**Laravel Curd API Documentation**

This documentation provides an overview of the **CRUD API** for managing products, including routes, request methods, expected responses, and integration with Blade views.

This API allows users to **create, retrieve, update, and delete products** in a MySQL database. Additionally, a Blade template is used to **display, edit, and delete products** via a web interface.

**Step 1**

Install a new laravel project with the name product-api

**composer create-project laravel/laravel product-api**

**Step 2**

Configure the Database in .env file

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=api\_db

DB\_USERNAME=root

DB\_PASSWORD=

**Step 3**

Create Product Model & Migration with the command is-

*php artisan make:model Product -m*

**Set the field according to the requirement in the migration file database/migrations/2025\_03\_27\_044504\_create\_products\_table.php**

**public function up(): void**

**{**

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

**$table->id();**

**$table->string('name');**

**$table->string('description');**

**$table->decimal('price', 10, 2);**

**$table->integer('stock')->default(0);**

**$table->string('image')->nullable();**

**$table->timestamps();**

**});**

**}**

**Then, run the command to create the table in the database**

***php artisan migrate***

**Step 4**

Create Controller with the command -

*php artisan make:controller ProductController --api*

Implement CRUD in app/Http/Controllers/ProductController.php

\* Display a listing of the resource.

\*/

public function index()

{

return response()->json(Product::all());

}

/\*\*

\* Store a newly created resource in storage.

\*/

public function store(Request $request)

{

$product = Product::create($request->all());

return response()->json($product, 201);

}

/\*\*

\* Display the specified resource.

\*/

public function show(string $id)

{

return response()->json(Product::findOrFail($id));

}

/\*\*

\* Update the specified resource in storage.

\*/

public function update(Request $request, string $id)

{

$product = Product::findOrFail($id);

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

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

}

/\*\*

\* Remove the specified resource from storage.

\*/

public function destroy(string $id)

{

Product::findOrFail($id)->delete();

return response()->json(null, 204);

}

**Step 5**

* Install the api with the command -

*php artisan install:api*

* Define API Routes in routes/api.php

use App\Http\Controllers\ProductController;

Route::apiResource('products', ProductController::class);

### **Test API with Postman**

* GET: /api/products
* POST: /api/products
* GET: /api/products/{id}
* PUT: /api/products/{id}
* DELETE: /api/products/{id}

**To Add Edit & Delete the API data with Blade file in Laravel, follow these steps:**

**Step 1**

Create function to Add Edit & Delete the API data inController

public function showProducts()

{

$products = Product::all(); // Fetch all products

return view('products', compact('products')); // Pass data to the view

}

public function addproduct(Request $request)

{

$product = Product::create($request->all());

return redirect('/products')->with('success', 'Product added successfully!');

}

public function delete($id)

{

$product = Product::findOrFail($id);

$product->delete();

return redirect('/products')->with('success', 'Product deleted successfully!');

}

// Show Edit Form

public function editProduct($id)

{

$product = Product::findOrFail($id);

return view('edit', compact('product'));

}

// Update Product

public function updateProduct(Request $request, $id)

{

$product = Product::findOrFail($id);

$product->update([

'name' => $request->name,

'description' => $request->description,

'price' => $request->price,

'stock' => $request->stock,

]);

return redirect('/products')->with('success', 'Product updated successfully!');

}

**Step 2**

Define Route in *route/web.php*

*Route::get('/products', [ProductController::class, 'showProducts']);*

*Route::get('/products', [ProductController::class, 'showProducts']);*

*Route::get('/products/edit/{id}', [ProductController::class, 'editProduct']);*

*Route::post('/products/update/{id}', [ProductController::class, 'updateProduct']);*

*Route::post('/products/delete/{id}', [ProductController::class, 'deleteProduct']);*

**Step 3**

* Create blade file to show all the products stored in database  *product.blade.php*

<!DOCTYPE html>

<html>

<head>

<title>Product List</title>

<!-- Latest compiled and minified CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

<!-- Latest compiled JavaScript -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-">

<h1> @if (session('success'))

<div class="alert alert-success">

{{ session('success') }}

</div>

@endif

</h1>

<h1>Product List</h1>

<button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">Add new

Product</button>

<table class="table table-bordered mt-4">

<tr>

<th>ID</th>

<th>Name</th>

<th>Description</th>

<th>Price</th>

<th>Stock</th>

<th>Action</th>

</tr>

@foreach($products as $product)

<tr>

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

<td>{{ $product->name }}</td>

<td>{{ $product->description }}</td>

<td>${{ $product->price }}</td>

<td>{{ $product->stock }}</td>

<td>

<form action="{{ url('/products/delete/' . $product->id) }}" method="POST"

style="display:inline;">

@csrf

<button class="btn btn-danger" type="submit" onclick="return confirm('Are you sure?')">Delete</button>

</form> / <a class="btn btn-primary" href="{{ url('/products/edit/' . $product->id) }}">Edit</a>

</td>

</tr>

@endforeach

</table>

<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel"

aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">Add Product</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal"

aria-label="Close"></button>

</div>

<div class="modal-body">

<form action="products" method="post" enctype="multipart/form-data">

@csrf

<div class="form-group mb-3">

<input type="text" name="name" value="" class="form-control"

placeholder="Product Name">

</div>

<div class="form-group mb-3">

<input type="text" name="description" value="" class="form-control"

placeholder="desription">

</div>

<div class="form-group mb-3">

<input type="text" name="price" value="" class="form-control"

placeholder="price">

</div>

<div class="form-group mb-3">

<input type="text" name="stock" value="" class="form-control"

placeholder="stock">

</div>

<div class="form-group mb-3">

<input type="file" name="image" value="" class="form-control">

</div>

<div class="form-group mb-3">

<input type="submit" name="submit" value="submit" class="form-control">

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

Create balde file to edit the product details *edit.blade.php*

<!DOCTYPE html>

<html>

<head>

<title>Edit Product</title>

<!-- Latest compiled and minified CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

<!-- Latest compiled JavaScript -->

,.<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-12">

<h1>Edit Product</h1>

<form action="{{ url('/products/update/' . $product->id) }}" method="post" enctype="multipart/form-data">

@csrf

<div class="form-group mb-3">

<input type="text" name="name" value="{{ $product->name }}" class="form-control"

placeholder="Product Name">

</div>

<div class="form-group mb-3">

<input type="text" name="description" value="{{ $product->description }}" class="form-control"

placeholder="desription">

</div>

<div class="form-group mb-3">

<input type="text" name="price" value="{{ $product->price }}" class="form-control"

placeholder="price">

</div>

<div class="form-group mb-3">

<input type="text" name="stock" value="{{ $product->stock }}" class="form-control"

placeholder="stock">

</div>

<div class="form-group mb-3">

<input type="submit" name="submit" value="submit" class="form-control">

</div>

</form>

</div>

</div>

</div>

</body>

</html>

**Step 4**

**Test the Application**

Run the app with *-php artisan serve*

View Products: <http://127.0.0.1:8000/products>

* Add Product: Click on the Add product.
* Edit Product: Click on the Edit button.
* Delete Product: Click the Delete button.