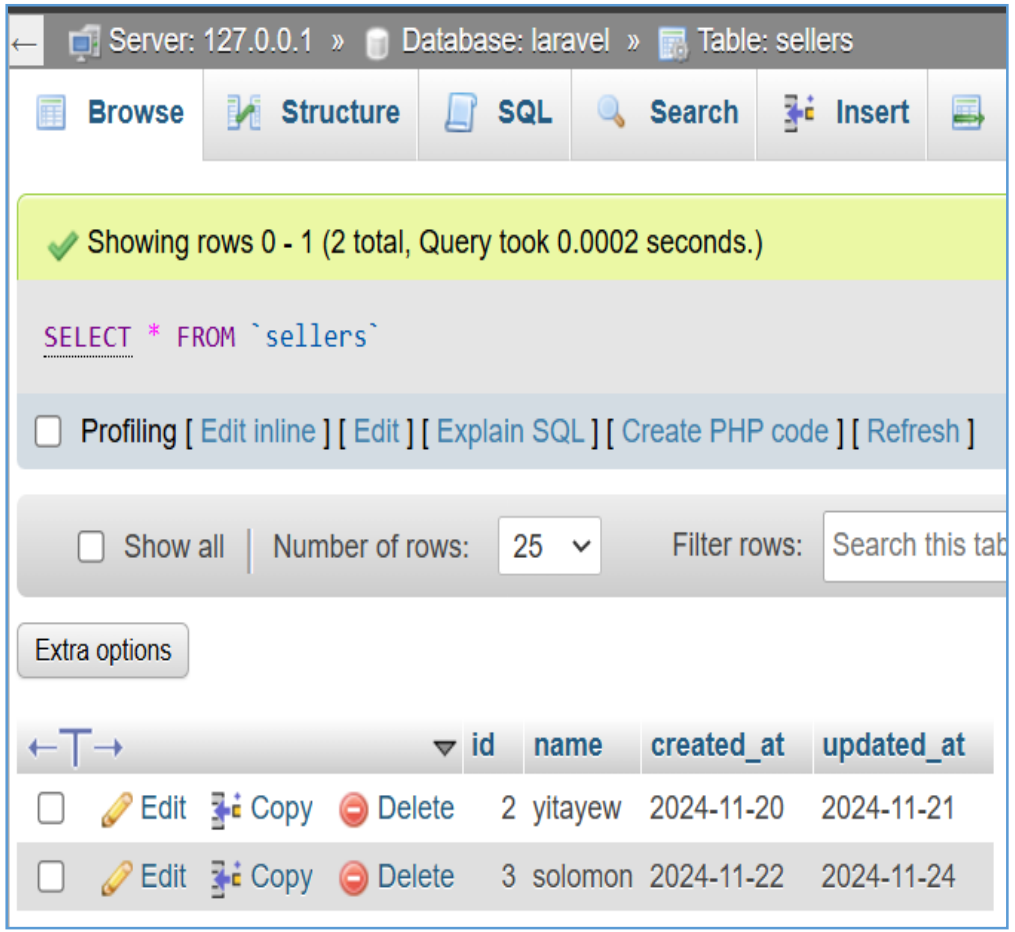
2. One-to-Many Relationship

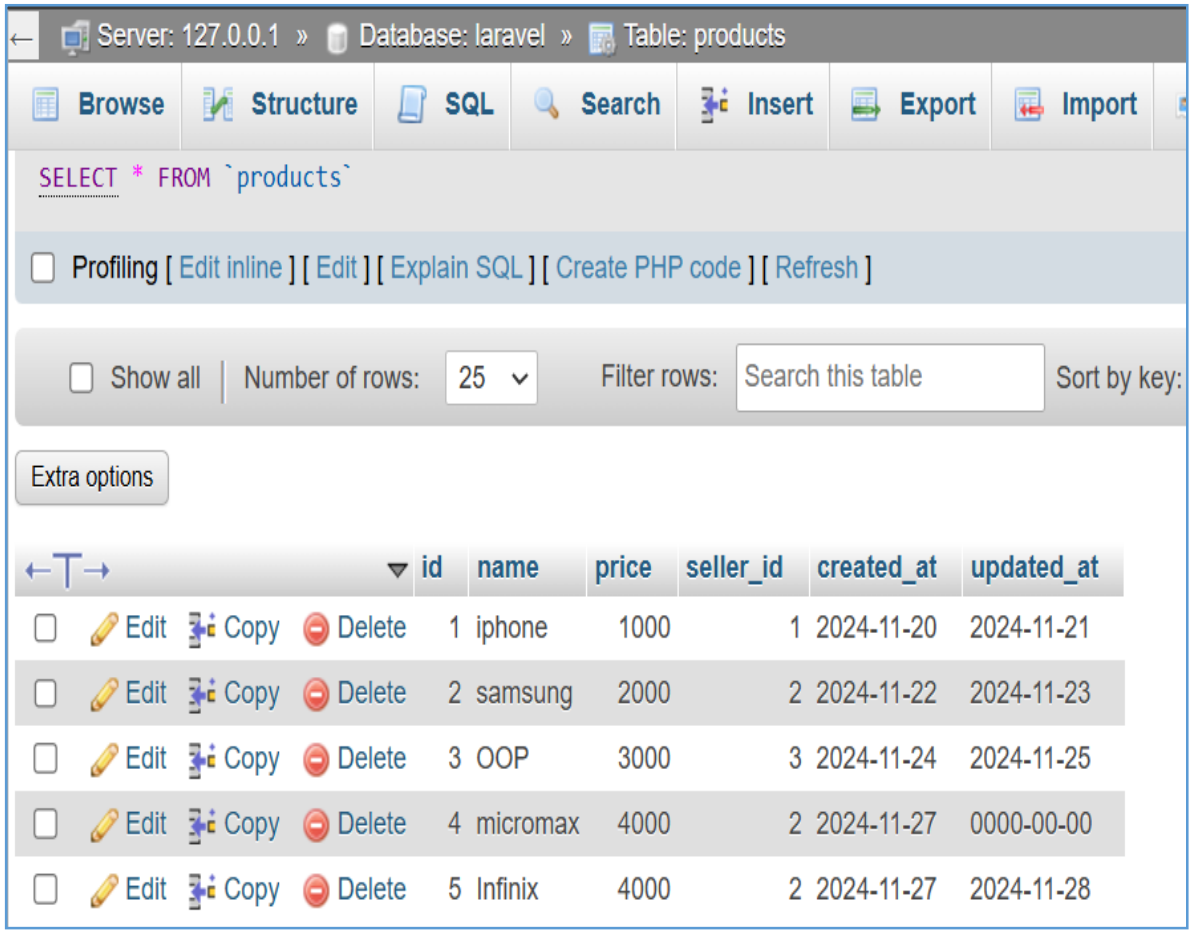
- **Description**: One record in a table can have multiple related records in another table.
- **Example**: A post has many comments.

Table Structure:

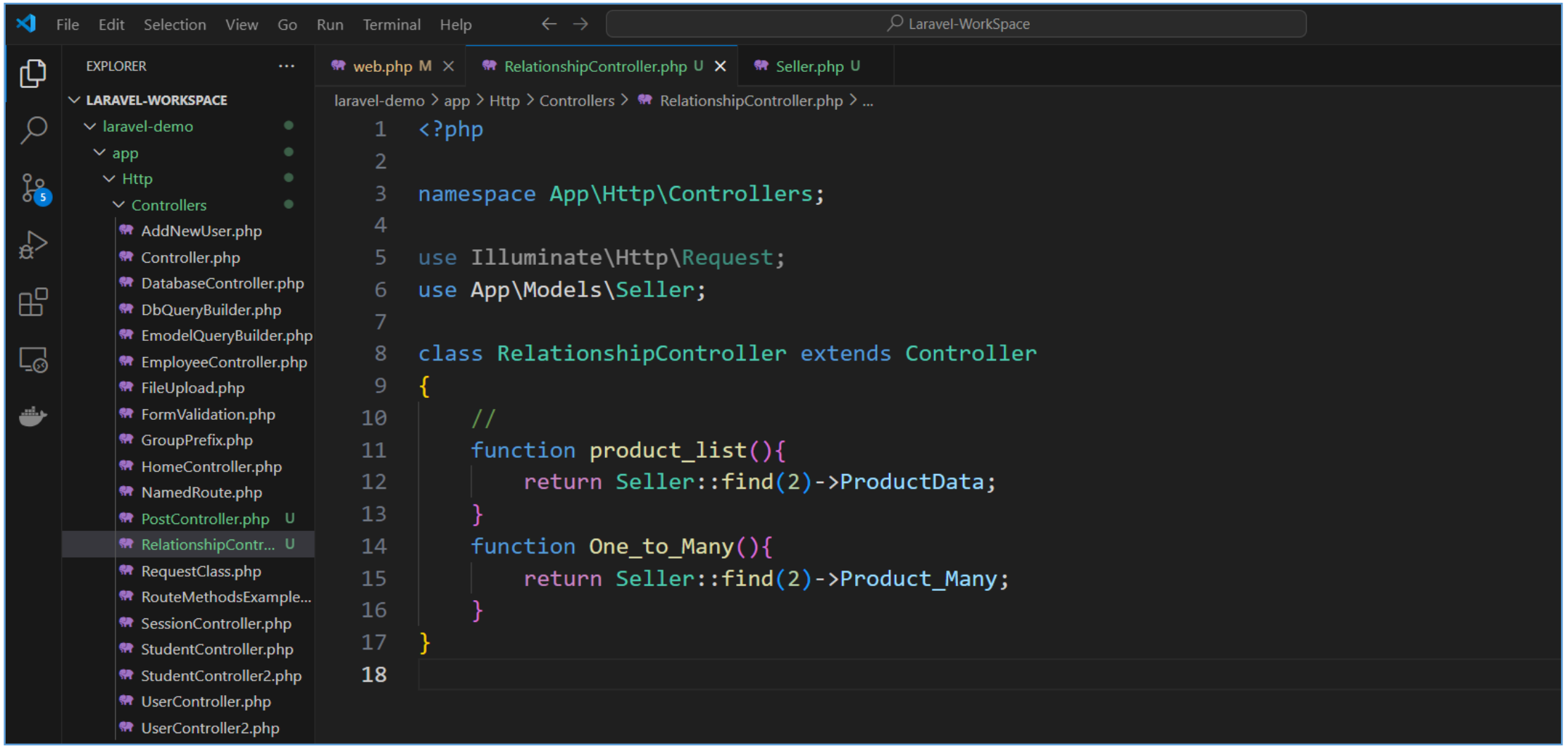
Posts Table	Comments Table
`id` (PK)	`id` (PK)
`title`	`post_id` (FK)
	`comment`

Example (Databases)

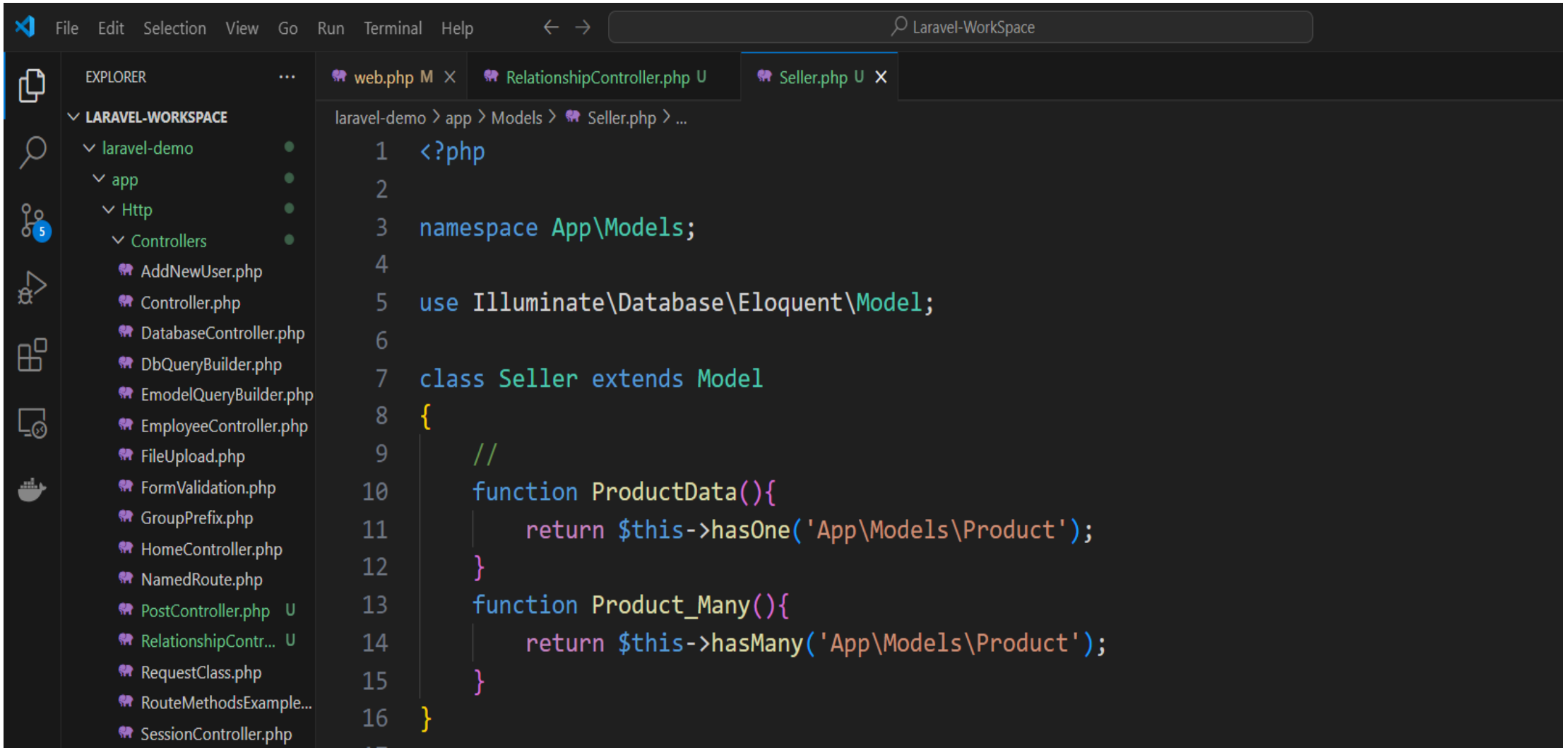




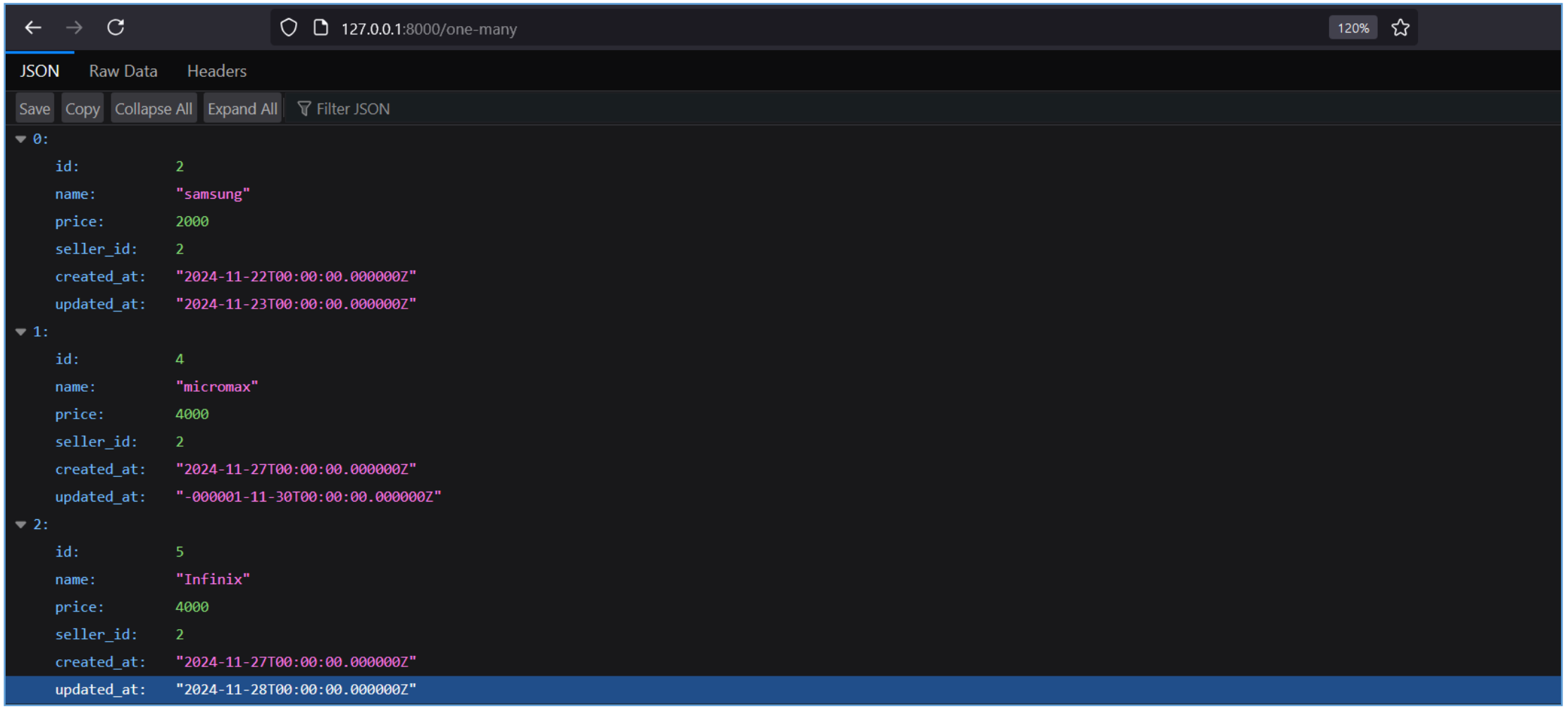
Controller (Relationship Controller)



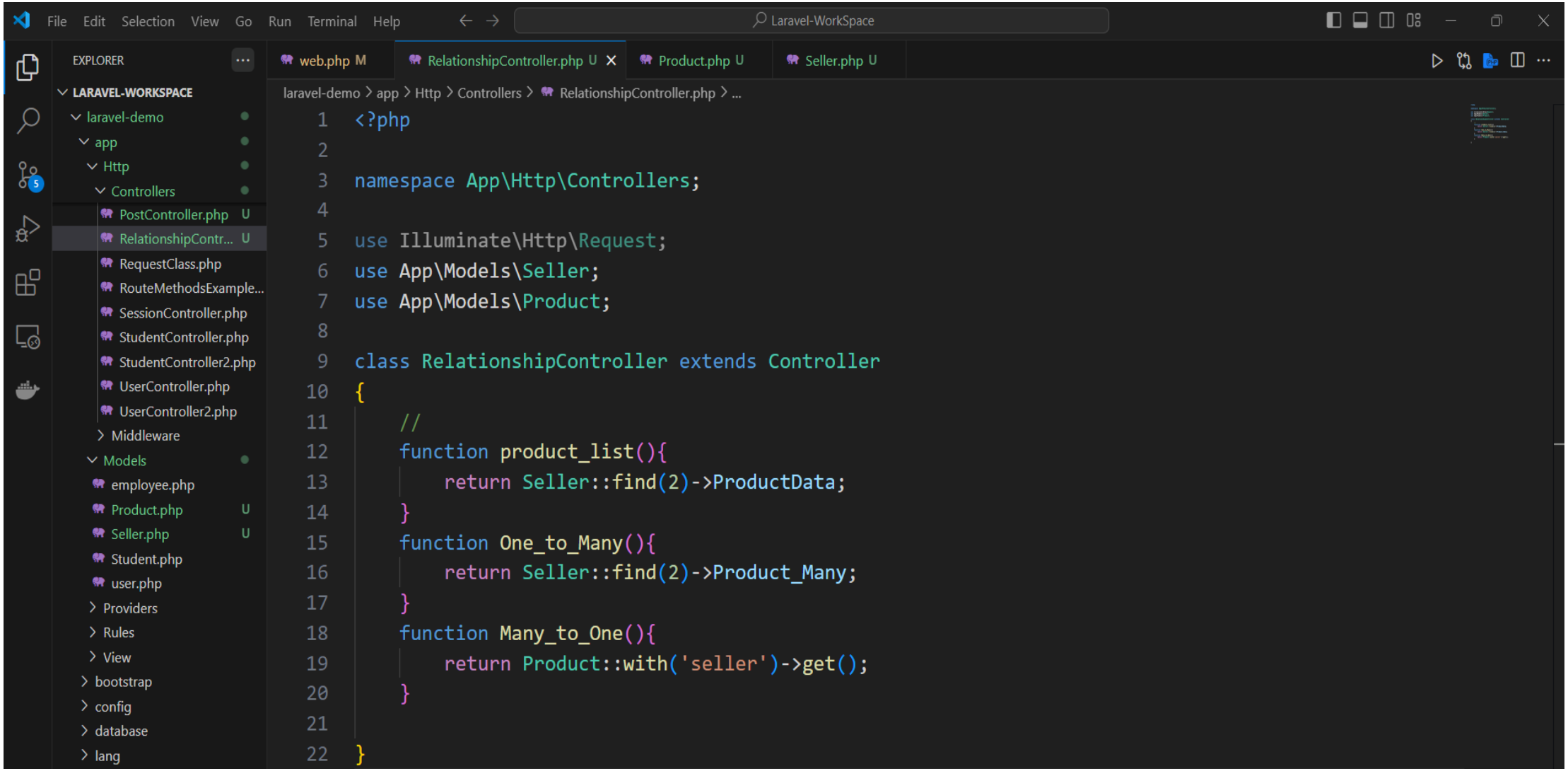
Model (Seller)



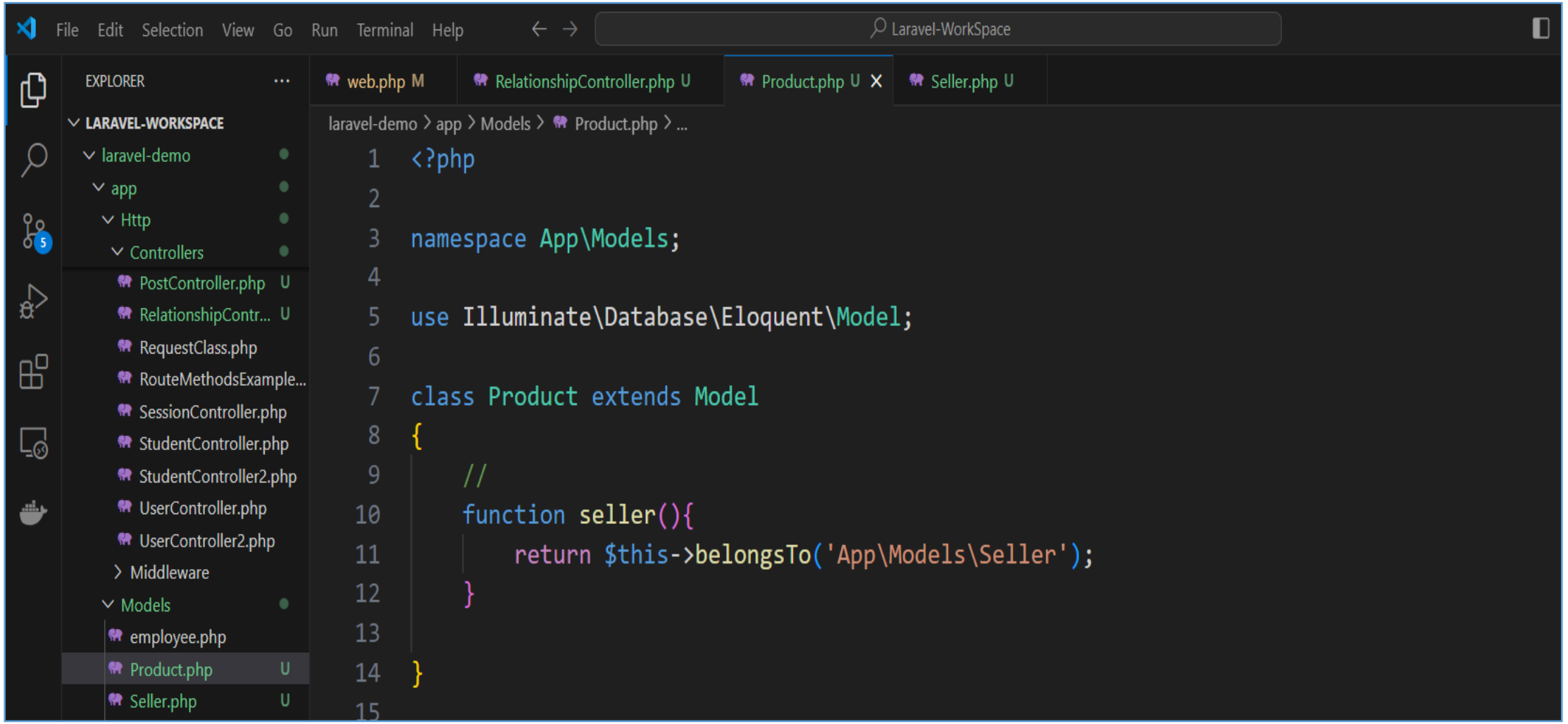
Output


127.0.0.1:8000/one-many
120%
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON
0:
id: 2
name: "samsung"
price: 2000
seller_id: 2
created_at: "2024-11-22T00:00:00.000000Z"
updated_at: "2024-11-23T00:00:00.000000Z"
1:
id: 4
name: "micromax"
price: 4000
seller_id: 2
created_at: "2024-11-27T00:00:00.000000Z"
updated_at: "-000001-11-30T00:00:00.000000Z"
2:
id: 5
name: "Infinix"
price: 4000
seller_id: 2
created_at: "2024-11-27T00:00:00.000000Z"
updated_at: "2024-11-28T00:00:00.000000Z"

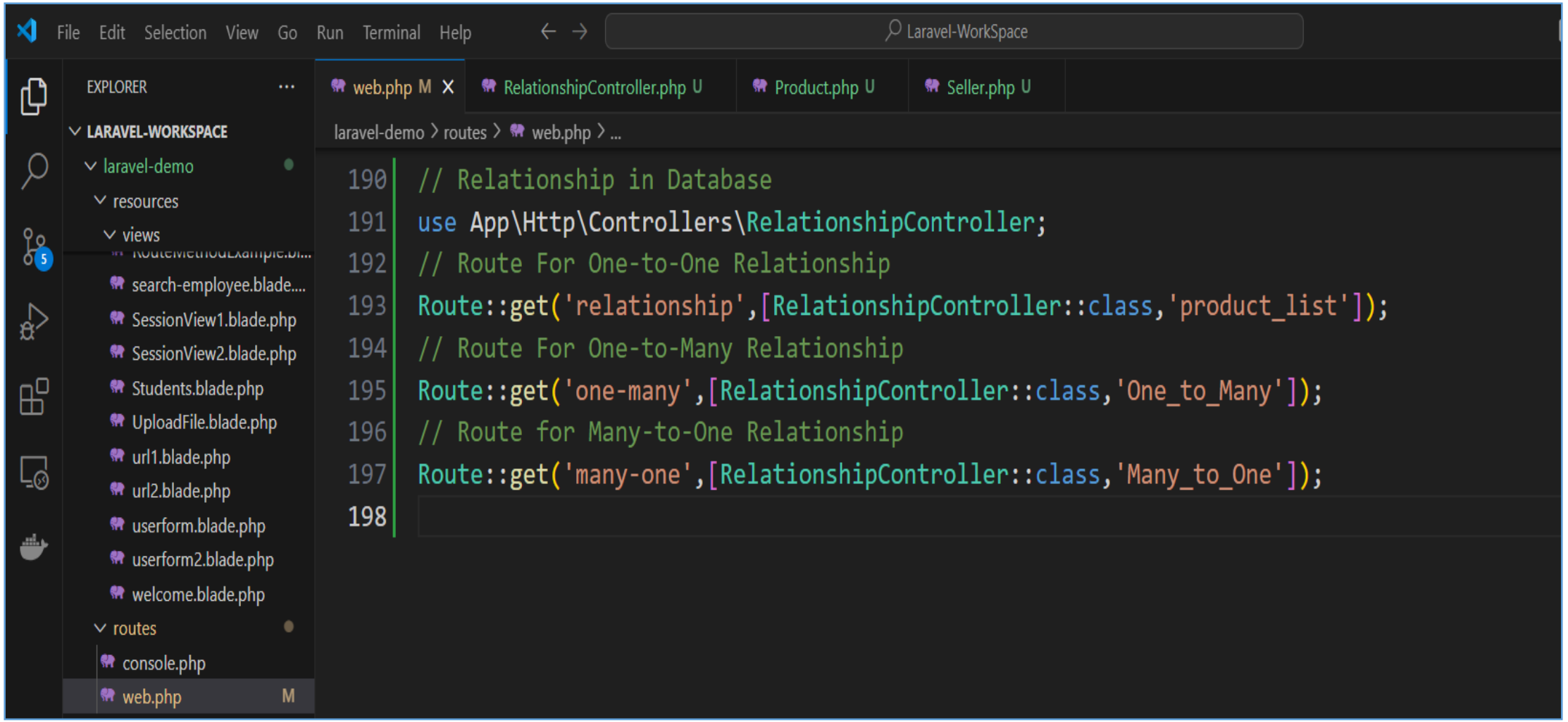
Many to One


File Edit Selection View Go Run Terminal Help
Laravel-WorkSpace
EXPLORER
LARAVEL-WORKSPACE
laravel-demo
app
Http
Controllers
PostController.php U
RelationshipContr... U
RequestClass.php
RouteMethodsExample...
SessionController.php
StudentController.php
StudentController2.php
UserController.php
UserController2.php
Middleware
Models
employee.php
Product.php U
Seller.php U
Student.php
user.php
Providers
Rules
View
bootstrap
config
database
lang
web.php M
RelationshipController.php U
Product.php U
Seller.php U
laravel-demo > app > Http > Controllers > RelationshipController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\Seller;
7 use App\Models\Product;
8
9 class RelationshipController extends Controller
10 {
11 //
12 function product_list(){
13 return Seller::find(2)->ProductData;
14 }
15 function One_to_Many(){
16 return Seller::find(2)->Product_Many;
17 }
18 function Many_to_One(){
19 return Product::with('seller')->get();
20 }
21
22 }

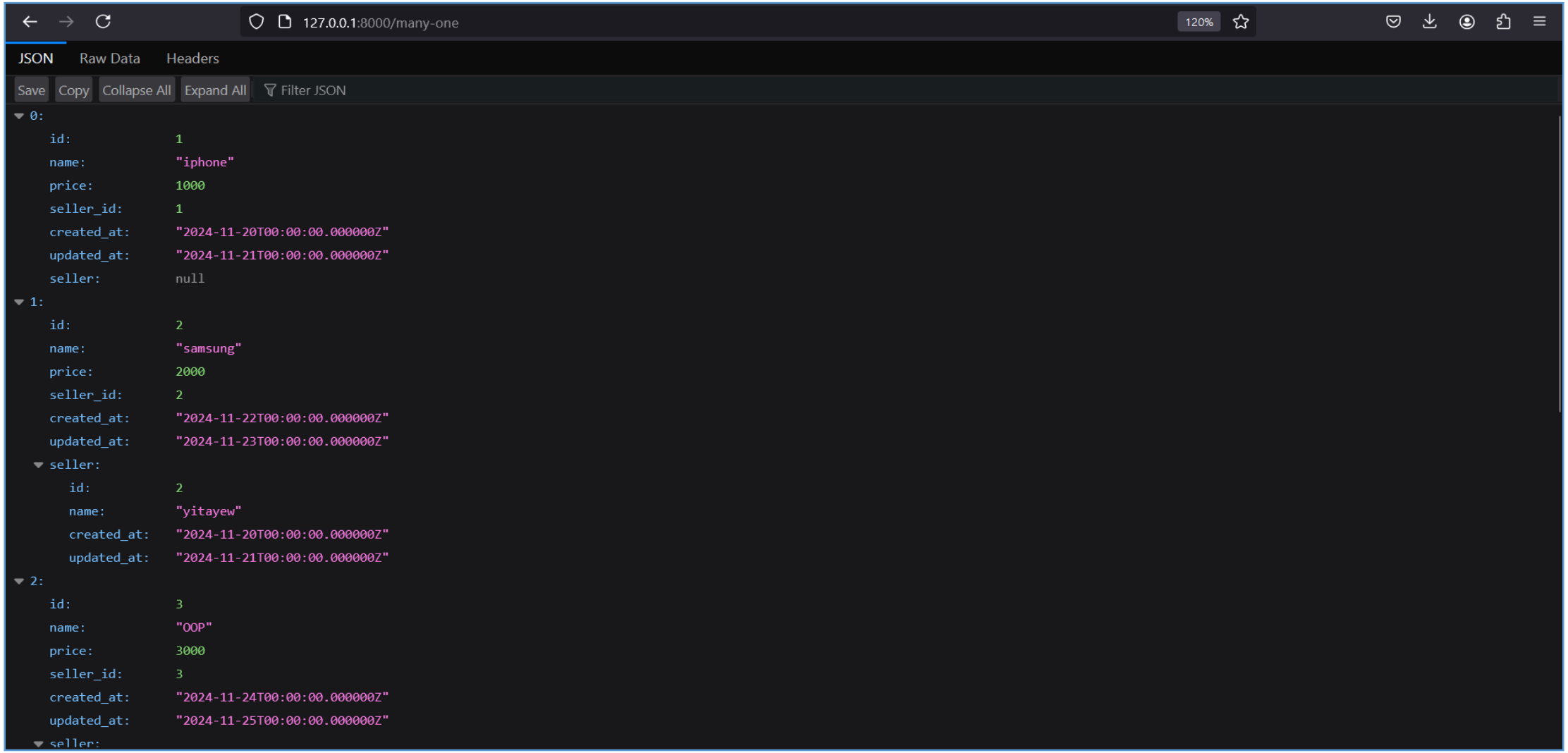
Model Definition (Product)



Route


File Edit Selection View Go Run Terminal Help
Laravel-WorkSpace
EXPLORER
LARAVEL-WORKSPACE
laravel-demo
resources
views
search-employee.blade....
SessionView1.blade.php
SessionView2.blade.php
Students.blade.php
UploadFile.blade.php
url1.blade.php
url2.blade.php
userform.blade.php
userform2.blade.php
welcome.blade.php
routes
console.php
web.php M
web.php M
RelationshipController.php U
Product.php U
Seller.php U
laravel-demo > routes > web.php > ...
190 // Relationship in Database
191 use App\Http\Controllers\RelationshipController;
192 // Route For One-to-One Relationship
193 Route::get('relationship',[RelationshipController::class,'product_list']);
194 // Route For One-to-Many Relationship
195 Route::get('one-many',[RelationshipController::class,'One_to_Many']);
196 // Route for Many-to-One Relationship
197 Route::get('many-one',[RelationshipController::class,'Many_to_One']);
198

Output


127.0.0.1:8000/many-one
120%
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON
0:
id: 1
name: "iphone"
price: 1000
seller_id: 1
created_at: "2024-11-20T00:00:00.000000Z"
updated_at: "2024-11-21T00:00:00.000000Z"
seller: null
1:
id: 2
name: "samsung"
price: 2000
seller_id: 2
created_at: "2024-11-22T00:00:00.000000Z"
updated_at: "2024-11-23T00:00:00.000000Z"
seller:
id: 2
name: "yitayew"
created_at: "2024-11-20T00:00:00.000000Z"
updated_at: "2024-11-21T00:00:00.000000Z"
2:
id: 3
name: "OOP"
price: 3000
seller_id: 3
created_at: "2024-11-24T00:00:00.000000Z"
updated_at: "2024-11-25T00:00:00.000000Z"
seller:

Many-to-Many Relationship

3. Many-to-Many Relationship

- **Description**: Multiple records in one table are associated with multiple records in another table.
- **Example**: Users can have many roles, and roles can belong to many users.

Table Structure:

Users Table	Roles Table	role_user Table (Pivot)
`id` (PK)	`id` (PK)	`user_id` (FK)
`name`	`role_name`	`role_id` (FK)

Example (Model Definition)

Model Definition:

php Copy code

```php
class User extends Model {
    public function roles() {
        return $this->belongsToMany(Role::class);
    }
}

class Role extends Model {
    public function users() {
        return $this->belongsToMany(User::class);
    }
}
```

API in laravel

- An API (Application Programming Interface) in Laravel is a set of rules and endpoints that allows different software systems or applications to communicate with each other.
- Laravel provides robust tools for building RESTful APIs, making it easier to create, read, update, and delete resources over HTTP.

Core Features of Laravel APIs

Core Features of Laravel APIs

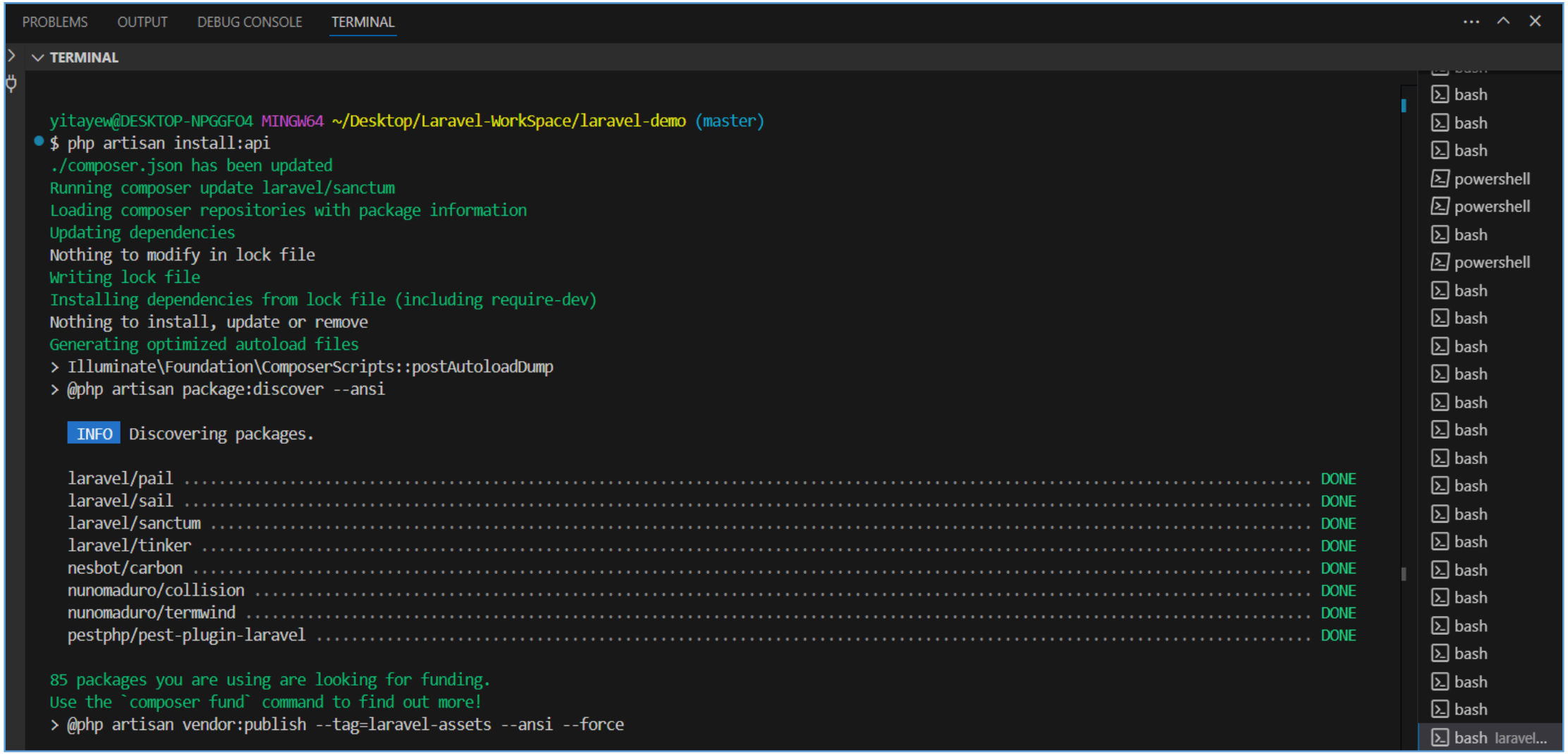
1. **Route Management**: Laravel allows defining API routes using `api.php`.
2. **Authentication**: Provides API token-based authentication through tools like Passport, Sanctum, or custom implementations.
3. **Validation**: Validates incoming requests using Laravel's built-in request validation.
4. **Eloquent Models**: Simplifies database interactions for API resources.
5. **Middleware**: Adds layers of security and processing to API routes.
6. **Resource Responses**: Formats API responses consistently using `Resources`.

Types of APIs in Laravel

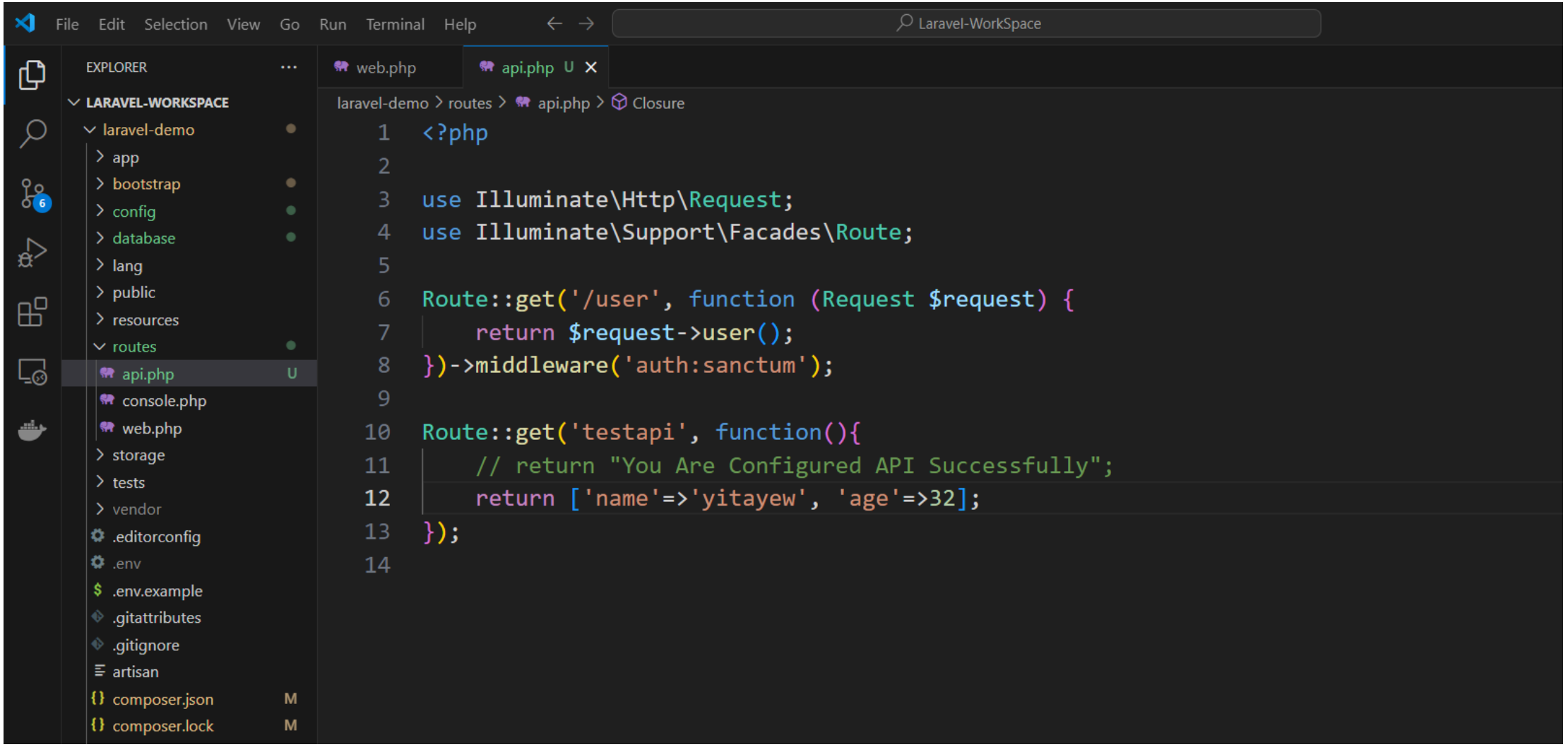
Types of APIs in Laravel

1. **RESTful APIs:**
    - Used for resource-based CRUD operations (Create, Read, Update, Delete).
    - Example: Managing user data, products, etc.
2. **Authentication APIs:**
    - Used for user login, logout, and token management.
    - Example: Login APIs for mobile apps.
3. **Third-Party APIs:**
    - Laravel applications can consume external APIs using HTTP clients like Guzzle or Laravel's HTTP client.

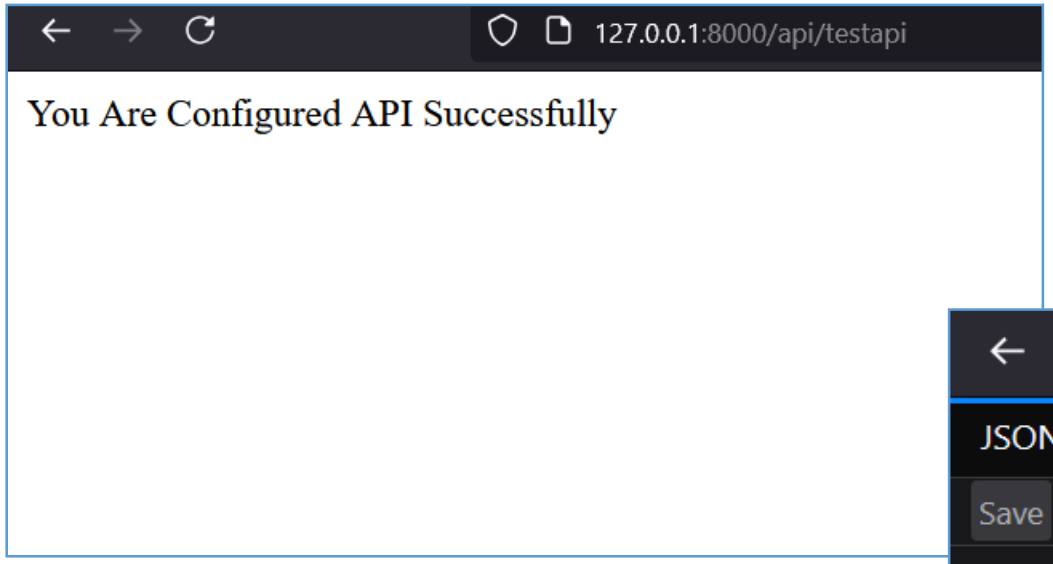
API Installation

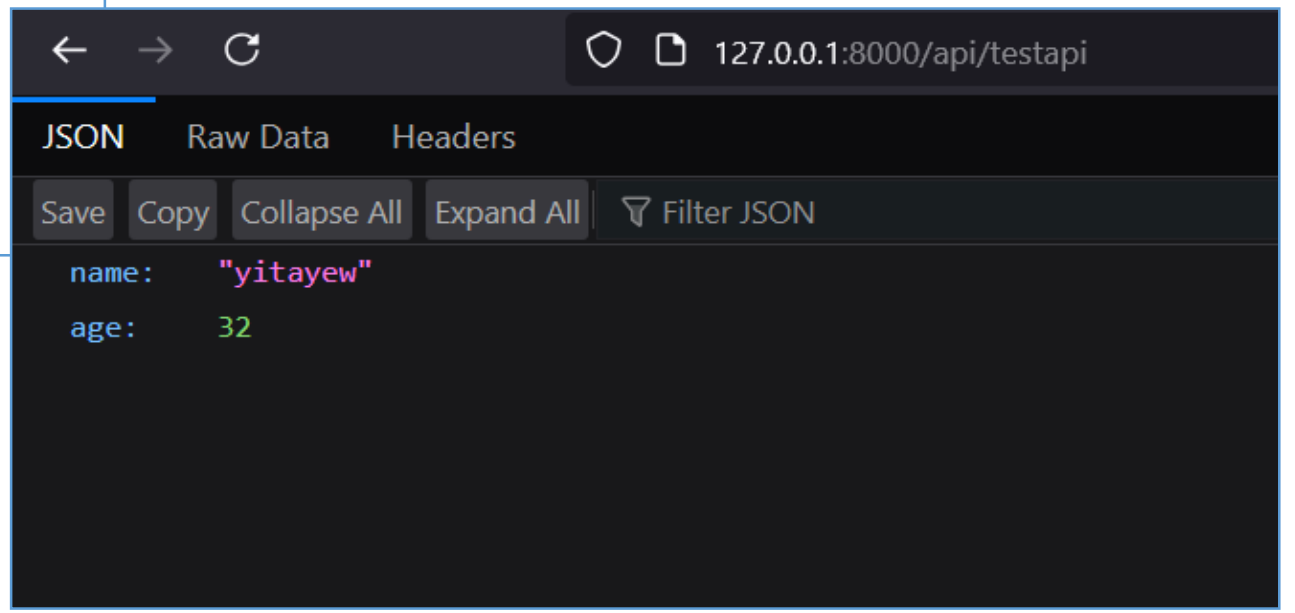


API (route/api.php)

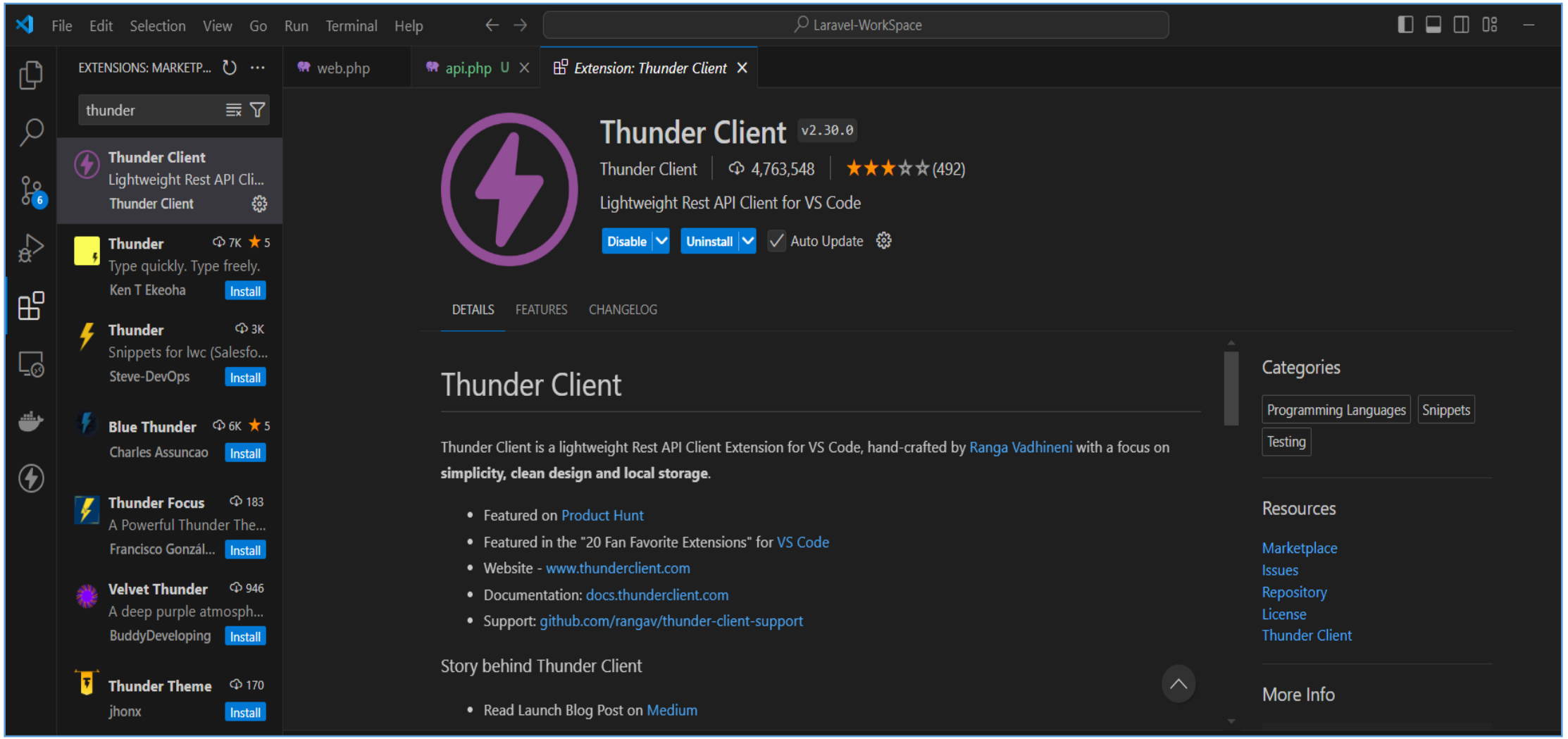


Output (http://127.0.0.1:8000/api/testapi)

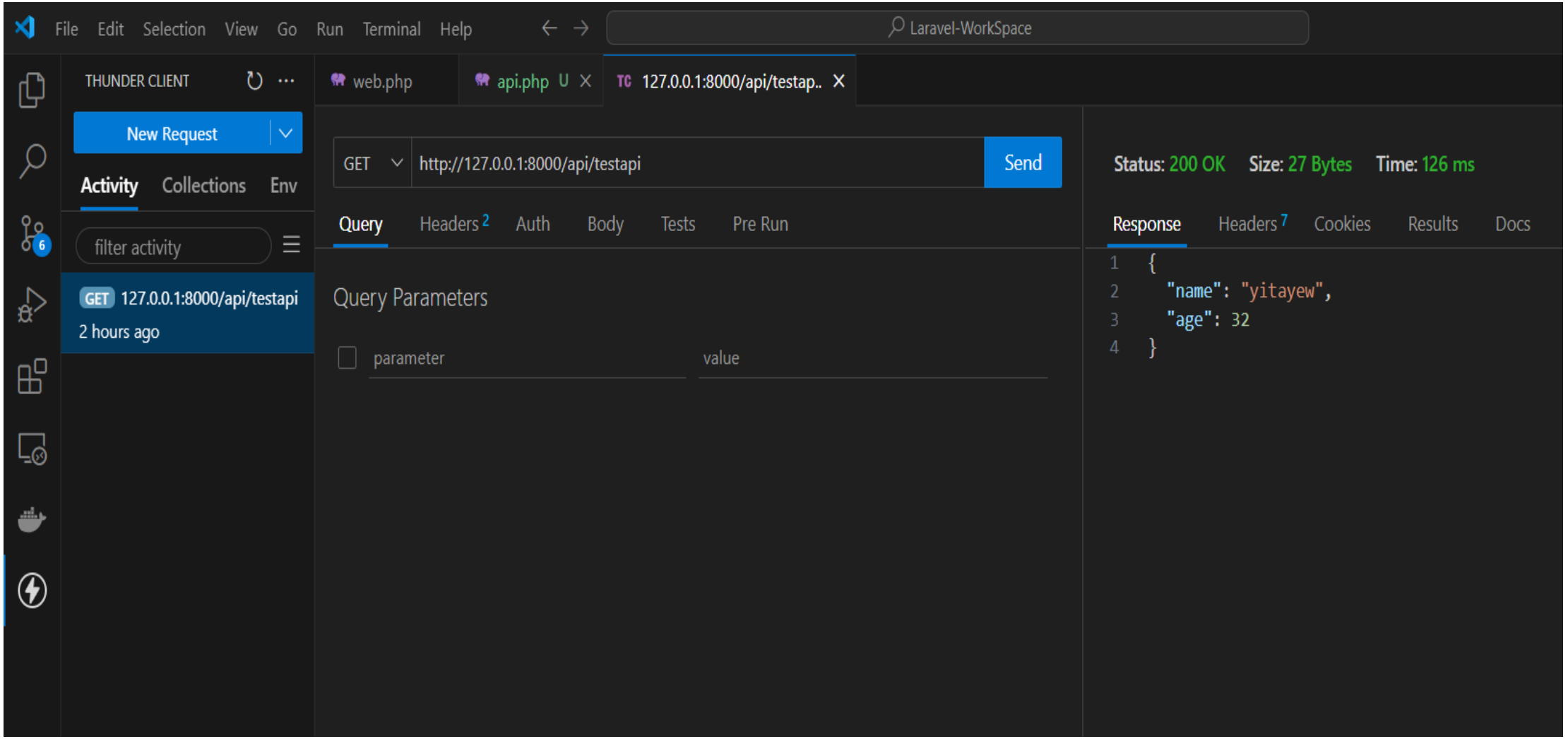




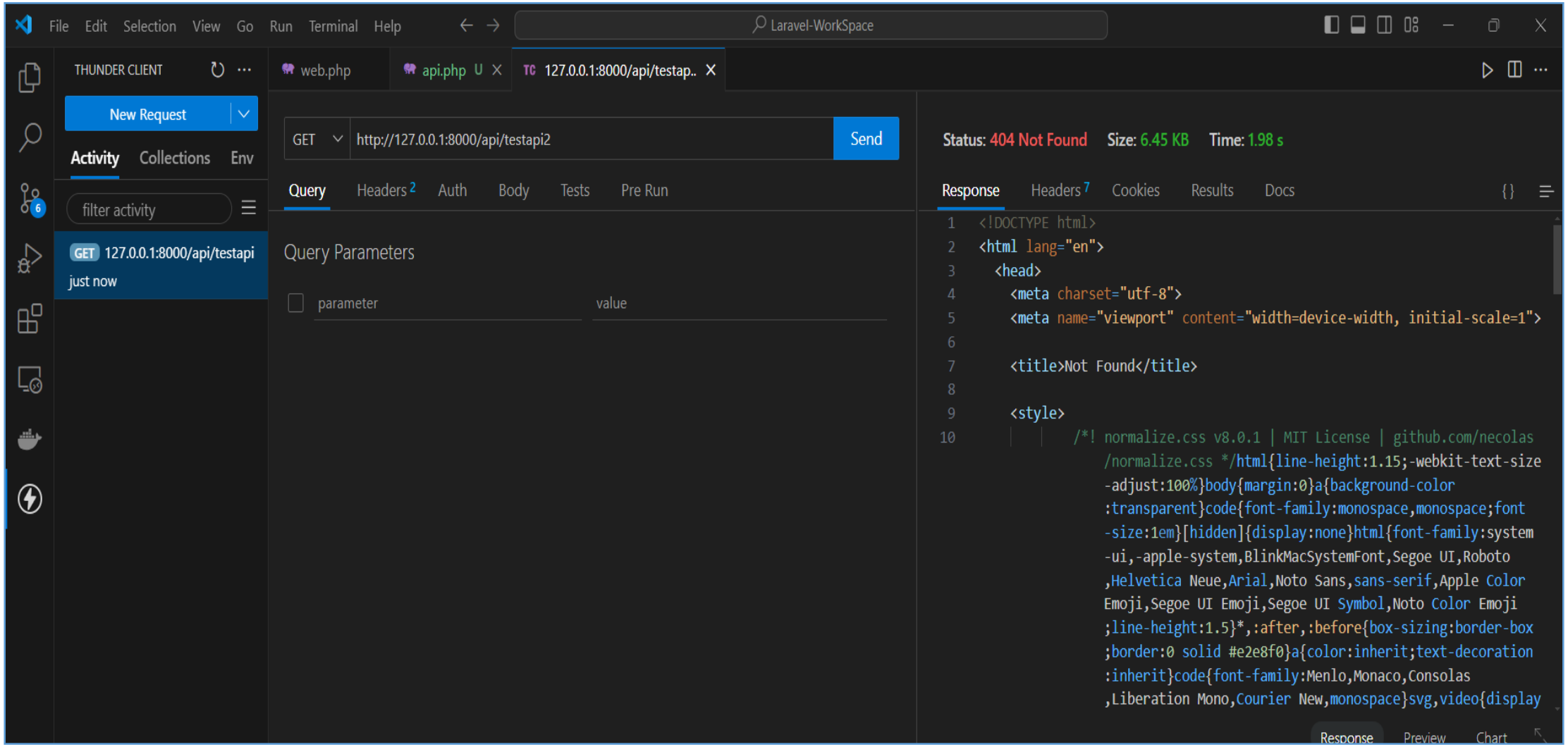
Test API with VS Code Thunder Client



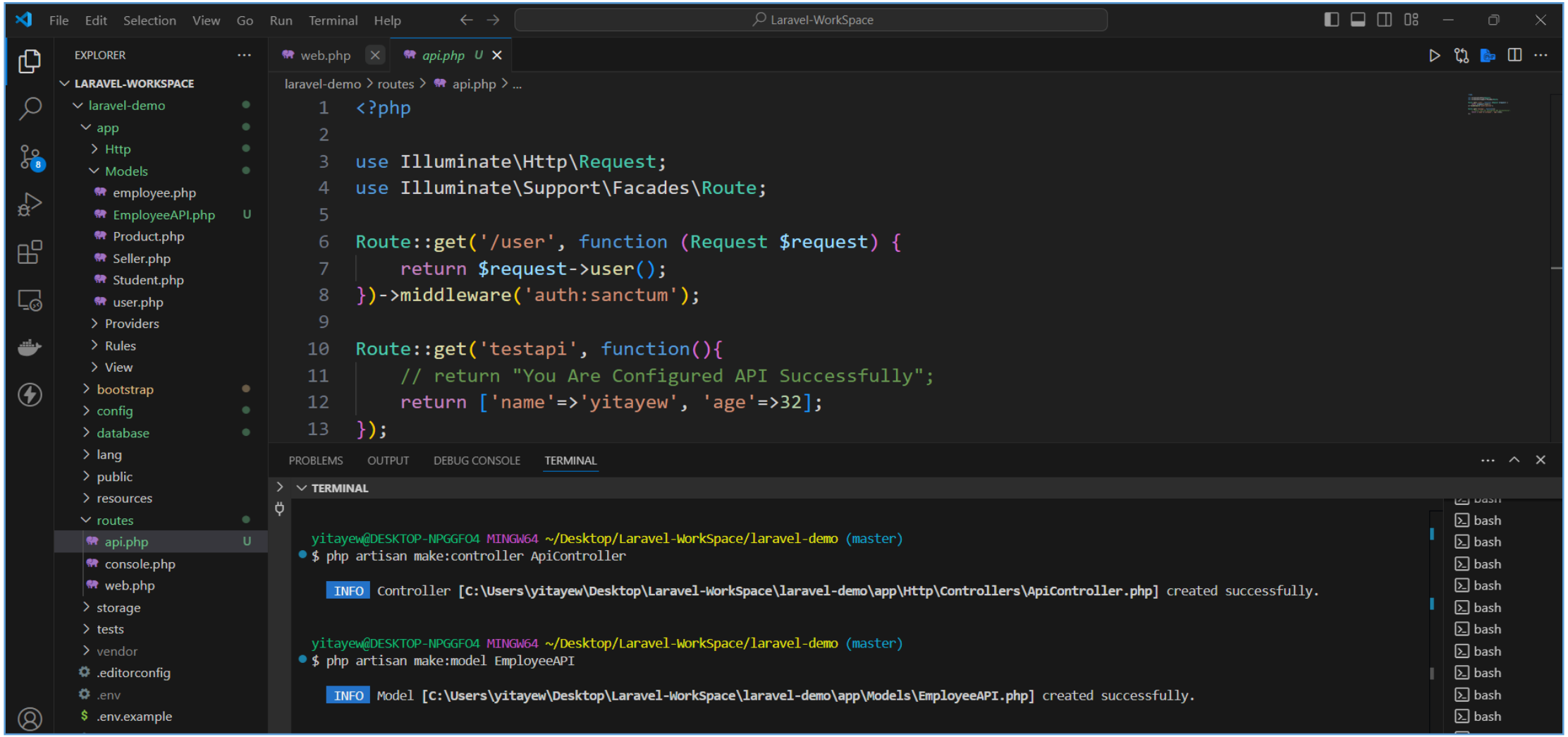
Output(http://127.0.0.1:8000/api/testapi)



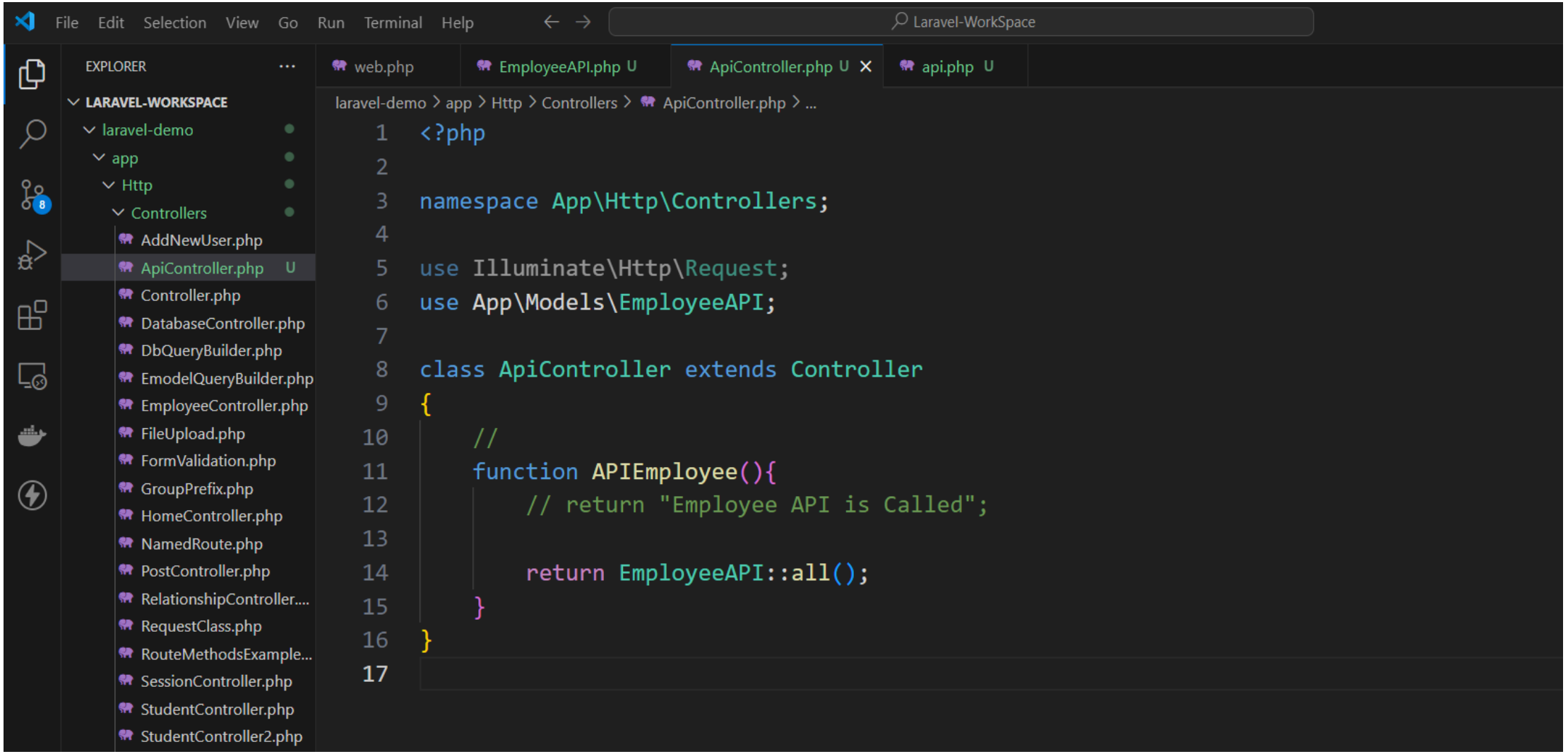
Output (http://127.0.0.1:8000/api/testapi2)



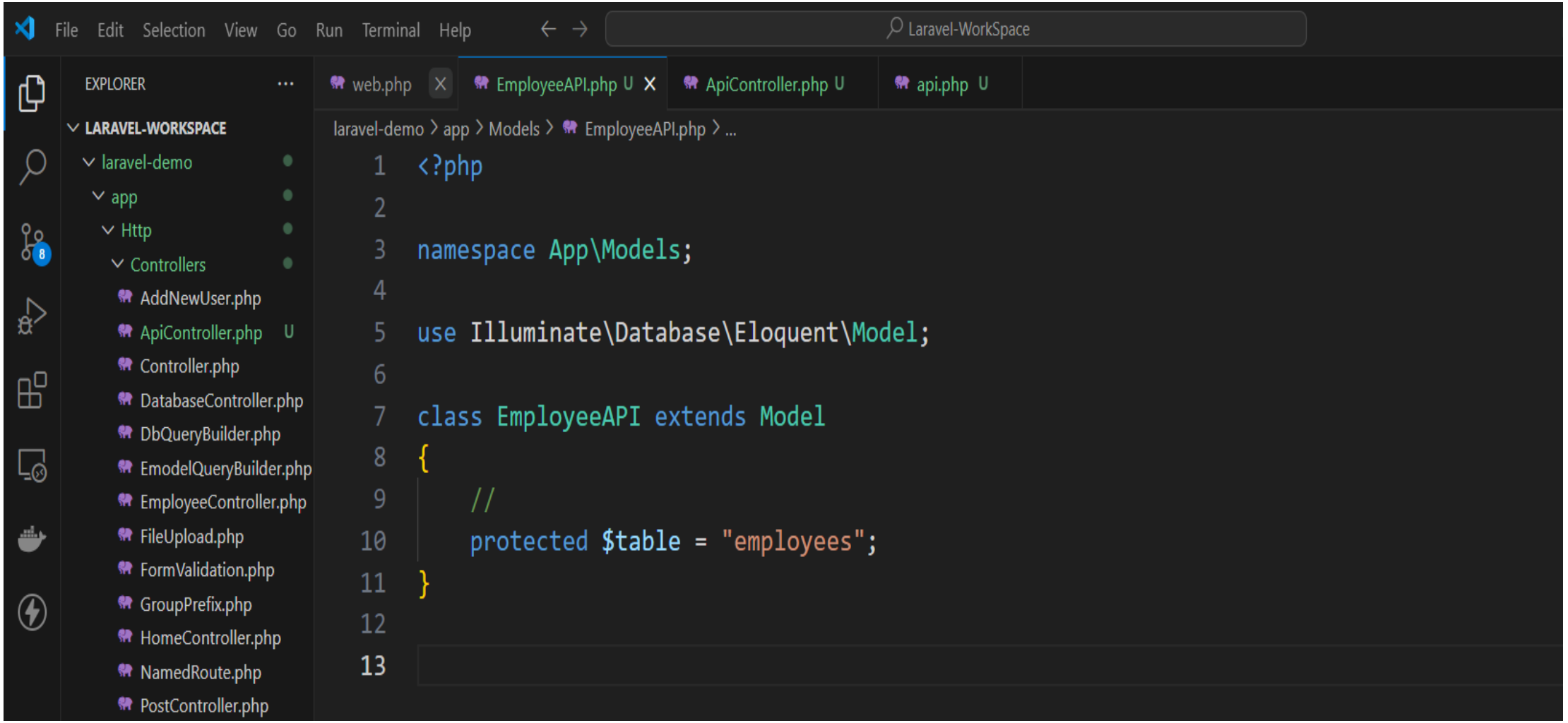
Make First GET API with Database



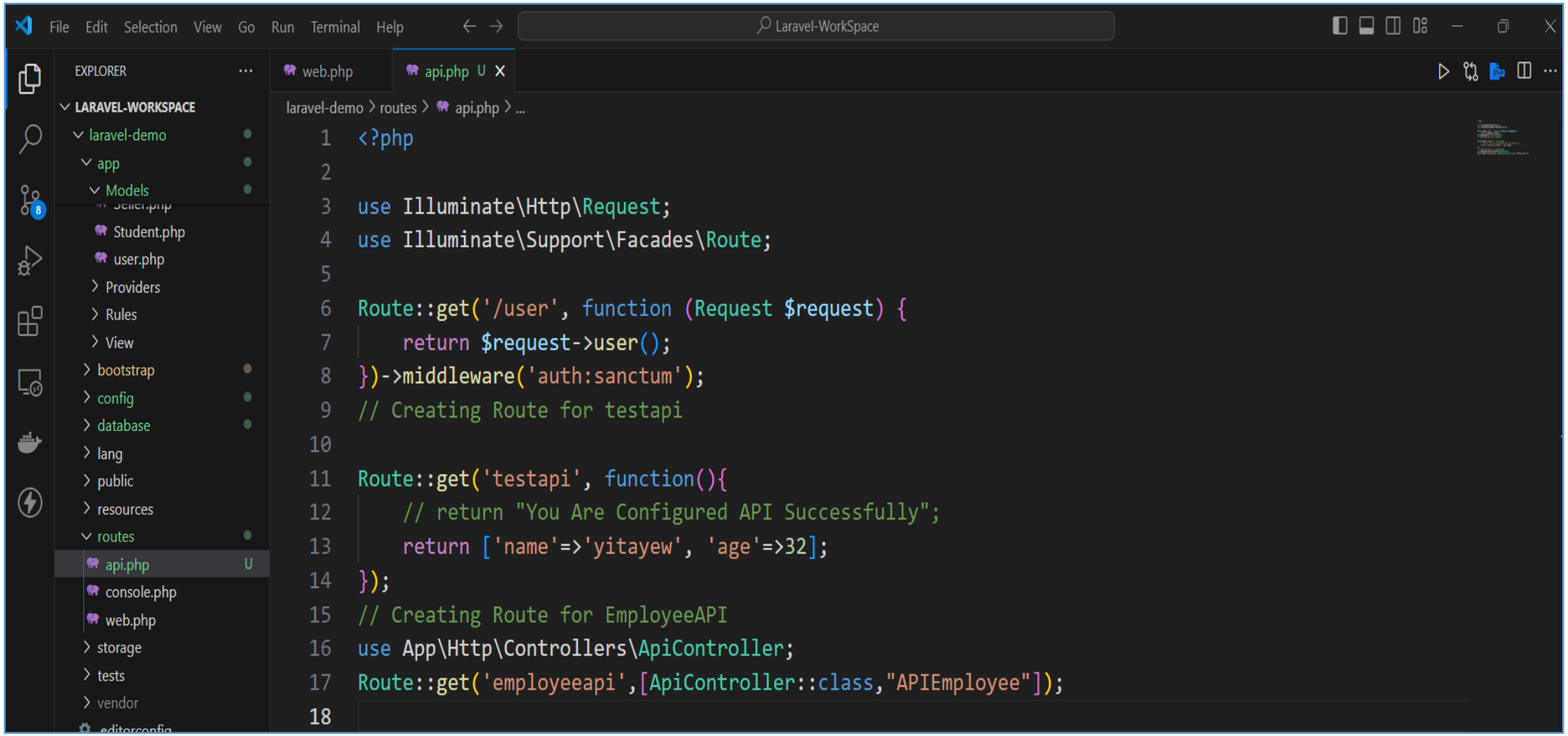
Controller (ApiController)



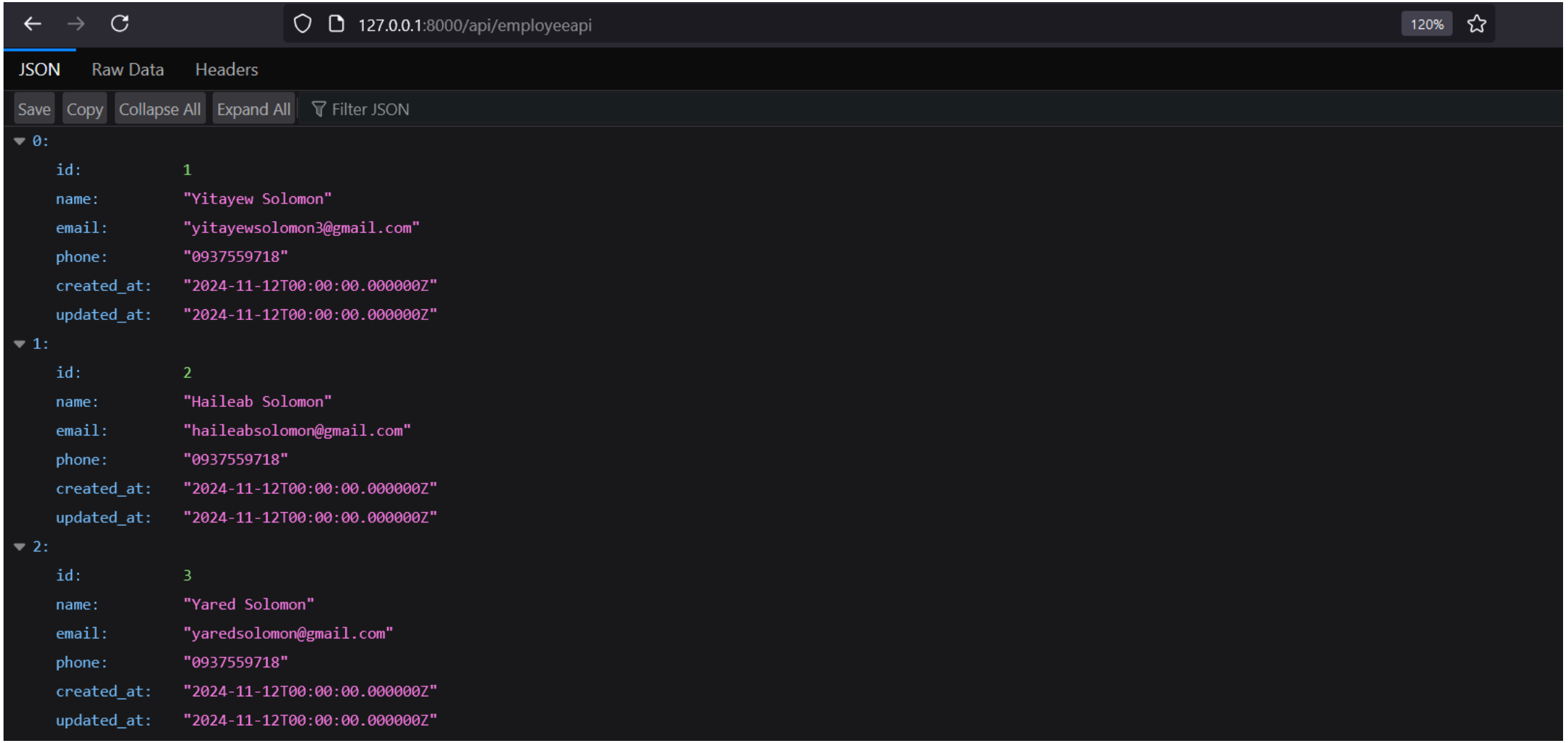
Model (EmployeeAPI)



API Route


File Edit Selection View Go Run Terminal Help
Laravel-WorkSpace
EXPLORER
LARAVEL-WORKSPACE
laravel-demo
app
Models
Student.php
user.php
Providers
Rules
View
bootstrap
config
database
lang
public
resources
routes
api.php U
console.php
web.php
storage
tests
vendor
web.php
api.php U
laravel-demo > routes > api.php > ...
1 <?php
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Route;
5
6 Route::get('/user', function (Request $request) {
7 return $request->user();
8 })->middleware('auth:sanctum');
9 // Creating Route for testapi
10
11 Route::get('testapi', function(){
12 // return "You Are Configured API Successfully";
13 return ['name'=>'yitayew', 'age'=>32];
14 });
15 // Creating Route for EmployeeAPI
16 use App\Http\Controllers\ApiController;
17 Route::get('employeeapi',[ApiController::class,"APIEmployee"]);
18

Output (http://127.0.0.1:8000/api/employeeapi)



API Methods is Laravel

- Laravel supports all major HTTP methods for building APIs. These methods allow developers to perform CRUD (Create, Read, Update, Delete) operations and more. Here's a breakdown of the methods Laravel supports, their purposes, and Description:

GET Method

1. GET

- **Purpose:** Retrieve data from the server.
- **Example:** Fetching a list of users or a single user by ID.

php Copy code

```php
Route::get('/users', [UserController::class, 'index']); // Fetch all users
Route::get('/users/{id}', [UserController::class, 'show']); // Fetch a user by ID
```

Post Method

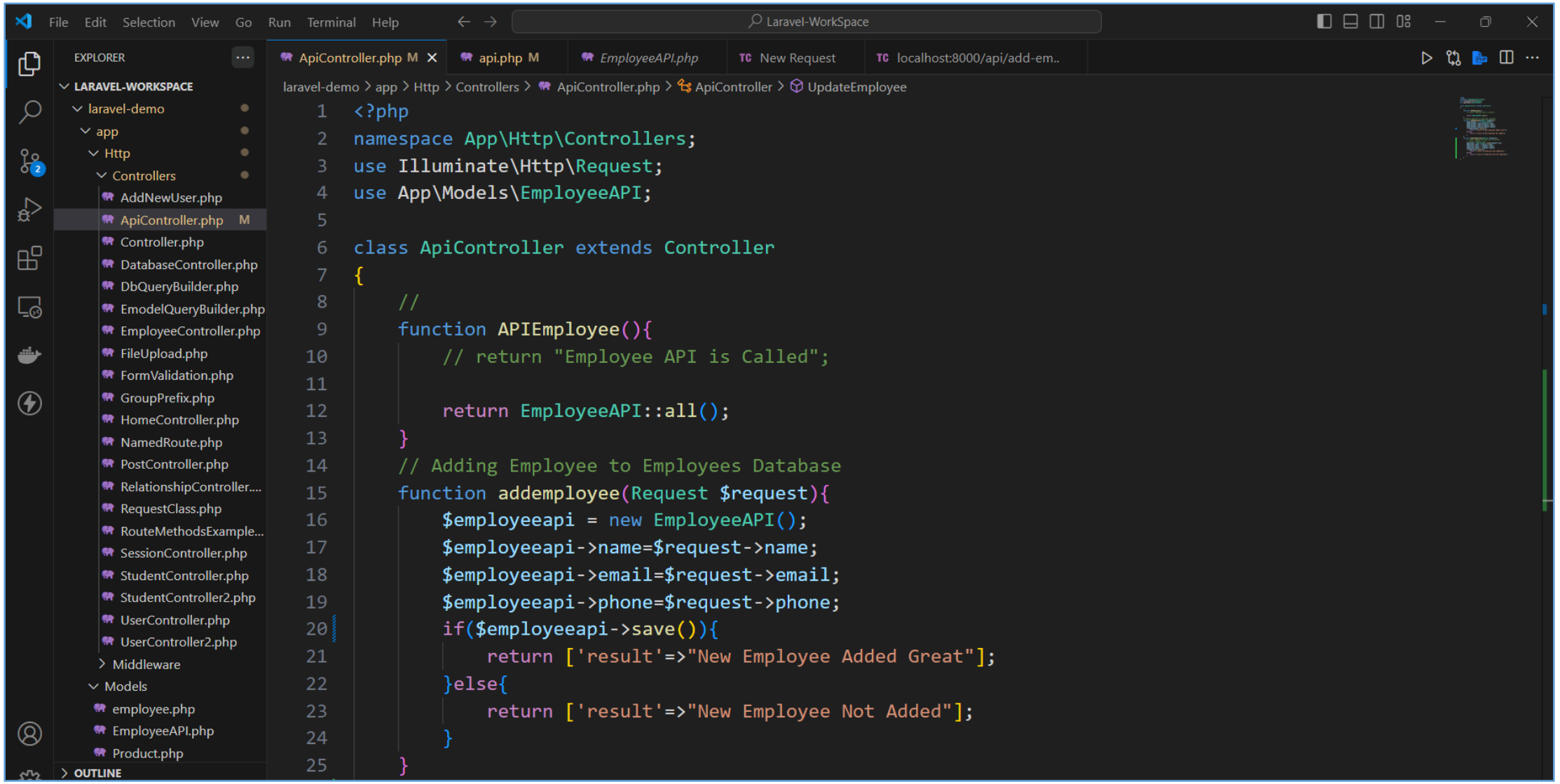
2. POST

- **Purpose**: Create new resources on the server.
- **Example**: Adding a new user to the database.

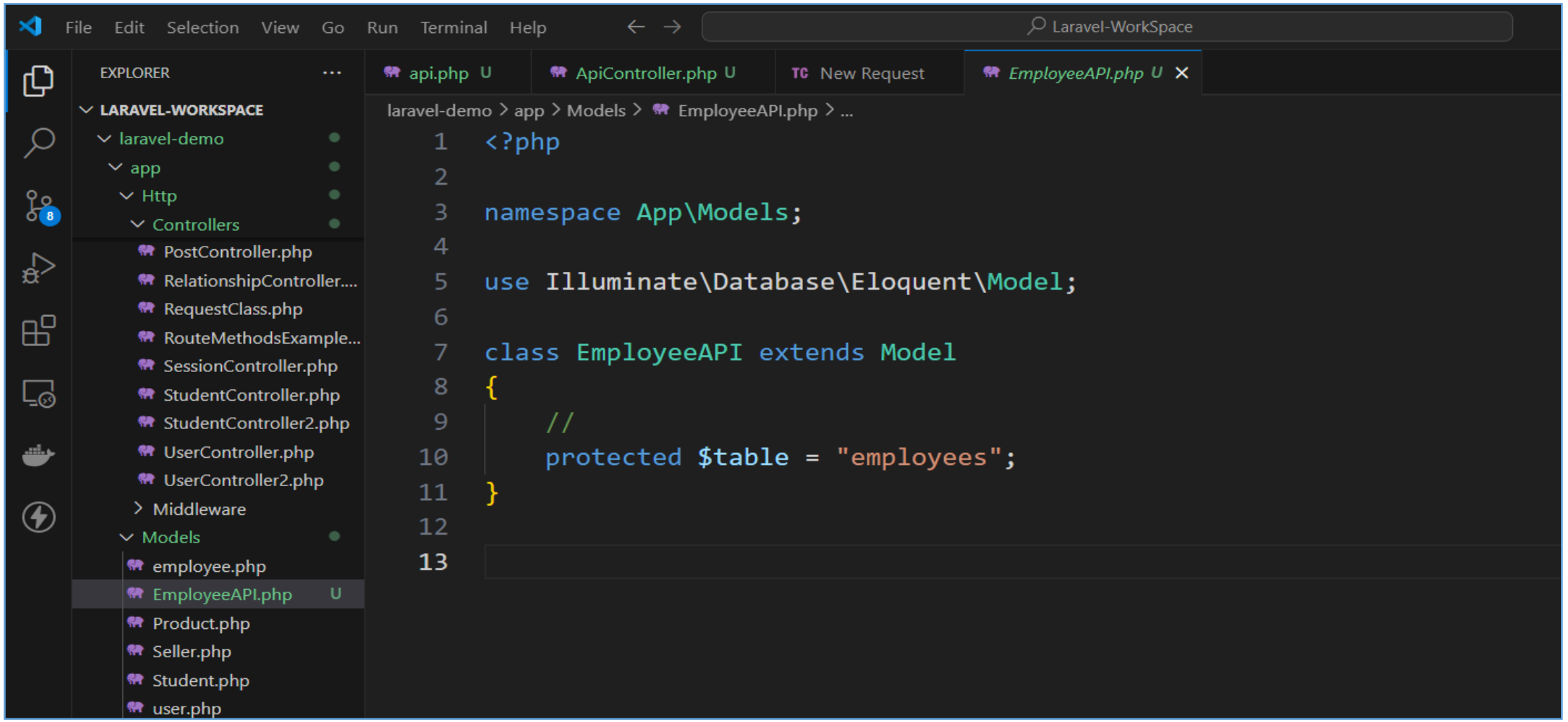
php Copy code

```php
Route::post('/users', [UserController::class, 'store']); // Create a new user
```

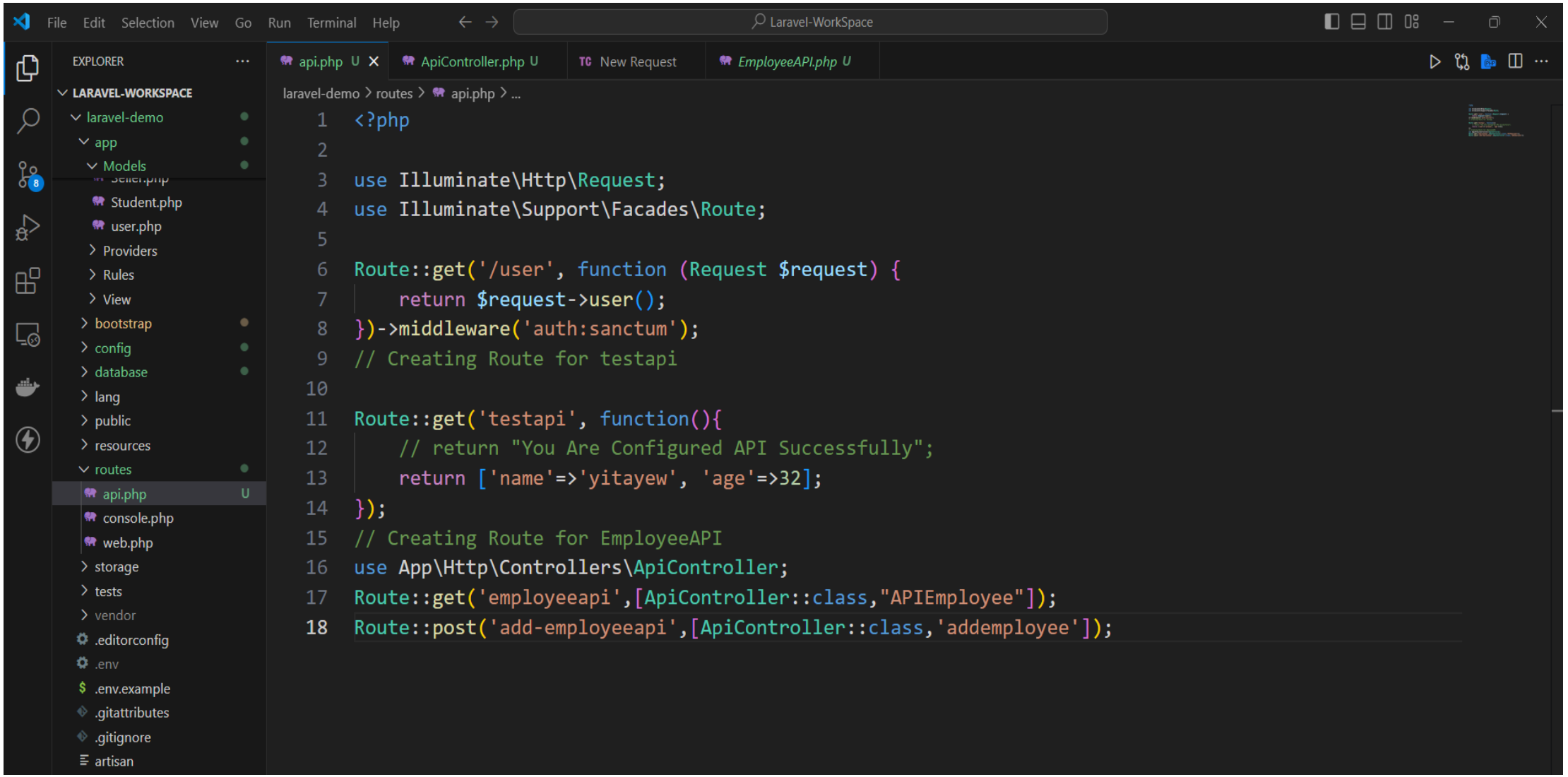
Example (APIController)



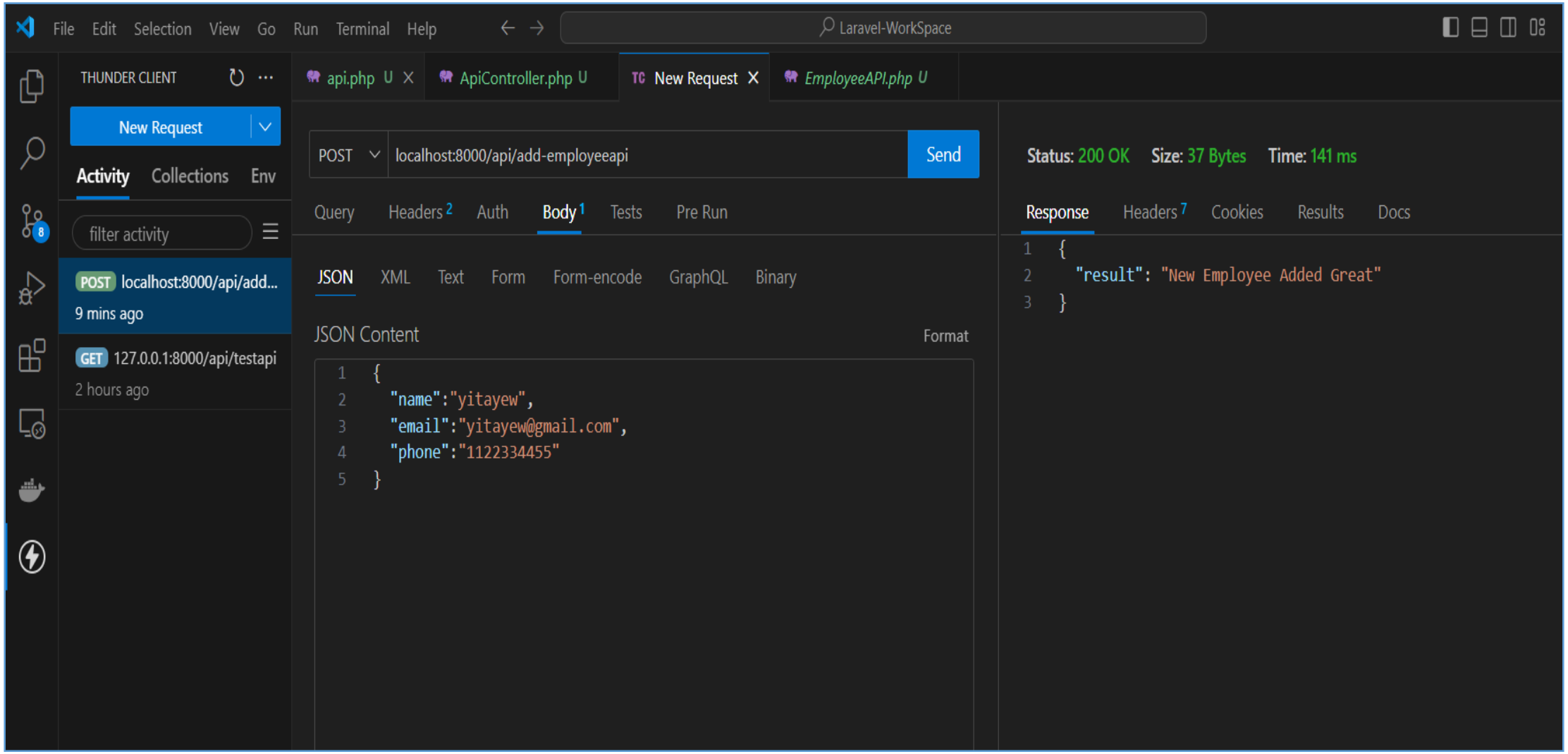
Model (EmployeeAPI)



Route (API)


File Edit Selection View Go Run Terminal Help
Laravel-WorkSpace
EXPLORER
LARAVEL-WORKSPACE
laravel-demo
app
Models
Student.php
user.php
Providers
Rules
View
bootstrap
config
database
lang
public
resources
routes
api.php U
console.php
web.php
storage
tests
vendor
.editorconfig
.env
.env.example
.gitattributes
.gitignore
artisan
api.php U
ApiController.php U
New Request
EmployeeAPI.php U
laravel-demo > routes > api.php > ...
1 <?php
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Route;
5
6 Route::get('/user', function (Request $request) {
7 return $request->user();
8 })->middleware('auth:sanctum');
9 // Creating Route for testapi
10
11 Route::get('testapi', function(){
12 // return "You Are Configured API Successfully";
13 return ['name'=>'yitayew', 'age'=>32];
14 });
15 // Creating Route for EmployeeAPI
16 use App\Http\Controllers\ApiController;
17 Route::get('employeeapi',[ApiController::class,"APIEmployee"]);
18 Route::post('add-employeeapi',[ApiController::class,'addemployee']);

Sending Add Request



Output (Database Table)

	id	name	email	phone	created_at	updated_at
Edit Copy Delete	1	Yitayew Solomon	yitayewsolomon3@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	2	Haileab Solomon	haileabsolomon@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	3	Yared Solomon	yaredsolomon@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	6	Natanim Yitayew	natan@gmail.com	0937559720	2024-11-12	2024-11-12
Edit Copy Delete	7	Dureti Guye	dure@gmail.com	0910010203	2024-11-13	2024-11-13
Edit Copy Delete	12	yitayew	yitayew@gmail.com	1122334455	2024-11-25	2024-11-25

PUT Method

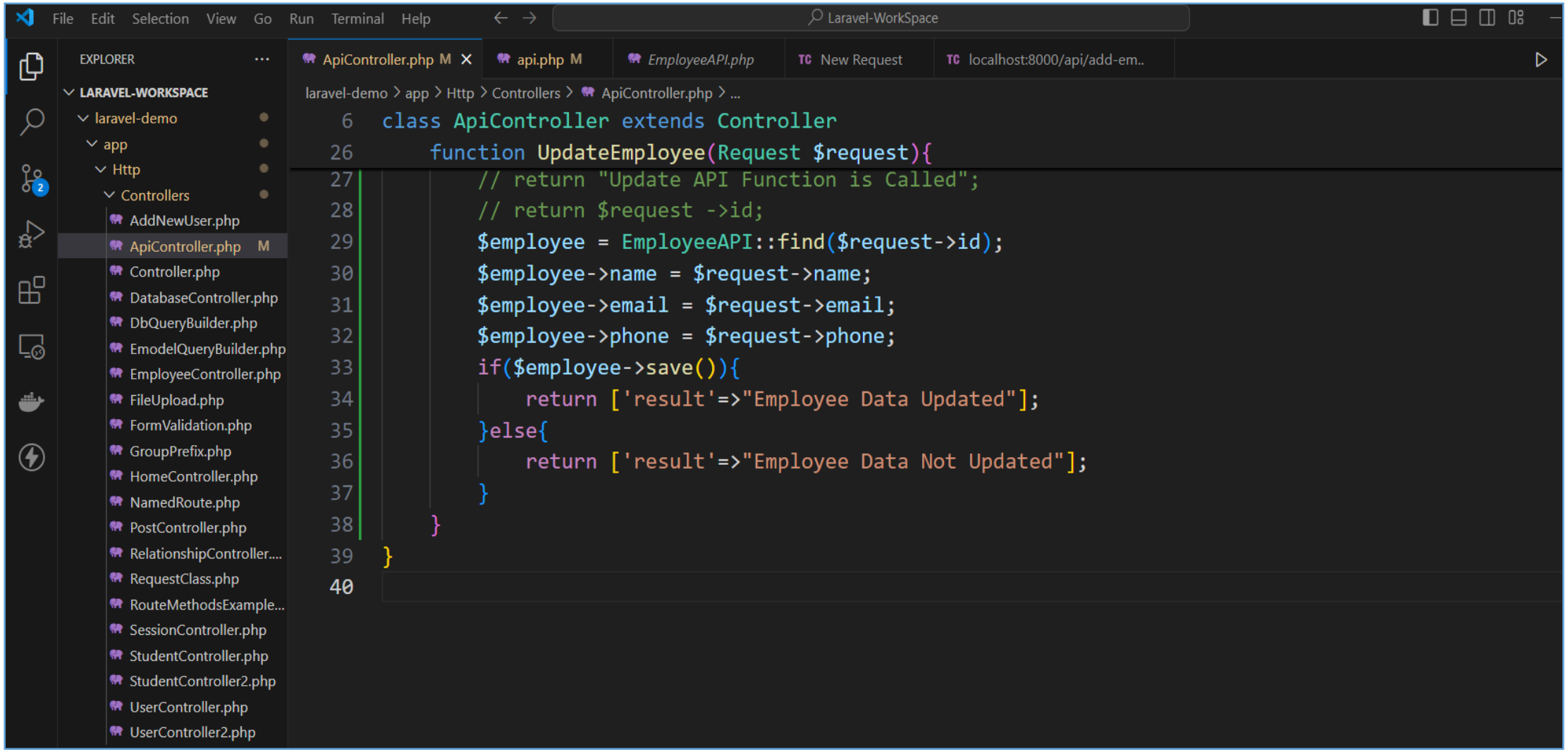
3. PUT

- **Purpose**: Update an existing resource completely.
- **Example**: Updating all fields of a user.

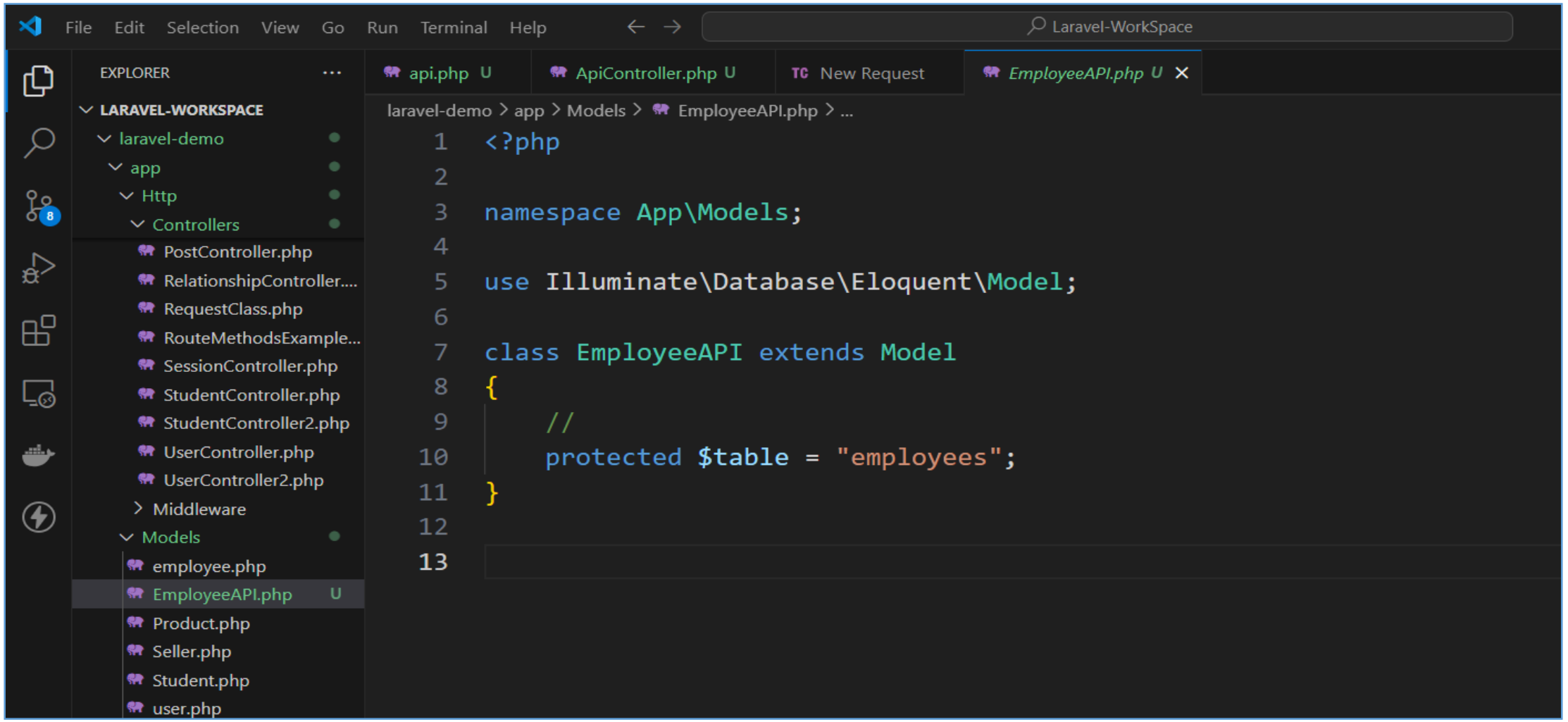
php Copy code

```php
Route::put('/users/{id}', [UserController::class, 'update']); // Update a user
```

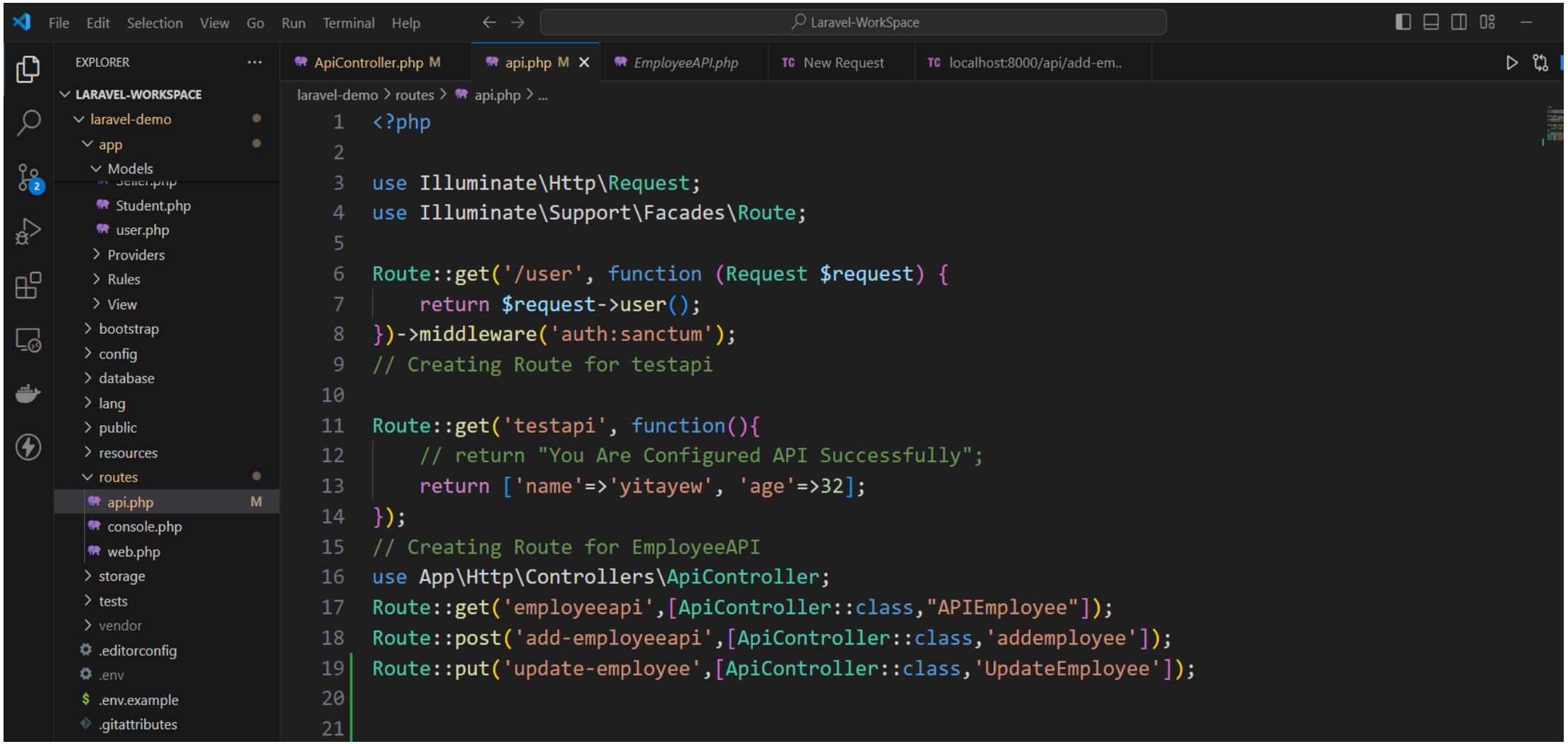
Example (APIController)



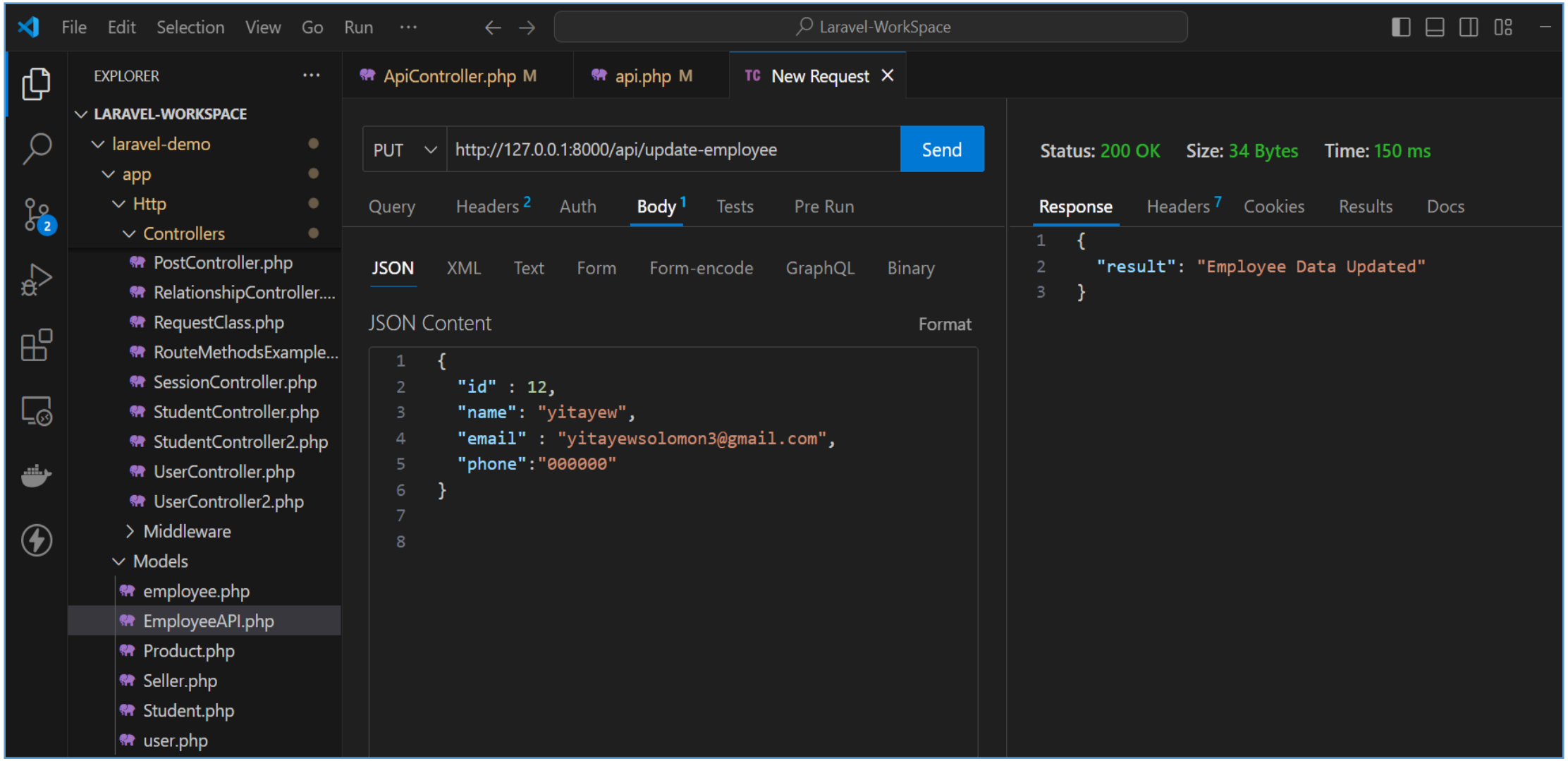
Model (EmployeeAPI)



Route


File Edit Selection View Go Run Terminal Help
Laravel-WorkSpace
EXPLORER
LARAVEL-WORKSPACE
laravel-demo
app
Models
Student.php
user.php
Providers
Rules
View
bootstrap
config
database
lang
public
resources
routes
api.php M
console.php
web.php
storage
tests
vendor
.editorconfig
.env
.env.example
.gitattributes
ApiController.php M
api.php M
EmployeeAPI.php
New Request
localhost:8000/api/add-em..
laravel-demo > routes > api.php > ...
1 <?php
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Route;
5
6 Route::get('/user', function (Request $request) {
7 return $request->user();
8 })->middleware('auth:sanctum');
9 // Creating Route for testapi
10
11 Route::get('testapi', function(){
12 // return "You Are Configured API Successfully";
13 return ['name'=>'yitayew', 'age'=>32];
14 });
15 // Creating Route for EmployeeAPI
16 use App\Http\Controllers\ApiController;
17 Route::get('employeeapi',[ApiController::class,"APIEmployee"]);
18 Route::post('add-employeeapi',[ApiController::class,'addemployee']);
19 Route::put('update-employee',[ApiController::class,'UpdateEmployee']);
20
21

Sending Update Request



Output (Database Table)

	id	name	email	phone	created_at	updated_at
Edit Copy Delete	1	Yitayew Solomon	yitayewsolomon3@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	2	Haileab Solomon	haileabsolomon@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	3	Yared Solomon	yaredsolomon@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	6	Natanim Yitayew	natan@gmail.com	0937559720	2024-11-12	2024-11-12
Edit Copy Delete	7	Dureti Guye	dure@gmail.com	0910010203	2024-11-13	2024-11-13
Edit Copy Delete	12	yitayew	yitayewsolomon3@gmail.com	000000	2024-11-25	2024-11-25

Check all *With selected:* Edit Copy Delete Export

PATCH Method

4. PATCH

- **Purpose**: Update specific fields of an existing resource.
- **Example**: Updating only the email of a user.

php Copy code

```php
Route::patch('/users/{id}', [UserController::class, 'update']); // Partial update
```

DELETE Method

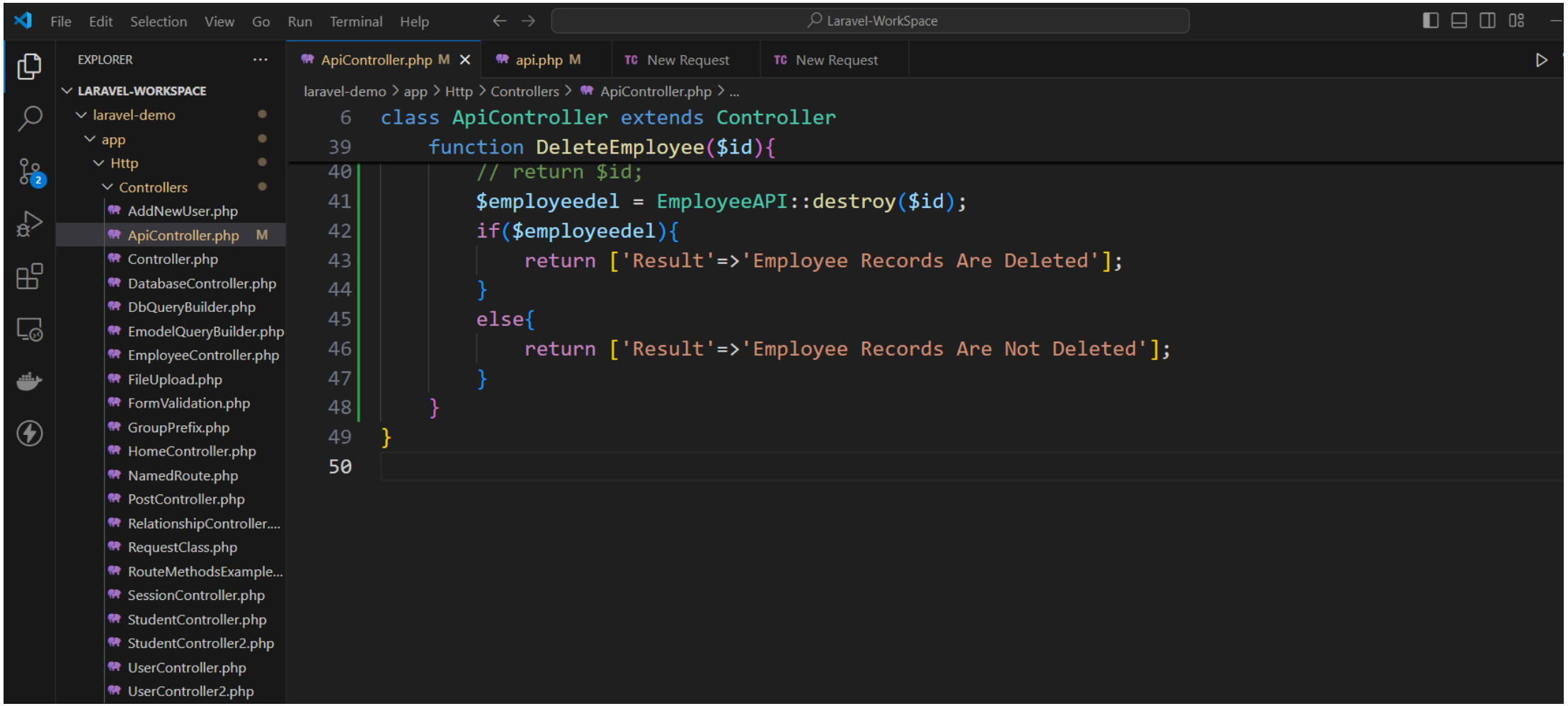
5. DELETE

- **Purpose:** Delete a resource from the server.
- **Example:** Removing a user from the database.

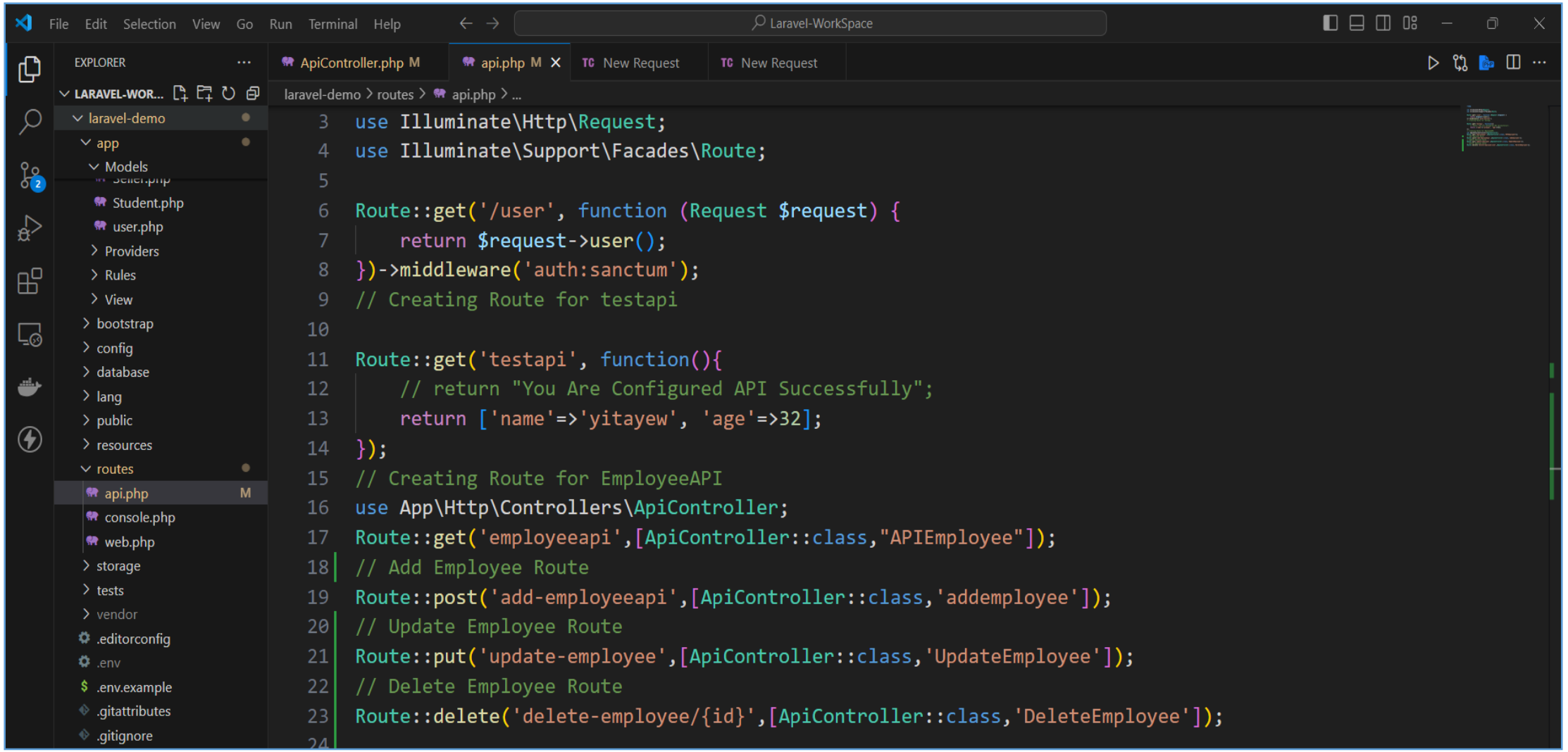
php Copy code

```php
Route::delete('/users/{id}', [UserController::class, 'destroy']); // Delete a user
```

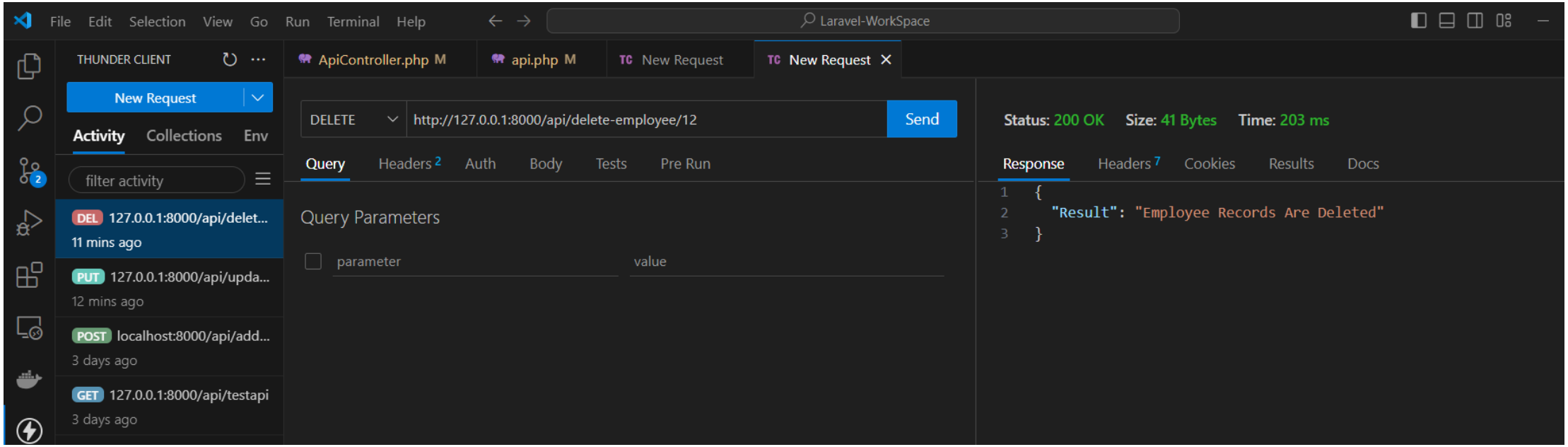
Example

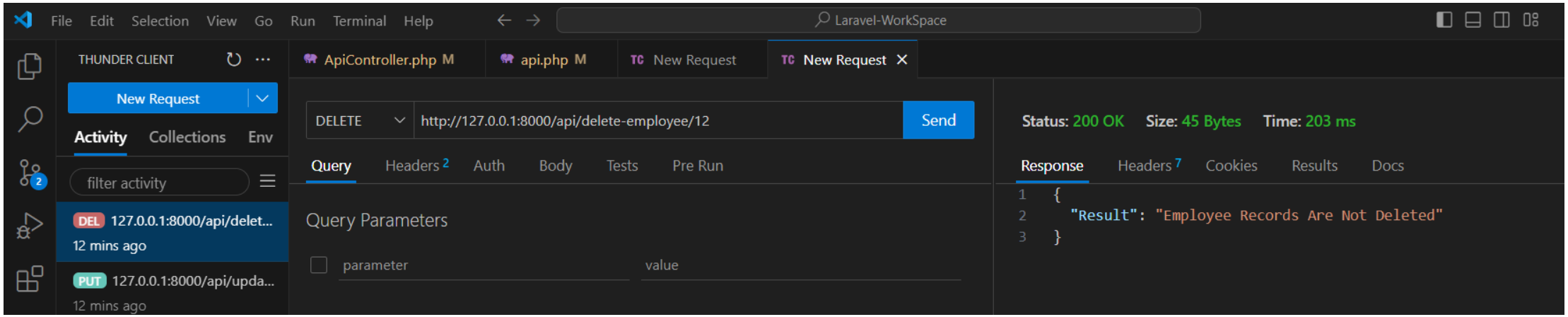

File Edit Selection View Go Run Terminal Help
Laravel-WorkSpace
EXPLORER
LARAVEL-WORKSPACE
laravel-demo
app
Http
Controllers
AddNewUser.php
ApiController.php M
Controller.php
DatabaseController.php
DbQueryBuilder.php
EmodelQueryBuilder.php
EmployeeController.php
FileUpload.php
FormValidation.php
GroupPrefix.php
HomeController.php
NamedRoute.php
PostController.php
RelationshipController....
RequestClass.php
RouteMethodsExample...
SessionController.php
StudentController.php
StudentController2.php
UserController.php
UserController2.php
ApiController.php M
api.php M
New Request
New Request
laravel-demo > app > Http > Controllers > ApiController.php > ...
6 class ApiController extends Controller
39 function DeleteEmployee($id){
40 // return $id;
41 $employeedel = EmployeeAPI::destroy($id);
42 if($employeedel){
43 return ['Result'=>'Employee Records Are Deleted'];
44 }
45 else{
46 return ['Result'=>'Employee Records Are Not Deleted'];
47 }
48 }
49 }
50

Route


File Edit Selection View Go Run Terminal Help
Laravel-WorkSpace
EXPLORER
LARAVEL-WOR...
laravel-demo
app
Models
Student.php
user.php
Providers
Rules
View
bootstrap
config
database
lang
public
resources
routes
api.php M
console.php
web.php
storage
tests
vendor
.editorconfig
.env
.env.example
.gitattributes
.gitignore
ApiController.php M
api.php M
New Request
New Request
laravel-demo > routes > api.php > ...
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
Route::get('/user', function (Request $request) {
return $request->user();
})->middleware('auth:sanctum');
// Creating Route for testapi
Route::get('testapi', function(){
// return "You Are Configured API Successfully";
return ['name'=>'yitayew', 'age'=>32];
});
// Creating Route for EmployeeAPI
use App\Http\Controllers\ApiController;
Route::get('employeeapi',[ApiController::class,"APIEmployee"]);
// Add Employee Route
Route::post('add-employeeapi',[ApiController::class,'addemployee']);
// Update Employee Route
Route::put('update-employee',[ApiController::class,'UpdateEmployee']);
// Delete Employee Route
Route::delete('delete-employee/{id}',[ApiController::class,'DeleteEmployee']);

Sending Delete Request





Output (Database Table)

	id	name	email	phone	created_at	updated_at
Edit Copy Delete	1	Yitayew Solomon	yitayewsolomon3@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	2	Haileab Solomon	haileabsolomon@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	3	Yared Solomon	yaredsolomon@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	6	Natanim Yitayew	natan@gmail.com	0937559720	2024-11-12	2024-11-12
Edit Copy Delete	7	Dureti Guye	dure@gmail.com	0910010203	2024-11-13	2024-11-13

Check all *With selected:* Edit Copy Delete Export

OPTION Method

6. OPTIONS

- **Purpose**: Retrieve communication options available for the target resource.
- **Example**: Typically used in CORS (Cross-Origin Resource Sharing) preflight requests.

php Copy code

```php
Route::options('/users', function () {
    return response()->json(['GET', 'POST', 'OPTIONS']);
});
```

HEAD Method

7. HEAD

- **Purpose**: Retrieve metadata about a resource without the actual content.
- **Example**: Checking headers to verify if a resource exists.

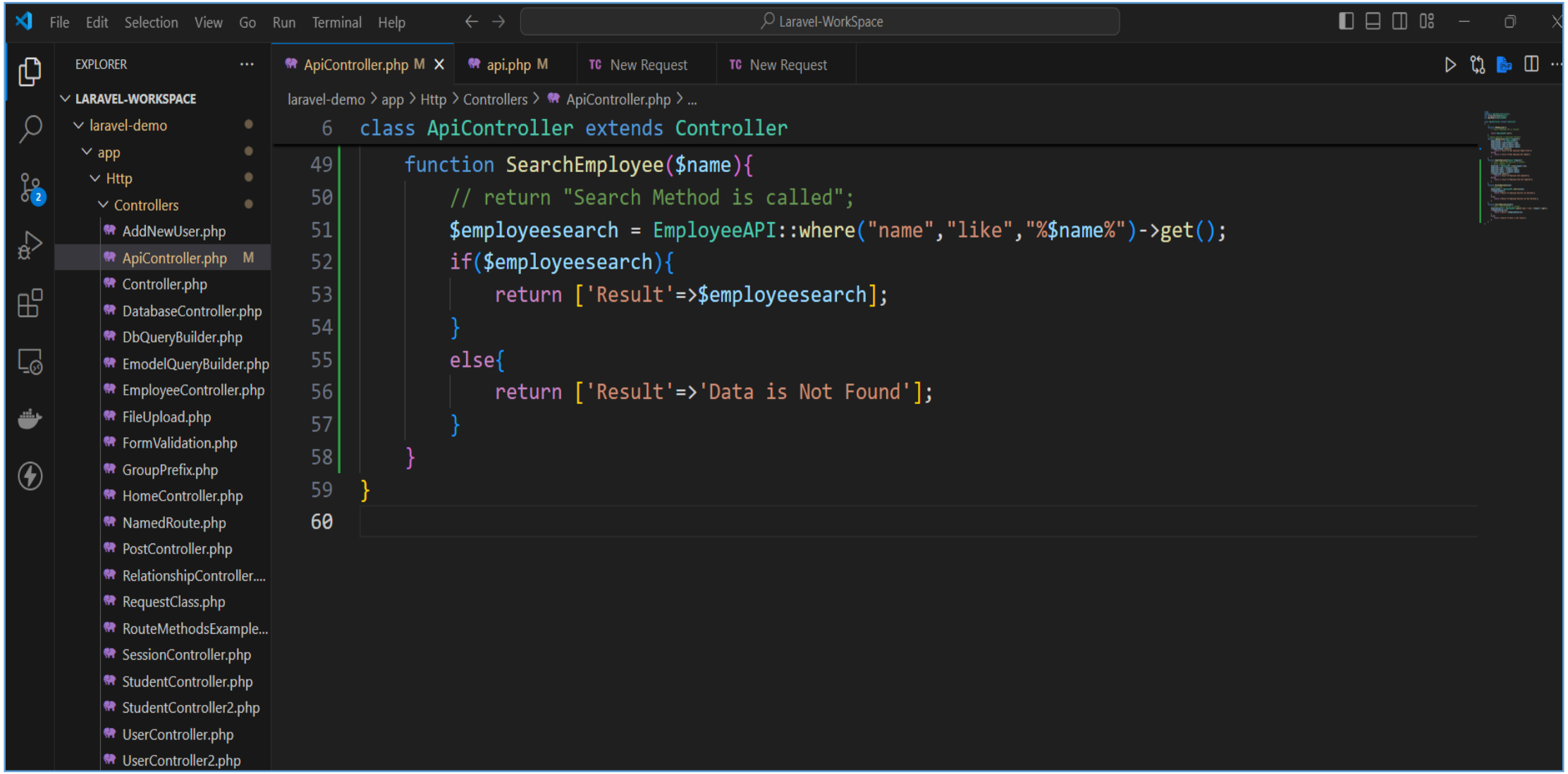
php Copy code

```php
Route::get('/users', [UserController::class, 'index'])->middleware('head');
```

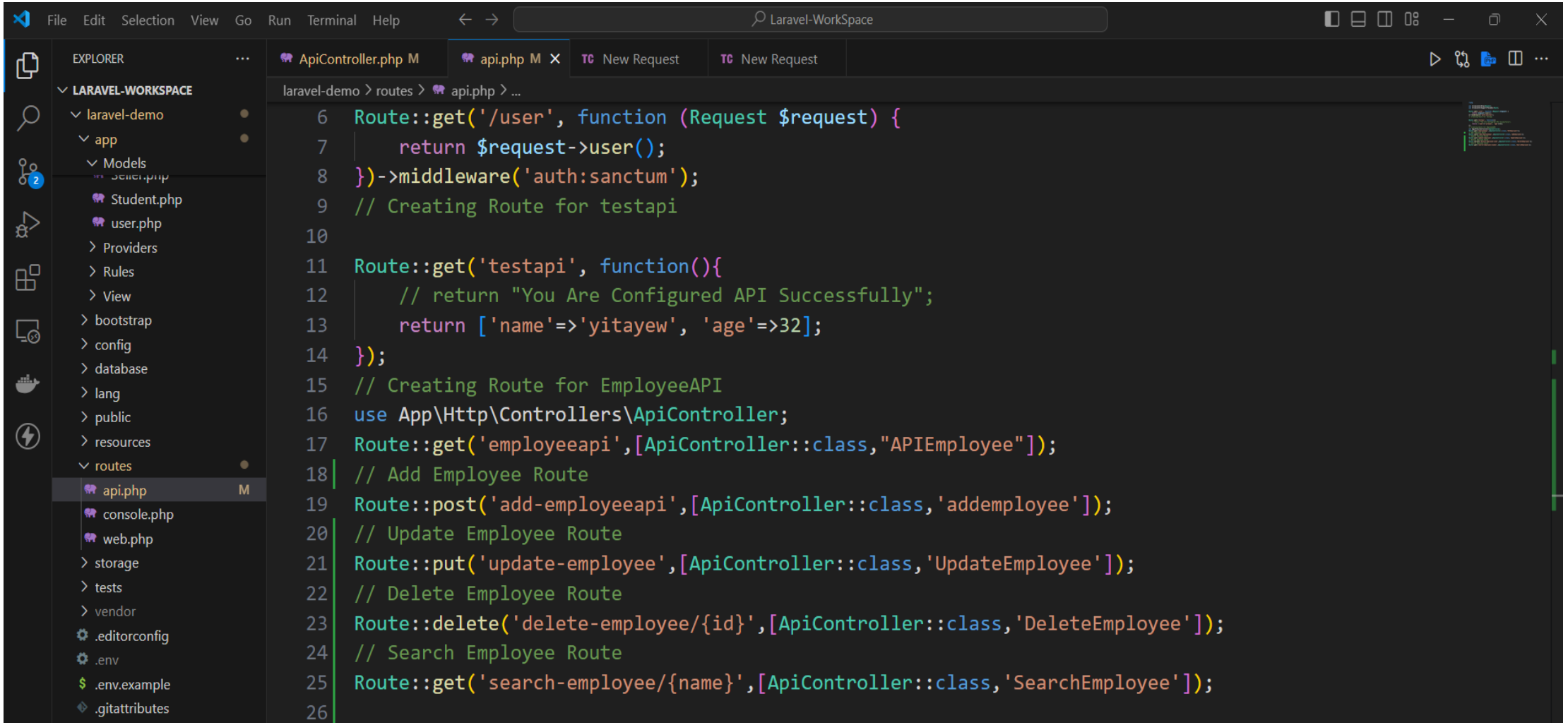
How Laravel Handles These Methods

Laravel uses **Route definitions** to handle these methods. The `Route` facade supports methods for each HTTP verb, as shown above.

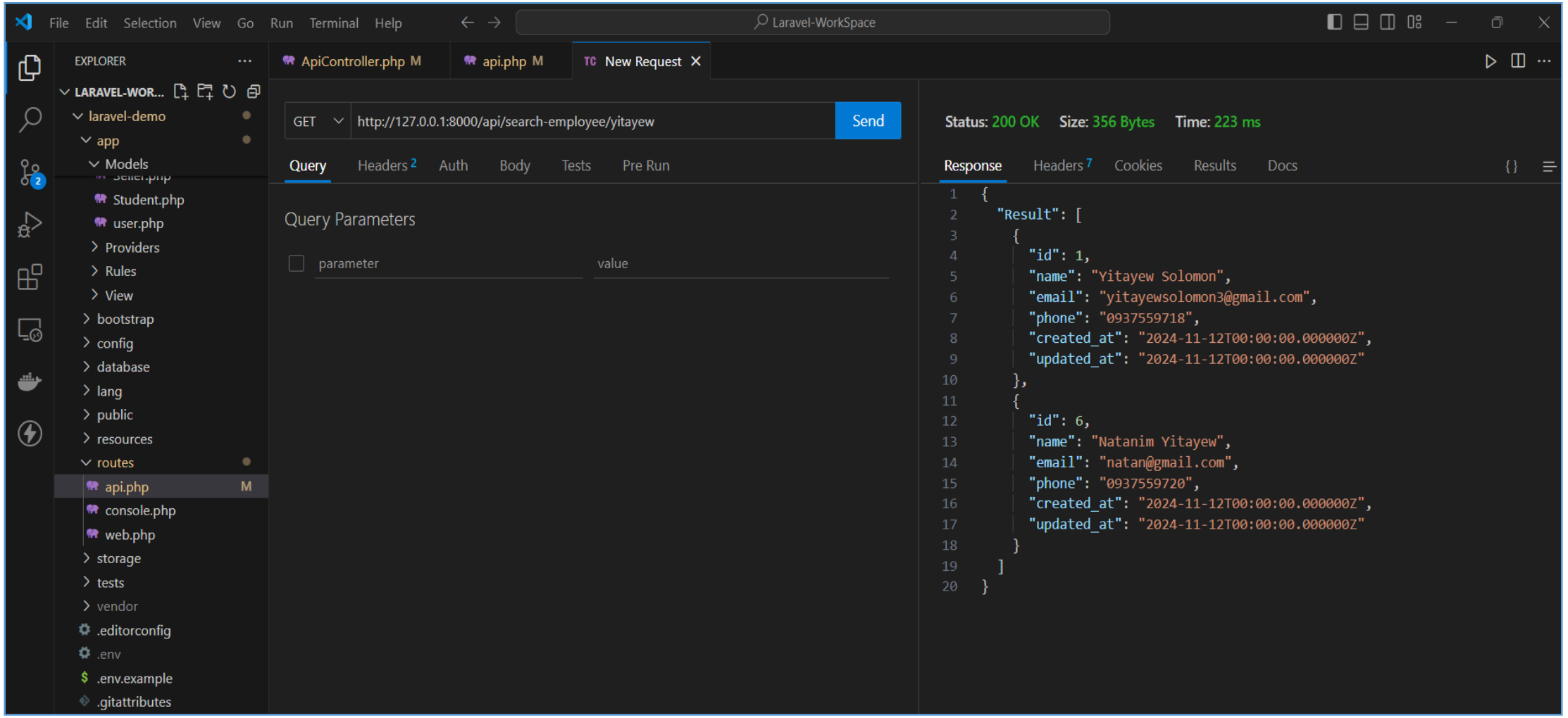
Search API


File Edit Selection View Go Run Terminal Help
Laravel-WorkSpace
EXPLORER
LARAVEL-WORKSPACE
laravel-demo
app
Http
Controllers
AddNewUser.php
ApiController.php M
Controller.php
DatabaseController.php
DbQueryBuilder.php
EmodelQueryBuilder.php
EmployeeController.php
FileUpload.php
FormValidation.php
GroupPrefix.php
HomeController.php
NamedRoute.php
PostController.php
RelationshipController....
RequestClass.php
RouteMethodsExample...
SessionController.php
StudentController.php
StudentController2.php
UserController.php
UserController2.php
ApiController.php M
api.php M
New Request
New Request
laravel-demo > app > Http > Controllers > ApiController.php > ...
6 class ApiController extends Controller
49 function SearchEmployee($name){
50 // return "Search Method is called";
51 $employeesearch = EmployeeAPI::where("name","like","%$name%")->get();
52 if($employeesearch){
53 return ['Result'=>$employeesearch];
54 }
55 else{
56 return ['Result'=>'Data is Not Found'];
57 }
58 }
59 }
60

Route


File Edit Selection View Go Run Terminal Help
Laravel-WorkSpace
EXPLORER
LARAVEL-WORKSPACE
laravel-demo
app
Models
Student.php
user.php
Providers
Rules
View
bootstrap
config
database
lang
public
resources
routes
api.php M
console.php
web.php
storage
tests
vendor
.editorconfig
.env
.env.example
.gitattributes
ApiController.php M
api.php M
New Request
New Request
laravel-demo > routes > api.php > ...
6 Route::get('/user', function (Request $request) {
7 return $request->user();
8 })->middleware('auth:sanctum');
9 // Creating Route for testapi
10
11 Route::get('testapi', function(){
12 // return "You Are Configured API Successfully";
13 return ['name'=>'yitayew', 'age'=>32];
14 });
15 // Creating Route for EmployeeAPI
16 use App\Http\Controllers\ApiController;
17 Route::get('employeeapi',[ApiController::class,"APIEmployee"]);
18 // Add Employee Route
19 Route::post('add-employeeapi',[ApiController::class,'addemployee']);
20 // Update Employee Route
21 Route::put('update-employee',[ApiController::class,'UpdateEmployee']);
22 // Delete Employee Route
23 Route::delete('delete-employee/{id}',[ApiController::class,'DeleteEmployee']);
24 // Search Employee Route
25 Route::get('search-employee/{name}',[ApiController::class,'SearchEmployee']);
26

Sending Search Request



Output (Database Table)

	id	name	email	phone	created_at	updated_at
Edit Copy Delete	1	Yitayew Solomon	yitayewsolomon3@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	2	Haileab Solomon	haileabsolomon@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	3	Yared Solomon	yaredsolomon@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	6	Natanim Yitayew	natan@gmail.com	0937559720	2024-11-12	2024-11-12
Edit Copy Delete	7	Dureti Guye	dure@gmail.com	0910010203	2024-11-13	2024-11-13

Check all *With selected:* Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Summary

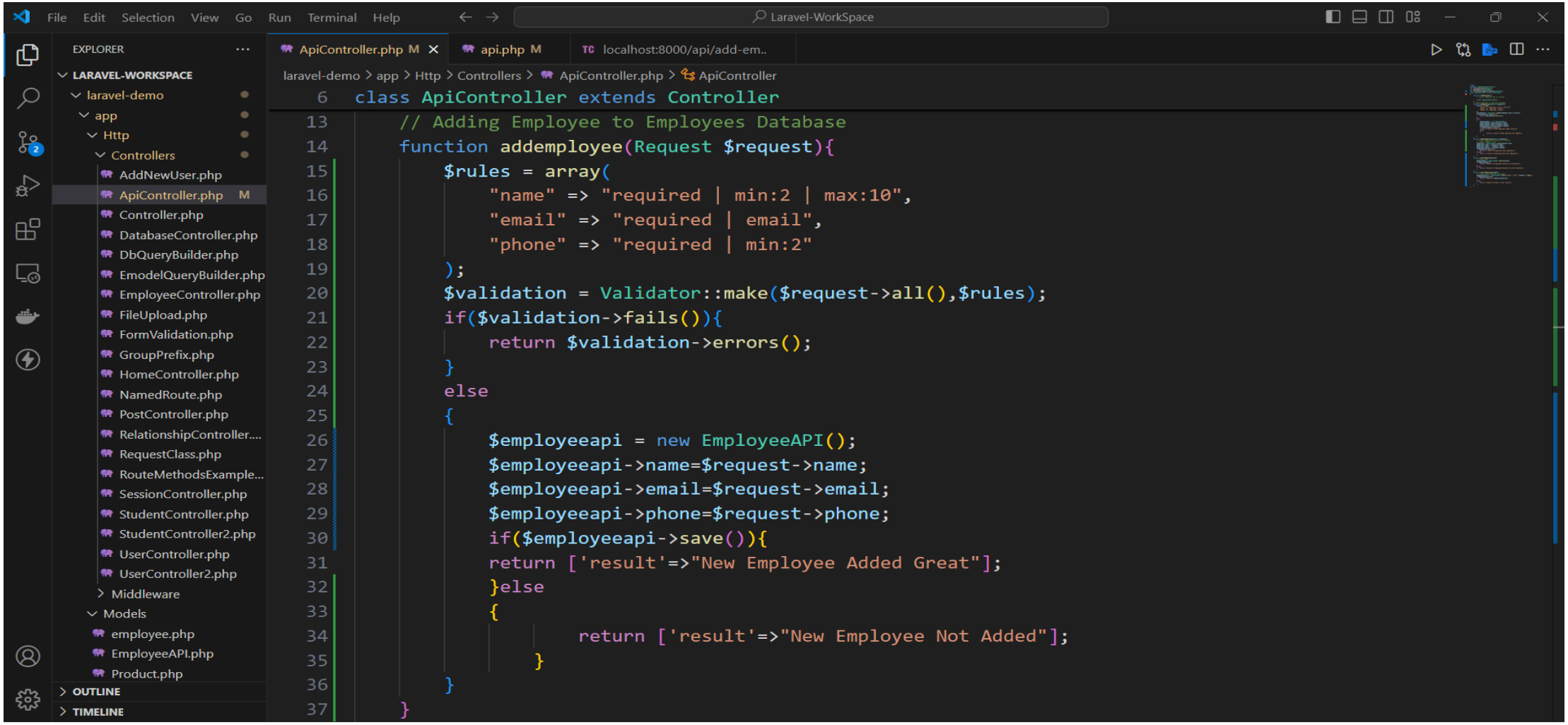
Summary Table of HTTP Methods

HTTP Method	Purpose	Laravel Route Example	Controller Method
GET	Retrieve data	`Route::get('/users')`	`index()` / `show()`
POST	Create a new resource	`Route::post('/users')`	`store()`
PUT	Update a resource fully	`Route::put('/users/{id}')`	`update()`
PATCH	Update specific fields	`Route::patch('/users/{id}')`	`update()`
DELETE	Remove a resource	`Route::delete('/users/{id}')`	`destroy()`
OPTIONS	Fetch communication options	`Route::options('/users')`	N/A
HEAD	Fetch resource metadata	`Route::get('/users')->middleware('head')`	N/A

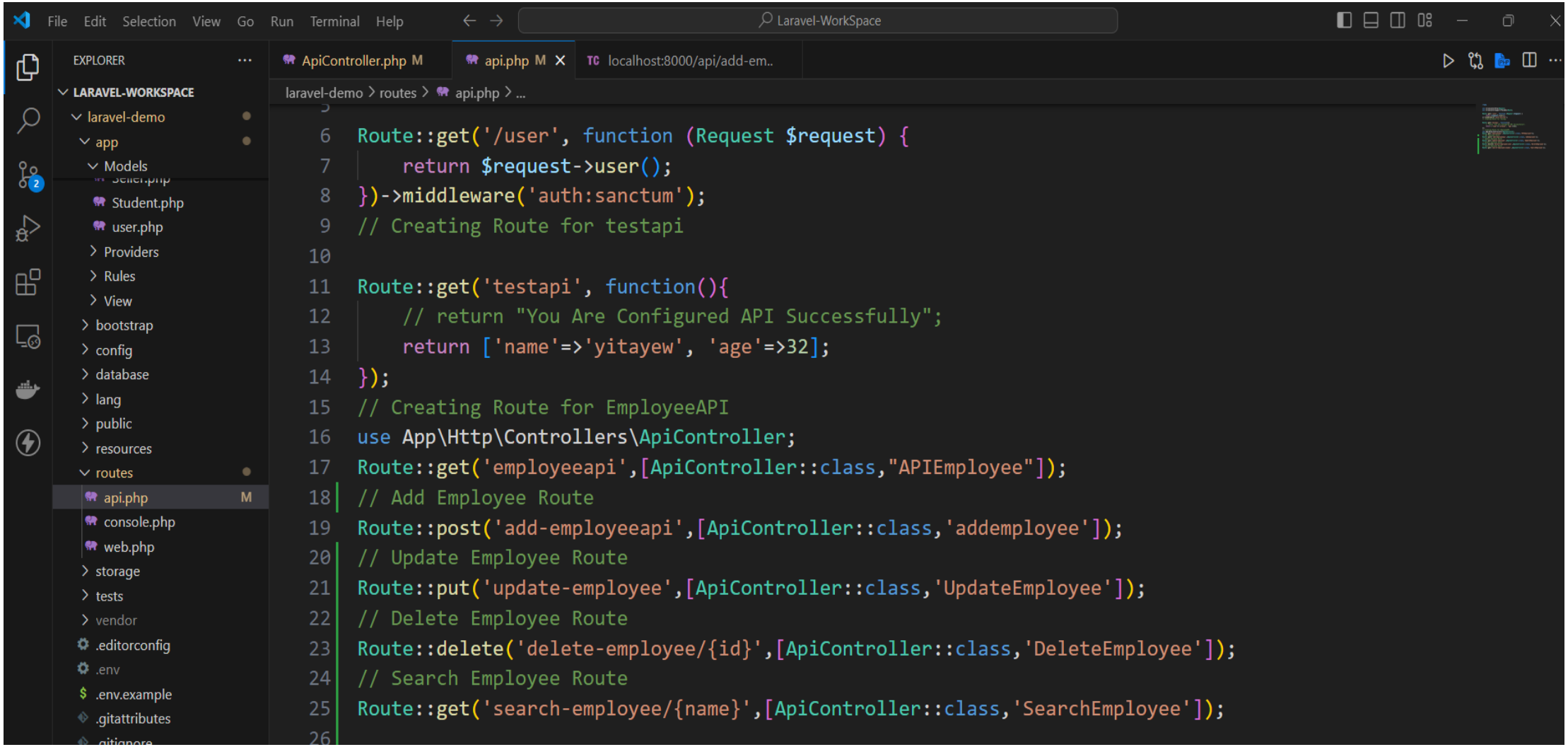
API Validation

- API validation in Laravel is used to ensure that the data provided in API requests meets specific rules and constraints before processing the request.
- Laravel provides a clean and robust way to validate API data using the validate method or form request classes.

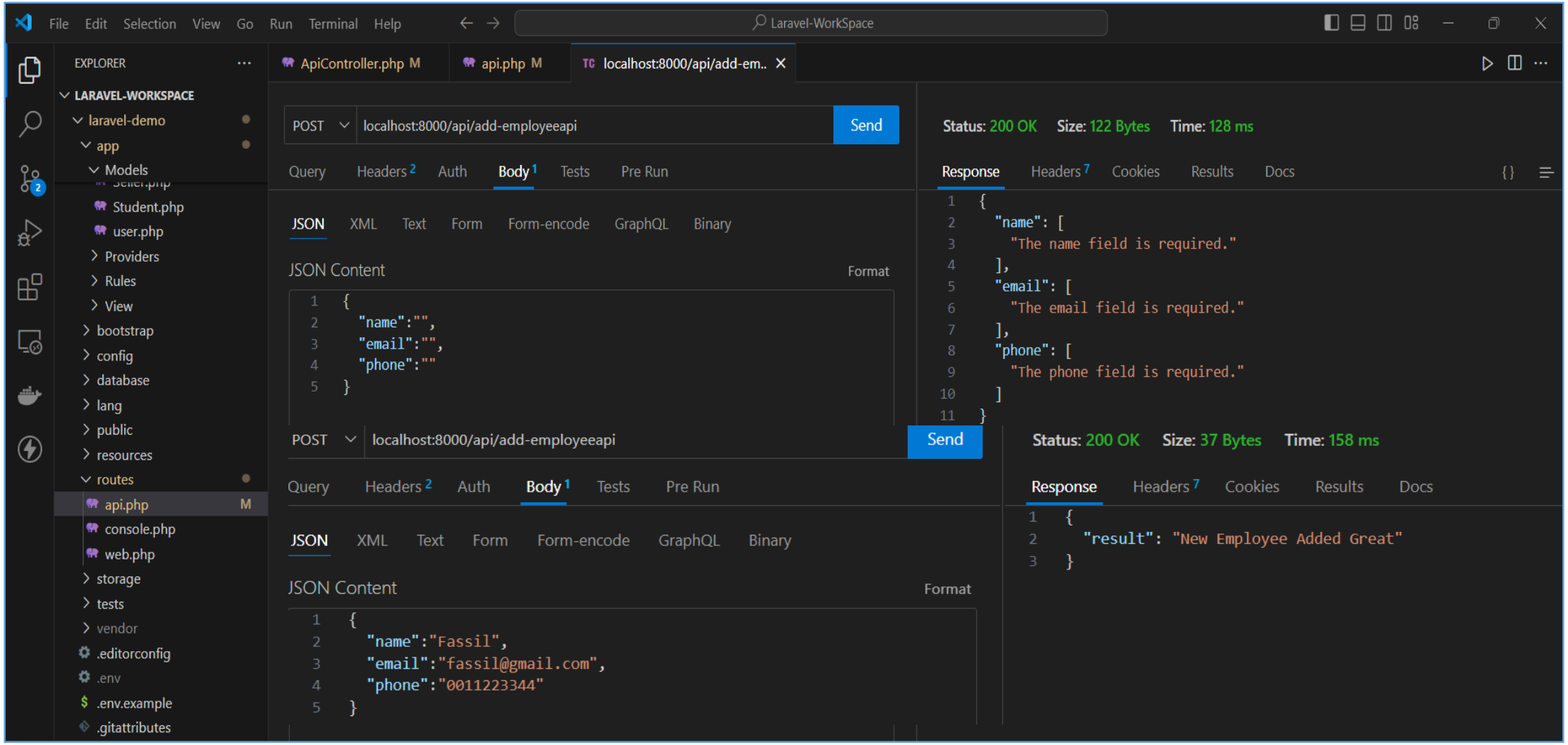
Example Applying Validation on add Employee



Route


File Edit Selection View Go Run Terminal Help
Laravel-WorkSpace
EXPLORER
LARAVEL-WORKSPACE
laravel-demo
app
Models
Student.php
user.php
Providers
Rules
View
bootstrap
config
database
lang
public
resources
routes
api.php M
console.php
web.php
storage
tests
vendor
.editorconfig
.env
.env.example
.gitattributes
ApiController.php M
api.php M
localhost:8000/api/add-em..
laravel-demo > routes > api.php > ...
6 Route::get('/user', function (Request $request) {
7 return $request->user();
8 })->middleware('auth:sanctum');
9 // Creating Route for testapi
10
11 Route::get('testapi', function(){
12 // return "You Are Configured API Successfully";
13 return ['name'=>'yitayew', 'age'=>32];
14 });
15 // Creating Route for EmployeeAPI
16 use App\Http\Controllers\ApiController;
17 Route::get('employeeapi',[ApiController::class,"APIEmployee"]);
18 // Add Employee Route
19 Route::post('add-employeeapi',[ApiController::class,'addemployee']);
20 // Update Employee Route
21 Route::put('update-employee',[ApiController::class,'UpdateEmployee']);
22 // Delete Employee Route
23 Route::delete('delete-employee/{id}',[ApiController::class,'DeleteEmployee']);
24 // Search Employee Route
25 Route::get('search-employee/{name}',[ApiController::class,'SearchEmployee']);
26

Sending Add Request



Output (Database Table)

	id	name	email	phone	created_at	updated_at
Edit Copy Delete	1	Yitayew Solomon	yitayewsolomon3@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	2	Haileab Solomon	haileabsolomon@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	3	Yared Solomon	yaredsolomon@gmail.com	0937559718	2024-11-12	2024-11-12
Edit Copy Delete	6	Natanim Yitayew	natan@gmail.com	0937559720	2024-11-12	2024-11-12
Edit Copy Delete	7	Dureti Guye	dure@gmail.com	0910010203	2024-11-13	2024-11-13
Edit Copy Delete	13	Fassil	fasil@gmail.com	112233	2024-11-25	2024-11-25
Edit Copy Delete	15	Fassil	fassil@gmail.com	0011223344	2024-11-25	2024-11-25

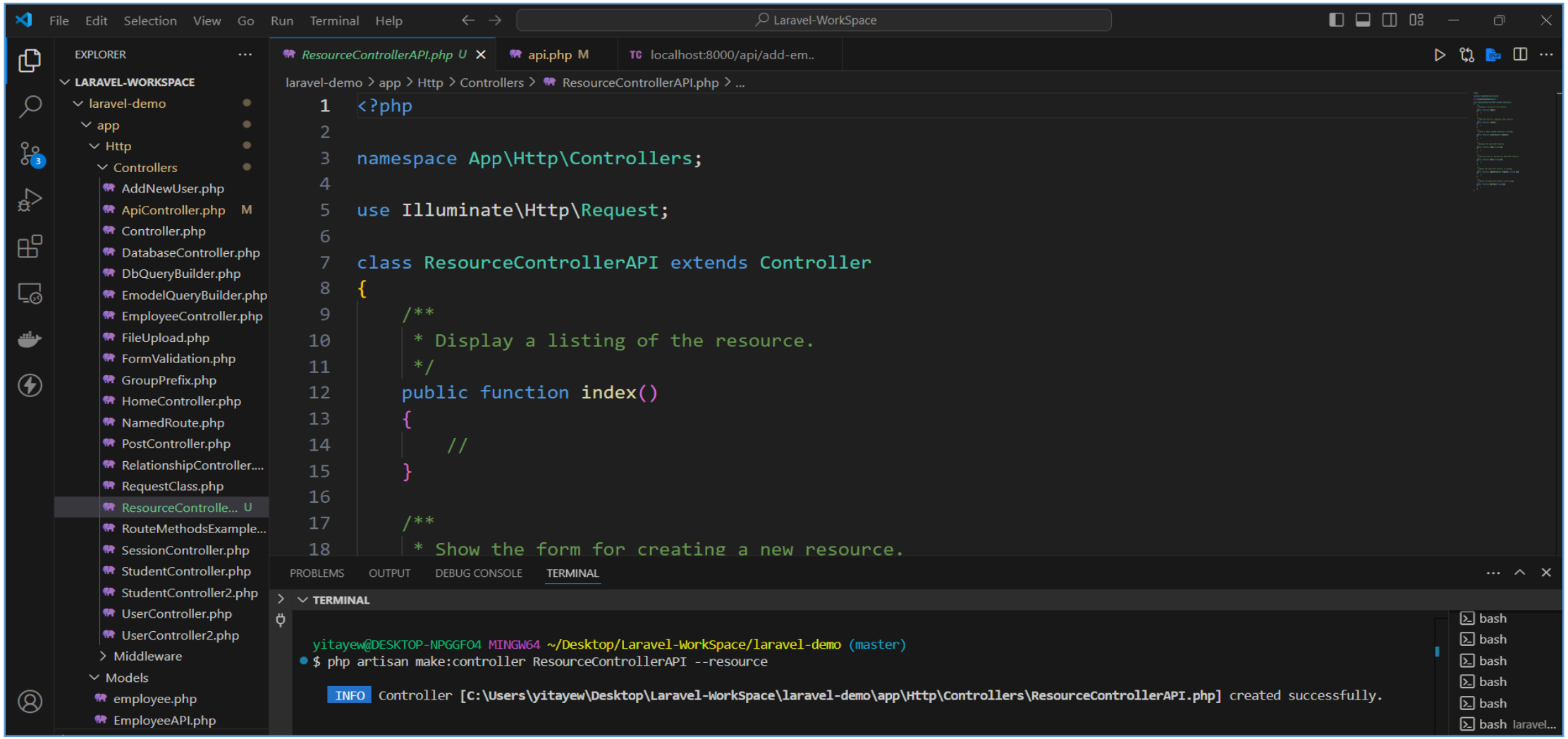
Check all *With selected:* Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

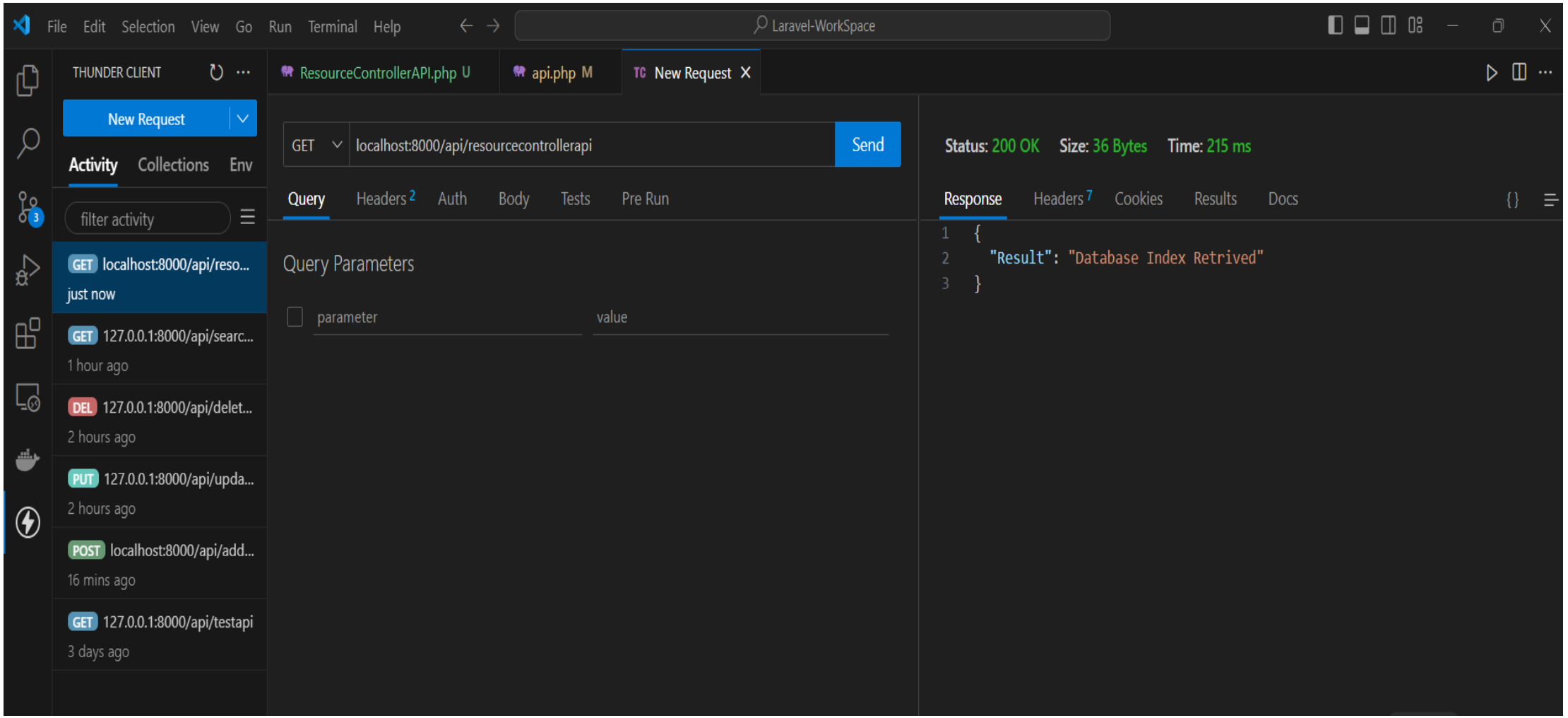
API with Resource Controller

- Using a Resource Controller in Laravel for API development is an efficient way to handle standard CRUD operations. A Resource Controller provides predefined methods corresponding to common actions like creating, retrieving, updating, and deleting records.

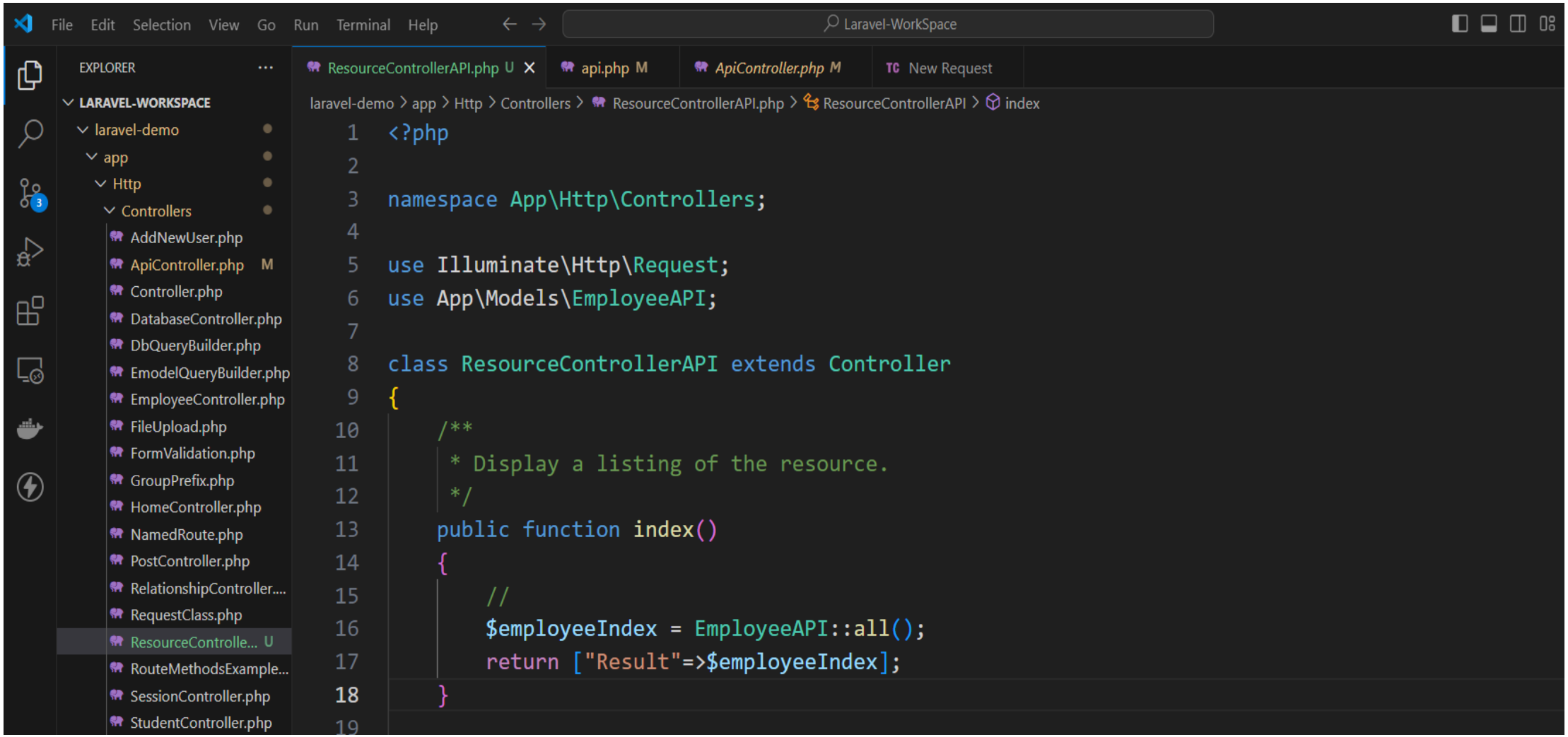
Example


File Edit Selection View Go Run Terminal Help
Laravel-WorkSpace
EXPLORER
LARAVEL-WORKSPACE
laravel-demo
app
Http
Controllers
AddNewUser.php
ApiController.php M
Controller.php
DatabaseController.php
DbQueryBuilder.php
EmodelQueryBuilder.php
EmployeeController.php
FileUpload.php
FormValidation.php
GroupPrefix.php
HomeController.php
NamedRoute.php
PostController.php
RelationshipController....
RequestClass.php
ResourceControlle... U
RouteMethodsExample...
SessionController.php
StudentController.php
StudentController2.php
UserController.php
UserController2.php
Middleware
Models
employee.php
EmployeeAPI.php
ResourceControllerAPI.php U
api.php M
localhost:8000/api/add-em..
laravel-demo > app > Http > Controllers > ResourceControllerAPI.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class ResourceControllerAPI extends Controller
8 {
9 /**
10 * Display a listing of the resource.
11 */
12 public function index()
13 {
14 //
15 }
16
17 /**
18 * Show the form for creating a new resource.
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
TERMINAL
yitayew@DESKTOP-NPGGFO4 MINGW64 ~/Desktop/Laravel-WorkSpace/laravel-demo (master)
$ php artisan make:controller ResourceControllerAPI --resource
INFO Controller [C:\Users\yitayew\Desktop\Laravel-WorkSpace\laravel-demo\app\Http\Controllers\ResourceControllerAPI.php] created successfully.
bash
bash
bash
bash
bash
bash laravel...

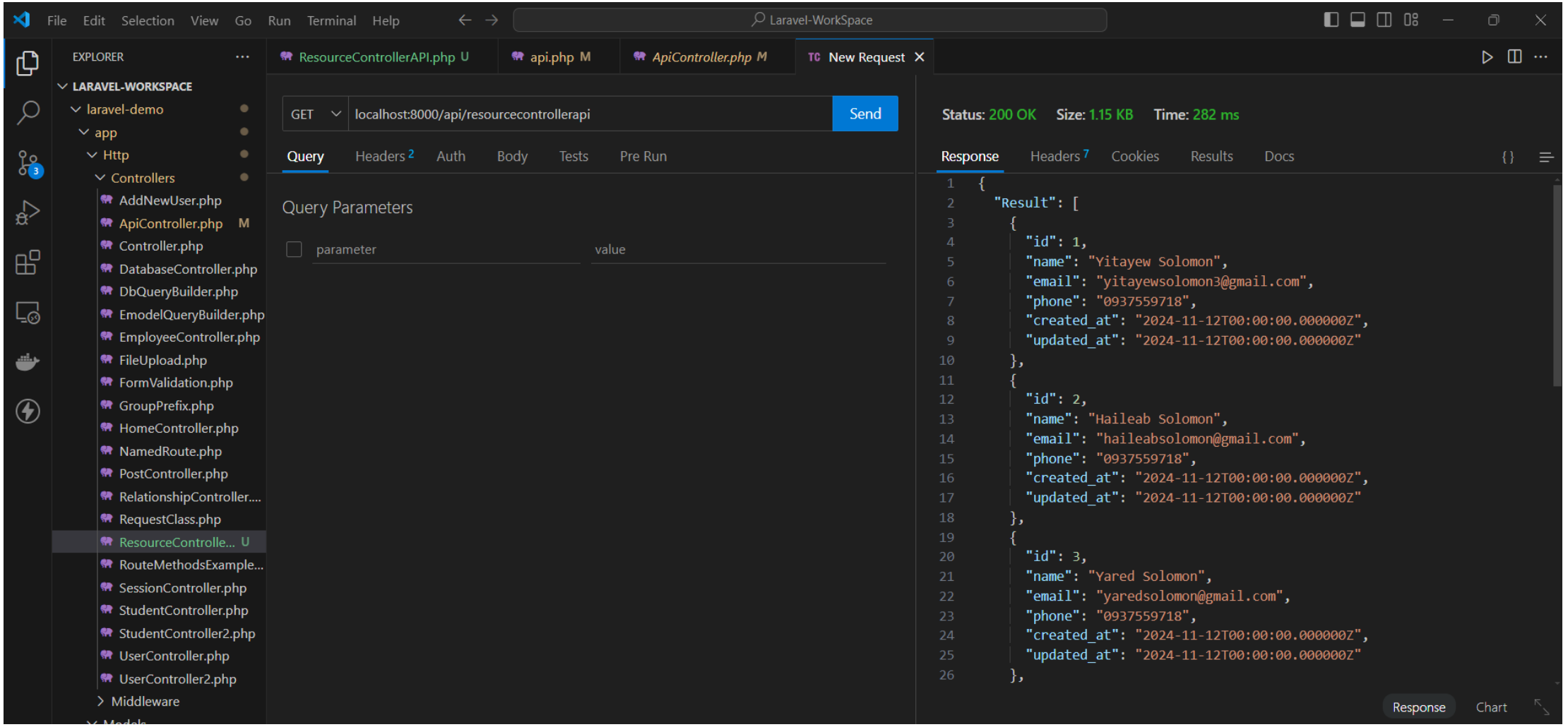
Sending Index Request



Sending Index Request



Sending Index Request



Thank you!

Appreciate your action.