|  |  |
| --- | --- |
| Materi | Nilai |
| Lumen 11-20 #7 | 95 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Membuat fungsi update dari controller.menghubungkan method update ke model kategori lalu mengambil id nya dan diberi perintah untuk update $request->all(), artinya apa yang dikirim oleh post man akan menjadi data yang dapat di kirim ke database sesuai perintah.
2. Untuk menggunakan perintah update, menggunakan tipe PUT saat di pasang di routes. lalu di panggil menggunakan post man tipe put, pilig tombol body lalu memilih tipe x-www-from-urlencoded dan masukan key sesuai kolom lalu value yang mau di ubah.
3. Menambahkan fungsi validasi di controller dapam function create. Hal ini digunakan untuk memvalidasi request apapun berdasarkan kolom yang kita pilih.
4. Lalu untuk menambahkan kolom yang sesuai dan apa yang harus di validasi menggunakan codingan, 'kategori' => 'required | uniqe:kategoris'. Artinya petik awal unutk kolomnya, lalu setelah => dan petik ke 2 adalah yang harus di validasi, dapat berupa required dan uniqe:kategoris, artinya uniqe berdasarkan tabel kategoris. Lalu simbol | untuk membuat perintah validasi lebih dari 1.
5. Membuat model,controllers dan migrate Menu. Lalu menambahkan kolom idmenu,idkategori,menu,gambar,harga dalam tabel menu.
6. Membuat fungsi upload gambar dari method create. Caranya yaitu membuat params request lalu params tsb bertipe file yang ada di kolom gembar, lalu getClientOriginalName() digunakan untuk mengambil nama dari file gambar.lalu jika sukses akan menampilkan respon json $gambar
7. Untuk memanggil menu dari routes digunakan codingan $router->post('menu', ['uses' => 'MenuController@create']);. Lalu unutk memanggilnya dari post man menggunakan tipe post, lalu ke body, lalu pilih form data yang kolom gambarnya bertipe file.
8. Lalu agar file dari gambar tersebut yang sudah kita panggil dapat di pindah di public/upload menggunakan perintah move(‘upload’,$gambar).
9. Membuat perintah create data tabel menu. Digunakan unuk mengisi tabel menu sesuai kolom nya melalui post man.
10. Pertama tama membuat validasi dulu agar kolom tersebut harus di isi. Lalu menaruh fungsi untuk upload image. Lalu membuat $data berupa array asosiaif yang isinya adalah key unutk kolomnya lalu value untuk request yang dilakukan dari post man. Lalu hubungkan $data tadi agar dapat terbaca oleh model menu nya. Lalu meresponse $data
11. Unutk menampilkan nya yaitu diperlukan routes get dulu unuk memanggil menuController, lalu pilih body dan masukan key sesuai nama kolom dan tipe nya, dan juga valunya sesuai perintah validasi. Jika sudah di send maka apa yang ada masukan ke value akan dikirim ke database sesuai kolomnya.
12. Menambahkan if dan else unutuk validasi status dan pesan dari $menu. Jika $menu suskses maka akan menampilkan $result yang isinya array asosiatif status => 200,pesan=>”data berhasil di tambahkan” dan data => $data . Jika salah akan menampilkan $result status=>400,pesan=>”data gagal ditambahkan”. Lalu memanggilnya melaluii response json dari $result.
13. Mempelajari cara kerja lumen.

* Request dilakukan apabila kita mengetik di dalam url. Di dalam lumen request akan ditangani oleh route yang sudah kita deklarasikan di dalam project lumen.
* Route pendeklarasian url.
* Middleware letaknya berada pada antara route dan controller. Jika kita tidak memasang middleware maka dari routes akan langsung diarahan ke controller.jika ada maka middleware akan melakukan pemfilteran dahulu lalu mengarahkan controller index.
* Controller akan memberikan respon lalu akan dikembalikan lagi pada request.
* Response adalah menampilkan apa yang telah di request dari url.

1. Middleware akan menangani request dulu, lalu membuat kondisi jika user dari request tersebut adalah bukan admin, maka user akan di lempar ke controller login.Tetapi jika user admin,maka middleware akan meneruskan prosesnya ke dalam controller index.
2. Unutk memasang middlewarenya agar bisa digunakan yaitu registrasikan dahulu di Bootstrap/app.php caranya 'user' => App\Http\Middleware\UserMiddleware::class. Dan memasangnya di routes group $router->group(['prefix' => 'api', 'middleware' => 'user'], function () use ($router) {}. Dan jangan lupa membuat routes login di luar group yang mengarahkan ke LoginController.
3. Middleware digunakan untuk memfilter user mana yang bisa masuk ke controller index dan controller lainnya. Jika user nya sesuai dengan middleware maka akan bisa masuk ke controller yang lain, jika user tidak sesuai middleware, maka user akan diarahkan ke controller login.
4. Agar user dapat masuk, kita harus menambahkan ?user=admin pada url setelah kita mengisi routes yang kita tuju. Itu artinya middleware menganggap user sebagai admin.
5. Menggunakan token sebagai pengaman. Karena token adalah bagian dari middleware. Token adalah perangkat security untuk app yang kita buat. Kita register kan dulu di app.php

$app->register(App\Providers\AuthServiceProvider::class);

Unutk setting agar middlewarenya token/auth yaitu dengan cara $router->group(['prefix' => 'api', 'middleware' => 'auth'], function () use ($router) {}.

1. token digunakan semacam password yang di dapat user dari login. agar request dapat di akses kita harus memasukan token yang sesuai. Jadi jika user ingin mengakses semua request yang di akses di dalam route group, user harus memasukan auth yang sesuai.
2. Cara membuat route yaitu, buka file authserviceprovider lalu masukan codingan berikut.

$this->app['auth']->viaRequest('api', function ($request) {

$token = "1234567890";

$header = $request->header('Api-Token');

if ($header && $header == $token) {

return new User();

}

return null;

Artinya yaitu auth akan menjalankan $token =1234567890; (sebagai psw nya) lalu $header akan menjalankan request dari post man yang nama header nya Api-Token. Lalu membuat pengkondisian jika $header dan $header == $token true, maka akan menjakankan User baru. Jika tidak, berarti tidak muncul apa2 atau null.

1. Membuat migrate user dan kolom kolomnya yang isinya email,password,level,api\_token,status,relasi. Lalu mengisi fileable sesuai kolom database pada model user.
2. Membuat fungsi register. Pertama buat $data yang isinya array asosiatif dan masukan kolom untuk key dan isi kolom unutk value di controller register . lalu tambahkan user::create($data), artinya unutk membuatkan isi dari kolom database user dari $data.
3. Cara ke 2 yaitu menambahkan isi kolom dari postman.pertama pasang dulu fungsi request agar dapat dijalankan dari postmannya yang use Illuminate\Http\Request; .lalu pasang request pada params register..Ubah value dari email,password,dan relasi menjadi $request->input(‘sesuai kolom’), itu digunakan agar dapat menginputkan kolom email,password, dan relasi dari post man.
4. Unutk pemanggilannya menggunakan return response()->json($data); agar dapat dilihat dari post man kolom2 yang berhasil terisi. Sebelum itu ubah dulu tipe routes pada register menjadi post, begitu juga saat mau mengisi kolom dari post man.
5. Membuat fungsi login. buat $email dan $password agar dapat di isikan dari postman.lalu buat $user yang dalamnya berisi tabel user dari database dan mencari email yang sesuai dari $email dan databse, lalu ->first() digunakan unutk mengambil baris dari tabel dari email yang cocok . lalu membuat kondisi jika password dalam $user cocok dengan password yang user ketik di postman saat login, maka akan dibuatkan token baru dengan cara $user->update() yang di isi array asosiatif kolom api\_token yang di isi dengan $token melalui $token =Str:random(20);. Jadi setiap user login, maka token akan selalu ter update.
6. Jika login berhasil maka akan menampilkan response pada post man melalui response json yang isinya array asosiatif kolom pesan berhasil, token yang di update dan datanya dari $user. Jika gagal akan keluar response yang sama dan array asosiatif nya isinya pesan gagal,dan datanya kosong.
7. Unutk pemanggilannya pertama ubah routes login menjadi post. Lalu isi kolom email dan password sesuai database pada id yang sesuai.
8. Menubah input menjadi header yang ada dalam auth. Lalu masuk ke api/kategori dan menambahkan token dari user yang sudah di update di dalam header yang ada di kolom api\_token.

**Saya Belum Mengerti**