

Nim: 312110291.

Nama: Nanda Rosma Anwar.

Kelas: TI.21.A.2.

Membuat REST Controller

Pada tahap ini, kita akan membuat file REST Controller yang berisi fungsi untuk *menampilkan*, *menambah*, *mengubah* dan *menghapus* data. Masuklah ke direktori **app\Controllers** dan buatlah file baru bernama **Post.php**. Kemudian, salin kode di bawah ini ke dalam file tersebut:

```
<?php

namespace App\Controllers;

use CodeIgniter\RESTful\ResourceController;
use CodeIgniter\API\ResponseTrait;
use App\Models\ArtikelModel;

class Post extends ResourceController
{
    use ResponseTrait;

    // all users
    public function index()
    {
        $model = new ArtikelModel();
        $data['artikel'] = $model->orderBy('id', 'DESC')->findAll();
        return $this->respond($data);
    }

    // create
    public function create()
    {
        $model = new ArtikelModel();
        $data = [
            'judul' => $this->request->getVar('judul'),
            'isi' => $this->request->getVar('isi'),
```

```

$model = new ArtikelModel();
$data = $model->where('id', $id)->first();
if ($data) {
    return $this->respond($data);
} else {
    return $this->failNotFound('Data tidak ditemukan.');
```

```

}
// update
```

```

public function update($id = null)
{
    $model = new ArtikelModel();
    $id = $this->request->getVar('id');
    $data = [
        'judul' => $this->request->getVar('judul'),
        'isi'   => $this->request->getVar('isi'),
    ];
    $model->update($id, $data);
    $response = [
        'status'   => 200,
        'error'    => null,
        'messages' => [
            'success' => 'Data artikel berhasil diubah.'
        ]
    ];
    return $this->respond($response);
}
```

```

// delete
```

```

public function delete($id = null)
{
```

Kode diatas berisi 5 method, yaitu:

- **index()** – Berfungsi untuk menampilkan seluruh data pada database.
- **create()** – Berfungsi untuk menambahkan data baru ke database.
- **show()** – Berfungsi untuk menampilkan suatu data spesifik dari database.
- **update()** – Berfungsi untuk mengubah suatu data pada database.
- **delete()** – Berfungsi untuk menghapus data dari database.

Membuat Routing REST API

Untuk mengakses REST API CodeIgniter, kita perlu mendefinisikan route-nya terlebih dulu. Caranya, masuklah ke direktori **app/Config** dan bukalah file **Routes.php**. Tambahkan kode di bawah ini:

```
$routes->resource('post');
```

Untuk mengecek route nya jalankan perintah berikut:

```
php spark routes
```

Selanjutnya akan muncul daftar route yang telah dibuat.

CodeIgniter v4.3.4 Command Line Tool - Server Time: 2023-06-18 06:52:25 UTC+00:00

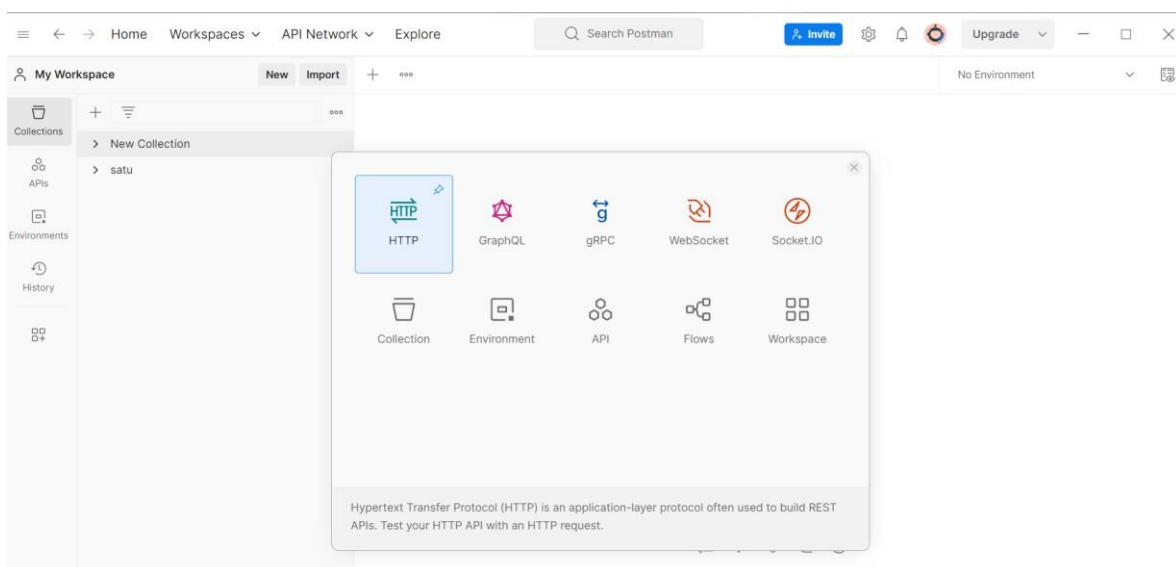
Method	Route	Name	Handler	Before Filters	After Filters
GET	/	»	\App\Controllers\Home::index		toolbar
GET	post	»	\App\Controllers\Post::index		toolbar
GET	post/new	»	\App\Controllers\Post::new		toolbar
GET	post/(.*)/edit	»	\App\Controllers\Post::edit/\$1		toolbar
GET	post/(.*)	»	\App\Controllers\Post::show/\$1		toolbar
POST	post	»	\App\Controllers\Post::create		toolbar
PATCH	post/(.*)	»	\App\Controllers\Post::update/\$1		toolbar
PUT	post/(.*)	»	\App\Controllers\Post::update/\$1		toolbar
DELETE	post/(.*)	»	\App\Controllers\Post::delete/\$1		toolbar

Seperti yang terlihat, satu baris kode routes yang di tambahkan akan menghasilkan banyak Endpoint.

Selanjutnya melakukan uji coba terhadap REST API CodeIgniter.

Testing REST API CodeIgniter

Buka aplikasi postman dan pilih create new → HTTP Request

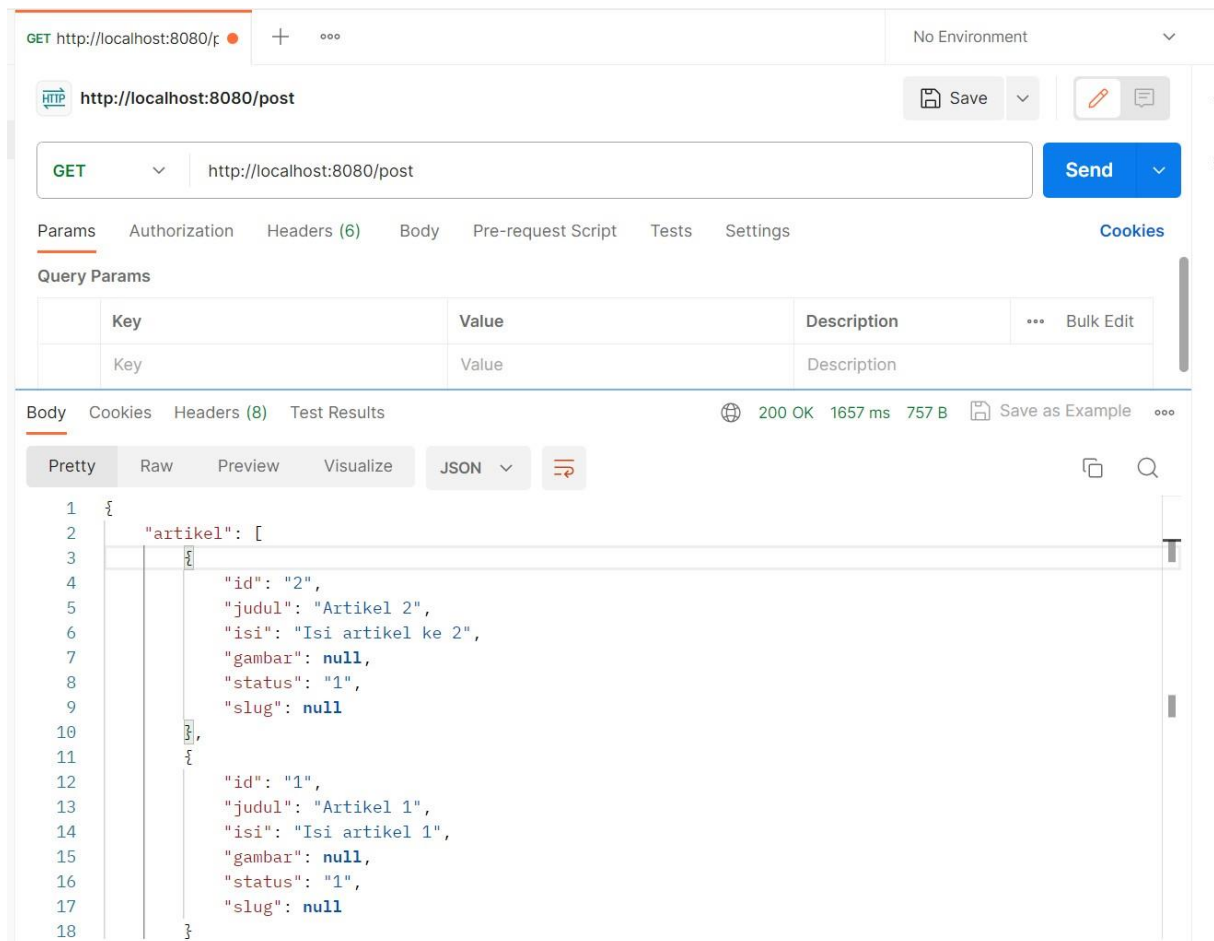


Menampilkan Semua Data

Pilih method **GET** dan masukkan URL berikut:

<http://localhost:8080/post>

Lalu, klik **Send**. Jika hasil test menampilkan semua data artikel dari database, maka pengujian berhasil.

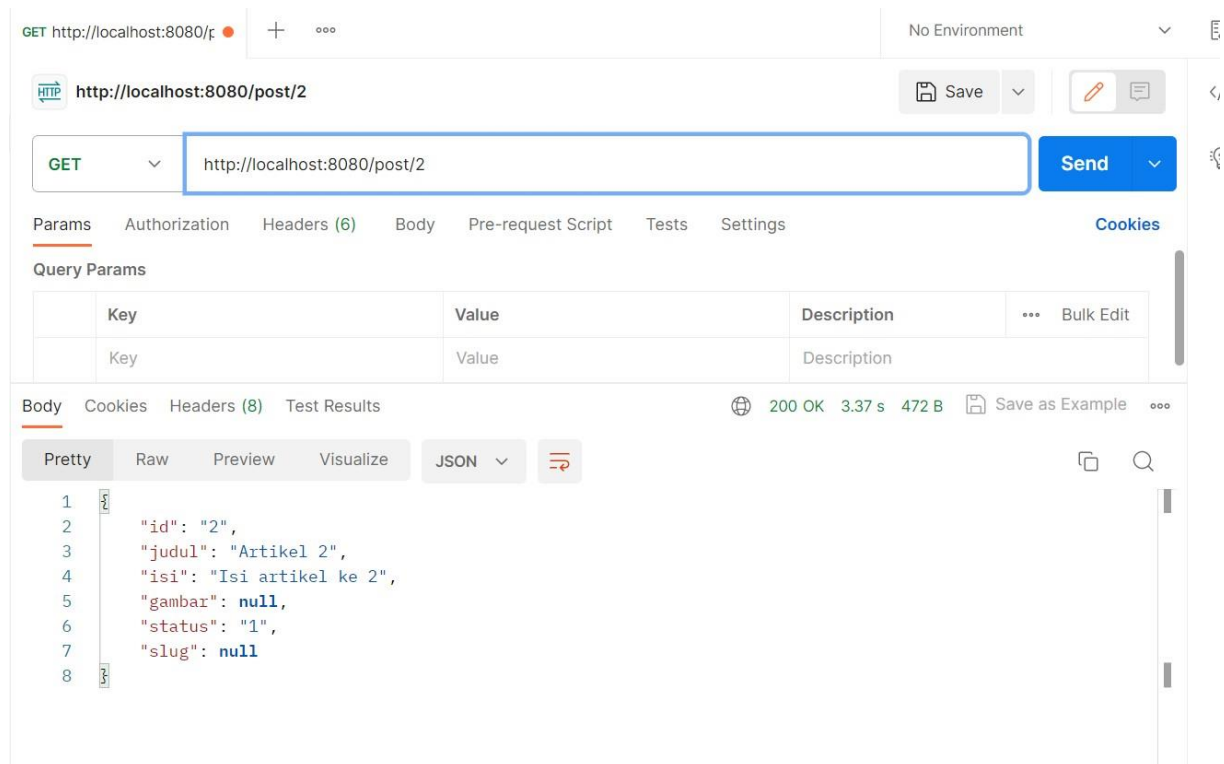


Menampilkan Data Spesifik

Masih menggunakan method **GET**, hanya perlu menambahkan ID artikel di belakang URL seperti ini:

<http://localhost:8080/post/2>

Selanjutnya, klik **Send**. Request tersebut akan menampilkan data artikel yang memiliki ID nomor **2** di database.



Mengubah Data

Untuk mengubah data, silakan ganti method menjadi **PUT**. Kemudian, masukkan URL artikel yang ingin diubah. Misalnya, ingin mengubah data artikel dengan ID nomor 2, maka masukkan URL berikut:

<http://localhost:8080/post/2>

Selanjutnya, pilih tab **Body**. Kemudian, pilih **x-www-form-urlencoded**. Masukkan nama atribut tabel pada kolom **KEY** dan nilai data yang baru pada kolom **VALUE**. Kalau sudah, klik **Send**.

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/post/2`
- Method:** `PUT`
- Body Type:** `x-www-form-urlencoded`
- Body Fields:**

Key	Value	Description
<input checked="" type="checkbox"/> judul	Judul Artikel 2 diubah	
<input checked="" type="checkbox"/> isi	Isi artikel ke 2 diubah melalui API	
- Response:**

```
1 {
2   "status": 200,
3   "error": null,
4   "messages": {
5     "success": "Data artikel berhasil diubah."
6   }
7 }
```

Menambahkan Data

Anda perlu menggunakan method **POST** untuk menambahkan data baru ke database. Kemudian, masukkan URL berikut:

<http://localhost:8080/post>

Pilih tab **Body**, lalu pilih **x-www-form-urlencoded**. Masukkan atribut tabel pada kolom **KEY** dan nilai data baru di kolom **VALUE**. Jangan lupa, klik **Send**.

HTTP <http://localhost:8080/post> Save Send

POST <http://localhost:8080/post> Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	judul	Judul Artikel 4			
<input checked="" type="checkbox"/>	isi	Isi artikel ke 4 ditambah melalui API			
	Key	Value	Description		

Body Cookies Headers (8) Test Results 201 Created 1884 ms 467 B Save as Example

Pretty Raw Preview Visualize JSON Send

```

1  {
2    "status": 201,
3    "error": null,
4    "messages": {
5      "success": "Data artikel berhasil ditambahkan."
6    }
7  }

```

Menghapus Data

Pilih method **DELETE** untuk menghapus data. Lalu, masukkan URL spesifik data mana yang ingin di hapus. Misalnya, ingin menghapus data nomor 4, maka URL-nya seperti ini:

<http://localhost:8080/post/4>

Langsung saja klik **Send**, maka akan mendapatkan pesan bahwa data telah berhasil dihapus dari database.

HTTP <http://localhost:8080/post/4> Save Send

DELETE <http://localhost:8080/post/4> Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (8) Test Results 200 OK 2.96 s 458 B Save as Example

Pretty Raw Preview Visualize JSON Send

```

1  {
2    "status": 200,
3    "error": null,
4    "messages": {
5      "success": "Data artikel berhasil dihapus."
6    }
7  }

```

