API dan React.js

implementasi CRUD (Create, Read, Update, Delete) API menggunakan Laravel dengan database MySQL:

Langkah 1: Membuat Migrasi dan Model Buka terminal Anda dan jalankan perintah-perintah berikut untuk membuat migrasi dan model yang diperlukan:

bash

# Membuat migrasi untuk tabel

php artisan make:migration create\_items\_table --create=items

# Membuat model

php artisan make:model Item

Langkah 2: Mengedit file migrasi Buka file migrasi yang baru dibuat (database/migrations/{timestamp}\_create\_items\_table.php) dan ubah fungsi up() seperti berikut:

php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateItemsTable extends Migration

{

public function up()

{

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

$table->id();

$table->string('name');

$table->text('description');

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('items');

}

}

Langkah 3: Menjalankan migrasi Jalankan perintah berikut di terminal untuk menjalankan migrasi:

bash

php artisan migrate

Langkah 4: Mengedit model Buka file model (app/Models/Item.php) dan tambahkan properti fillable seperti di bawah ini:

php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Item extends Model

{

use HasFactory;

protected $fillable = [

'name',

'description',

];

}

Langkah 5: Membuat Controller Jalankan perintah berikut di terminal untuk membuat controller:

bash

php artisan make:controller ItemController --resource --model=Item

Langkah 6: Mengedit controller Buka file controller yang baru saja dibuat (app/Http/Controllers/ItemController.php) dan tambahkan kode berikut:

php

namespace App\Http\Controllers;

use App\Models\Item;

use Illuminate\Http\Request;

class ItemController extends Controller

{

public function index()

{

$items = Item::all();

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

}

public function store(Request $request)

{

$item = Item::create($request->all());

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

}

public function show($id)

{

$item = Item::findOrFail($id);

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

}

public function update(Request $request, $id)

{

$item = Item::findOrFail($id);

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

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

}

public function destroy($id)

{

$item = Item::findOrFail($id);

$item->delete();

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

}

}

Langkah 7: Mengatur rute Buka file rute (routes/api.php) dan tambahkan kode berikut:

php

use App\Http\Controllers\ItemController;

Route::get('/items', [ItemController::class, 'index']);

Route::post('/items', [ItemController::class, 'store']);

Route::get('/items/{id}', [ItemController::class, 'show']);

Route::put('/items/{id}', [ItemController::class, 'update']);

Route::delete('/items/{id}', [ItemController::class, 'destroy']);

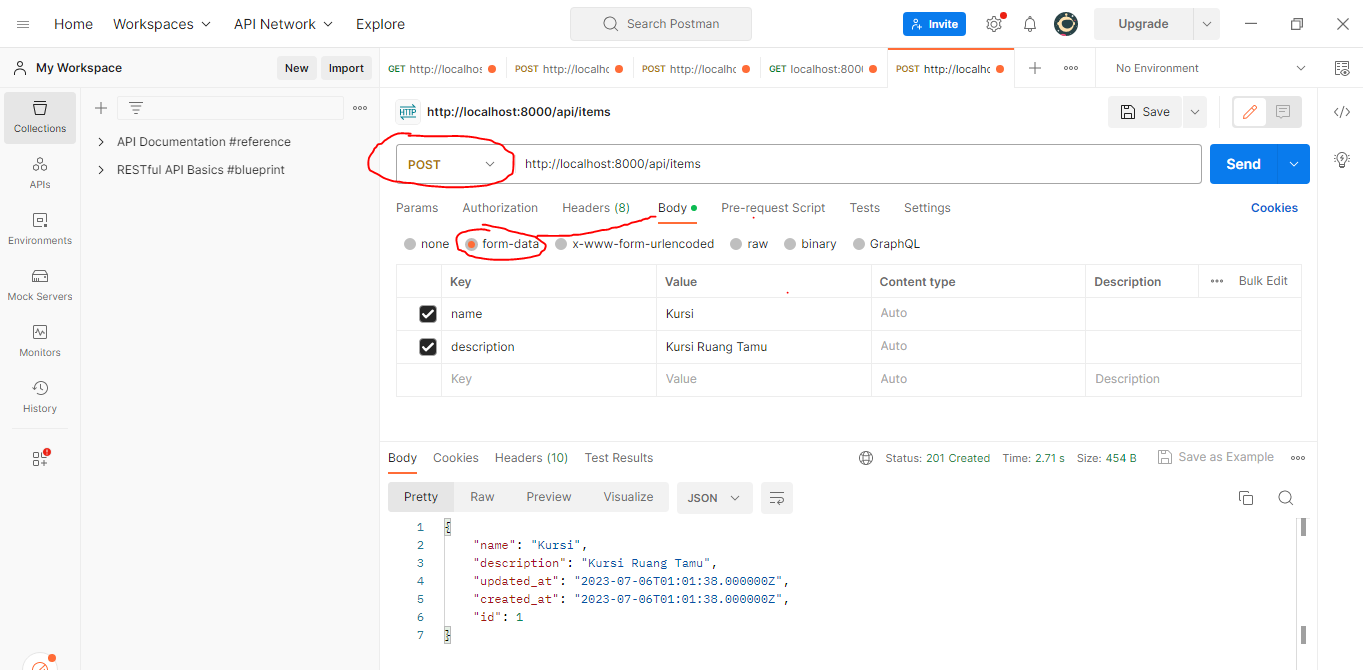
Langkah 8: Menjalankan aplikasi Terakhir, jalankan server PHP Laravel dengan menjalankan perintah berikut di terminal:

bash

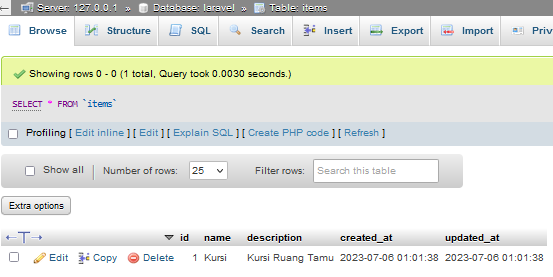
php artisan serve

API CRUD untuk entitas "Item" sekarang sudah tersedia di URL http://localhost:8000/api/items dengan metode HTTP yang sesuai untuk setiap operasi CRUD. Anda dapat menggunakan Postman atau alat serupa untuk menguji endpoint-endpoint tersebut.

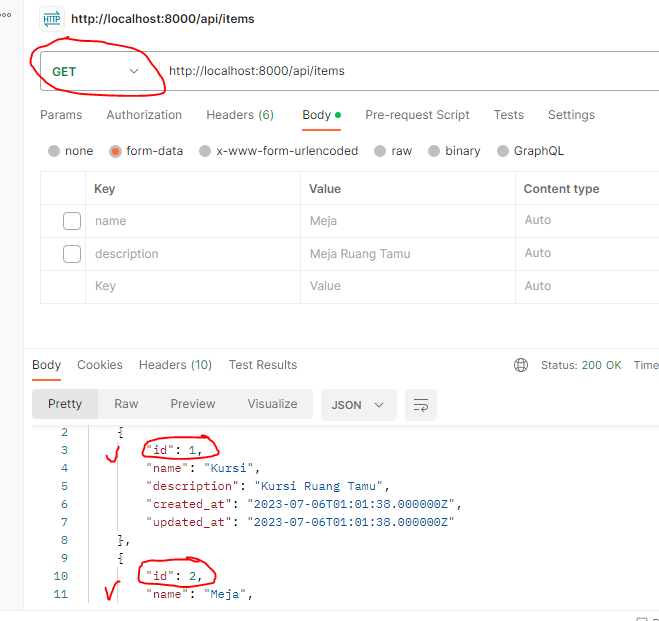
Penting: Pastikan Anda sudah mengonfigurasi koneksi database MySQL di file .env dengan benar sebelum menjalankan migrasi.



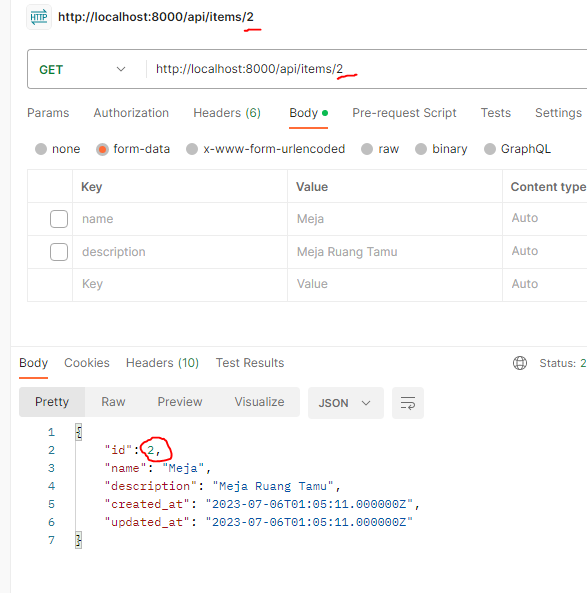
Ketika berhasil data akan masuk dalam Tabel items seperti berikut:



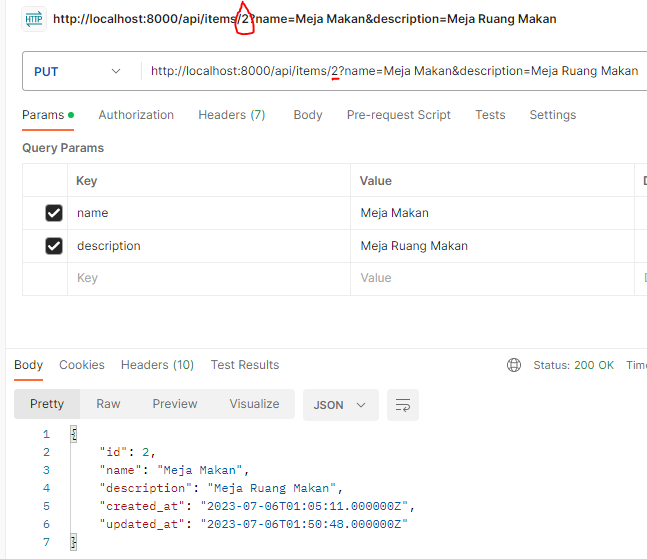
Kemudian coba masukan data lagi dengan post, selanjutnya panggil dengan perintah get pada postman seperti berikut:



Coba tampilkan dengan filter id salah satu data yang telah diinput , jika berhasil maka hanya data yang dipanggil id-nya saja yang muncul:



Coba dengan PUT:



Dan dengan Delete untuk menghapus data berdasarkan id (sebagai cntoh saya gunakan id=2.

Setelah API dengan fungsi CRUD berjalan maka Langkah selanjutnya kita akan buat frontend API dengan react.js

Langkah 1: Membuat Proyek React.js Buka terminal Anda dan jalankan perintah berikut untuk membuat proyek React.js baru:

bash

npx create-react-app react-crud-app

Langkah 2: Menginstal Dependencies Masuk ke direktori proyek yang baru dibuat:

bash

cd react-crud-app

Jalankan perintah berikut untuk menginstal axios, library yang akan kita gunakan untuk mengakses API:

bash

npm install axios

Langkah 3: Membuat Komponen Buka file src/App.js pada folder **react-crud-app** dan gantikan seluruh kode di dalamnya dengan kode berikut:

jsx

import React, { useState, useEffect } from 'react';

import axios from 'axios';

const App = () => {

const [items, setItems] = useState([]);

const [name, setName] = useState('');

const [description, setDescription] = useState('');

useEffect(() => {

fetchItems();

}, []);

const fetchItems = async () => {

try {

const response = await axios.get('http://localhost:8000/api/items');

setItems(response.data);

} catch (error) {

console.error(error);

}

};

const createItem = async (e) => {

e.preventDefault();

try {

const response = await axios.post('http://localhost:8000/api/items', {

name: name,

description: description,

});

setItems([...items, response.data]);

setName('');

setDescription('');

} catch (error) {

console.error(error);

}

};

const updateItem = async (id, newName, newDescription) => {

try {

const response = await axios.put(`http://localhost:8000/api/items/${id}`, {

name: newName,

description: newDescription,

});

setItems((prevItems) =>

prevItems.map((item) =>

item.id === id ? { ...item, name: newName, description: newDescription } : item

)

);

} catch (error) {

console.error(error);

}

};

const deleteItem = async (id) => {

try {

await axios.delete(`http://localhost:8000/api/items/${id}`);

setItems(items.filter((item) => item.id !== id));

} catch (error) {

console.error(error);

}

};

return (

<div>

<h1>CRUD App</h1>

<form onSubmit={createItem}>

<input

type="text"

placeholder="Name"

value={name}

onChange={(e) => setName(e.target.value)}

/>

<input

type="text"

placeholder="Description"

value={description}

onChange={(e) => setDescription(e.target.value)}

/>

<button type="submit">Add Item</button>

</form>

<ul>

{items.map((item) => (

<li key={item.id}>

{item.name} - {item.description}

<button onClick={() => deleteItem(item.id)}>Delete</button>

<button

onClick={() => {

const newName = prompt('Enter new name:', item.name);

const newDescription = prompt('Enter new description:', item.description);

if (newName && newDescription) {

updateItem(item.id, newName, newDescription);

}

}}

>

Edit

</button>

</li>

))} </ul>

</div>

);

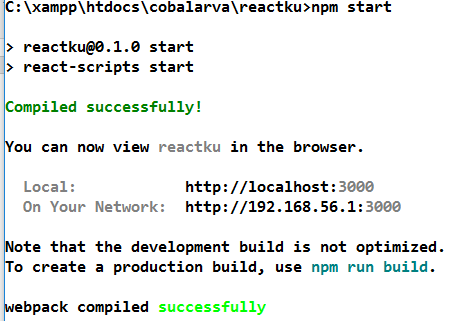
};

export default App;

Langkah 4: Menjalankan Aplikasi Jalankan aplikasi React.js dengan menjalankan perintah berikut di terminal:

Bash (masuk ke direktori react anda dan ketik perintah:

npm start



Aplikasi akan dijalankan di http://localhost:3000 dan Anda akan melihat formulir untuk menambahkan item baru serta daftar item yang sudah ada. Anda dapat menambahkan item baru, menghapus item, dan melihat pembaruan secara real-time di antarmuka React.js.

Pastikan juga bahwa Laravel server masih berjalan di http://localhost:8000 untuk memastikan aplikasi React.js dapat berkomunikasi dengan API Laravel dengan benar.

Jika berhasil akan tampil sebagai berikut:

