Dokumentasi Laravel

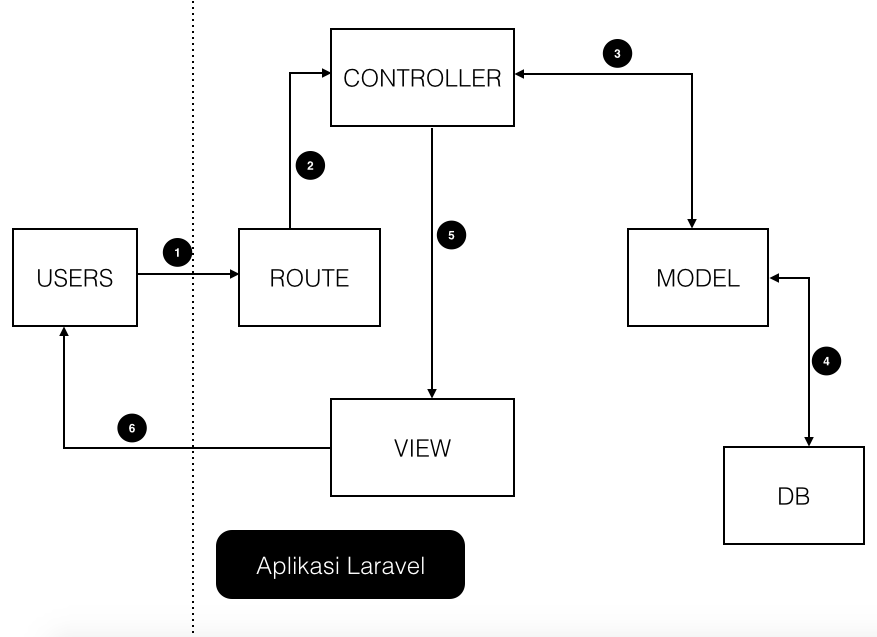
Pada laravel saya belum membuat management web service(manajemen backend) jadi laravel hanya saya gunakan sebagai web service api saja. Dan untuk gambar-gambar saya masih mengambilnya secara online jadi pastikan connect ke internet. Dan untuk gambar pada navigation disamping nama dan pada profile ada gambar untuk user yang sudah ada pada direktori storage/app/public/avatar/ yang sudah saya link kan ke public/storage. Apabila tidak muncul berarti

Domain-domani api dapat dilihat pada direktori Routes/api.php disitu saya menggunakan routing grup prefix v1 jadi apabila ingin mengakses route api harus diawali dengan v1 contoh tugasakhir.test/v1/categories. Projek ini akan saya terus kembangkan hingga menjadi portofolio yang sempurna. Masih banyak bug dan kekurangan dari projek ini

struktur laravel:

* app -> folder ini berisi logic dari aplikasi kita seperti controller, middleware dan model.
* bootstrap -> folder ini berisi file-file yang diakses pertama kali setelah index dan config.
* config -> folder ini berisi konfigurasi aplikasi.
* database -> folder ini berisi file migration, dan seeding database.
* public -> folder ini berisi file-file yang boleh diakses oleh user melalui web browser.
* resources -> folder ini berisi asset yang digunakan pada aplikasi seperti css, js, image, dan view (blade template).
* routes -> folder ini berisi file-file yang digunakan untuk mengatur routing aplikasi.
* storage -> folder ini digunakan sebagai tempat penyimpanan cache, session, dan log aplikasi. Pastikan folder ini dapat ditulisi oleh PHP.
* test -> folder ini berisi kode-kode testing
* vendor -> folder ini berisi pustaka yang digunakan oleh aplikasi.

MVC pada laravel :



user mengakses aplikasi melalui route tertentu, route tersebut oleh aplikasi dipetakan ke controller action, controller action akan menggunakan model untuk mengakses data, atau mengembalikan view tanpa data, model berinteraksi dengan database untuk emndapatkan data, setelah berhasil mendapatkan data melalui model, controller akan mengembalikan sebuah view sekaligus data jika ada, view tersebut pada akhirnya yang dilihat oleh user.

Routing pada laravel :

* api.php -> konfigurasi routing khusus untuk web service
* web.php -> konfigurasi routing untuk aplikasi web biasa
* channels.php -> konfigurasi routing untuk broadcast message seperti notification dan chat.
* console.php -> konfigurasi routing untuk diakses pada console

pada tugas ini saya menggunakan routing api(web service), web service sendiri merupakan standart yang digunakan untuk pertukaran data antar aplikasi atau sistem berbasis web, dan disini saya menggunakan standart Restful api web service , routing api ini bersifat stateless, Tujuan utama dari stateless adalah memudahkan peningkatan (scale-up) concurrent access terhadap web service,

http methods pada laravel diantaranya adalah GET, POST, PUT, PATCH, DELETE, OPTIONS, dan agar routing pada laravel bisa diakses oleh beberapa methods sekaligus dapat menggunakan MATCH dan agar bisa diakses oleh semua method bisa menggunakan ANY. Untuk penggunaan routing bisa dilihat pada folder routes, lalu api.php karena disini saya menggunakan routing api. Seharusnya untuk menggunakan routing api ini harus menggunakan prefix api contoh : <http://tugasakhir.test/api/v1/menu/> , untuk menghapus prefix ini kita bisa masuk ke folder App\Providers\RouteServiceProvider, tepatnya pada fungsi mapApiRoutes().

Controller pada laravel

adalah action yang menggunakan model untuk mengakses data, atau langsung mengembalikan View tanpa data, controller mempunyai hubungan erat dengan routing , dimana kita bisa meletakkan logic atau pertukaran proses pada controller sehigga definisi routing tidak tercampur dengan logic. Controller disimpan pada direktori app/Http/Controller.

Untuk membuat sebuah controller gunakan perintah : php artisan make:controller NamaController, nama controller biasanya diikuti dengan kata controller pada akhir nama.

View Pada laravel

bertangung jawab untuk memberikan tampilan ke user, view digunakan untuk meletakkan kode html, css, dan javascript. Laravel menggunakan tamplating engine bawaan Blade maka file view diakhiri dengan .blade.php. View terletak pada folder resources/views/. Untuk membuat view tidak diperlukan perintah khusu sebagaimana controller kita bisa membuatnya sendiri secara manual.

Model pada laravel

model pada laravel tidak mempunyai folder khusus layaknya view dan controller, model pada laravel biasanya diletakkan pada folder App/ dikarenakan model pada laravel memiliki 2 definisi yaitu model sebagai entitas yang berinteraksi dengan sumber data dan model sebagai representasi model di proses pertukaran. Disini saya menggunakan model keduanya yaitu model yang berinteraksi dengan sumber data dan model diproses pertukaran dapat dilihat pada model User.php saya menggunakan function generateToken yang digunakan untuk pertukaran proses yang digunakan untuk menggenerate token antara login dan logout, cek pada AuthController. Model bertugas untuk berinteraksi dengan sumber data seperti query database, insert data , update, atau hapus record pada database. Ketika membuat model dilaravel sebetulnya kita sudah membuat elequent model. Model pada laravel merupakan turunan dari ORM elequent model yang merupakan ORM atau objek relational models yang digunakan oleh laravel untuk mengimplementasikan konsep activeRecord . Coba cek pada salah satu model ketika sesudah dibuat pada file tersebut akan akan code use Illuminate\Database\Elequent\Model;, dengan ini model yang kita buat memiliki method2 yang ada pada elequent yang bisa kita gunakan .

Middleware pada laravel

middleware merupakan mekanisme dilaravel yang memungkinkan kita melakukan filtering atau pengecekan terhadap HTTP request yang masuk ke aplikasi kita. Middleware diletakkan pada direktori app/Http/Middleware. Middleware bawaan laravel diantaranya adalah otentikasi dan proteksi CSRF, kita bisa juga menambahkan sendiri. Pada tugas ini saya membuat sebuah middleware yaitu untuk CORS(Cross-Origin Resource Sharing). Cors dimana javascript tidak diizinkan melakukan HTTP request ke domain server berbeda. Misalnya web cafeshop.id saya mengakses web service laravel saya [http://tugasakhir.test](http://tugasakhir.test/), untuk mengatasi ini saya membuat middleware Cors, dapat dilihat pada folder app/Http/Middleware/Cors.php.

Database pada laravel.

Konfigurasi database laravel terletak pada file .env dalam folder web laravel kita.

Pada laravel ada migration merupakan version control untuk database yang mengizinkan kita dengan mudah memodifikasi atau mengshare skema database pada projek aplikasi kita, sehingga perubahan dari skema database dapat terlacak. Dapat dilihat pada folder database/migration. Untuk membuat sebuah migration kita dapat menggunakan perintah php artisan make:migration namamigrationtable. Setelah jadi file migtrationnya makan akan ada waktu pembuatannya pada nama file migration. Setelah konfigurasi migration selesai dapat menjalankan perintah php artisan migrate untuk menggenerate file migration. Pada file migration akan ada function up dan function down, fungsi up digunakan untuk menambahkan atau mengedit sebuah table pada database. Dan fungsi down sebaliknya. Pada laravel juga ada yang dinamakan seeding yang fokus pada data yang tersimpan pada tabelnya. Umunnya seedung digunakan untuk menginject data dummy untuk keperluan testing aplikasi. Seeder disimpan pada direktori database/seeds. Untuk membuat seeder jalankan perintah php artisan make:seeder namaSeeder. Untuk menggenerate file seeder jalankan perintah php artisan <db:seed> –class=namaSeeder. Atau apabila banyak seeder kita dapat mendaftarkan seeder pada DatabaseSeeder.php pada direktori seeder lalu tambahkan $this→call(namaSeeder::class) didalam function run. Lalu jalankan perintah php artisan <db:seed>.

API Resource pada laravel

ketika kita membangun sebuah web service kita perlu mengubah data yang berasal dari elequent berformat JSON sebagai response kepada client. Resource merupakan sebuah class yang dapat kita generate scaffoldingnya menggunakan perintah php artisan make:resource NamaResource. File resource terletak pada direktori app/Http/Resources.