

Application Programming Interface

Pertemuan 11

Gelar Aditya Pratama, S.T., M.Kom.
gelar@stmik-amikbandung.ac.id



laravel api & mysql – create model & migration

Folder project



Dalam folder praktek, klik kanan, pilih opsi “Open in Terminal”

Jalankan command “composer create-project laravel/laravel:^9.0 api-laravel”

```
Windows PowerShell
PS D:\Dokumen\Dosen\STMIK AMIK Bandung\2022-2023 Genap\Application Programming Interface\praktek> composer create-project laravel/laravel:^9.0 api-laravel
Creating a "laravel/laravel:9.0" project at "./api-laravel"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v9.0.0)
- Installing laravel/laravel (v9.0.0): Extracting archive
Created project in D:\Dokumen\Dosen\STMIK AMIK Bandung\2022-2023 Genap\Application Programming Interface\praktek\api-laravel
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 112 installs, 0 updates, 0 removals
- Locking asm89/stack-cors (v2.1.1)
- Locking brick/math (0.11.0)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/deprecations (v1.0.0)
- Locking doctrine/inflector (2.0.6)
- Locking doctrine/instantiator (1.5.0)
- Locking doctrine/lexer (2.1.0)
- Locking dragonmantank/cron-expression (v3.3.2)
- Locking egulias/email-validator (3.2.5)
- Locking fakerphp/faker (v1.22.0)
- Locking filp/whoops (2.15.2)
```

Folder project



Buka folder “api-laravel”, buka file .env
Kemudian edit bagian DB

```
# .env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:RoRnQTM2nOUtqYUY6I090VEBSiqIfynCdrJs9m2xcYo=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=api-laravel
15 DB_USERNAME=root
16 DB_PASSWORD=
```

Folder project



Kembali ke terminal,

Jalankan command "cd api-laravel"

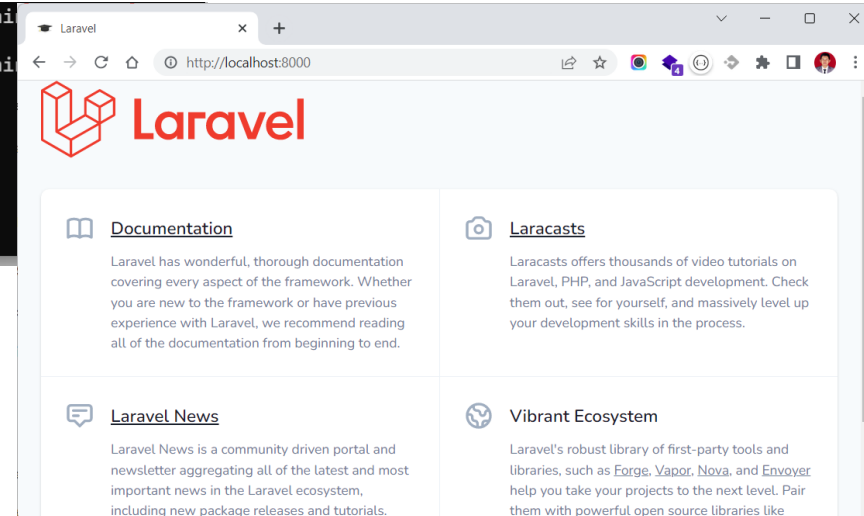
Jalankan command "php artisan serve"

Buka browser, ketikkan url "localhost:8000"

```
PS D:\Dokumen\Dosen\STMIK AMIK Bandung\2022-2023 Genap\Application Programmi
praktek> cd api-laravel
PS D:\Dokumen\Dosen\STMIK AMIK Bandung\2022-2023 Genap\Application Programmi
praktek\api-laravel> php artisan serve
```

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server



Global function



Buka file

/app/Providers/AppServiceProvider.php

Tambahkan import Schema

Tambahkan script di bagian function boot()

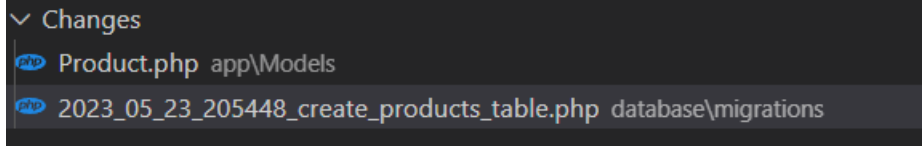
```
AppServiceProvider.php M X
app > Providers > AppServiceProvider.php > AppServiceProvider > boot
1  <?php
2
3  namespace App\Providers;
4
5  use Illuminate\Support\ServiceProvider;
6  use Illuminate\Support\Facades\Schema;
7
8  class AppServiceProvider extends ServiceProvider
9  {
10     /** ...
11     *
12     * public function register()
13     *
14     * { ...
15     *
16     * }
17     *
18     /** ...
19     *
20     * public function boot()
21     *
22     * { ...
23     *
24     *     Schema::defaultStringLength(191);
25     *
26     * }
27 }
28
29
30
```

Create Model



Buka terminal baru, klik kanan folder api-laravel, pilih opsi “Open in Terminal”
Jalankan command “php artisan make:model Product -m”

Akan terbentuk 2 file baru



Create Model



Edit file /database/migration/
_create_products_table.php

Tambahkan field-field berikut

- Name
- Image
- Description
- Type
- Price

```
AppServiceProvider.php M 2023_05_23_205448_create_products_table.php U X
database > migrations > 2023_05_23_205448_create_products_table.php > #Class#94008774 > up
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9  > ... /** ...
14 ... public function up()
15 ... {
16 ... Schema::create('products', function (Blueprint $table) {
17 ...     $table->id();
18 ...     $table->string('name');
19 ...     $table->string('image');
20 ...     $table->text('description');
21 ...     $table->string('type');
22 ...     $table->double('price');
23 ...     $table->timestamps();
24 ... });
25 ... }
26
27 > ... /** ...
32 ... public function down()
33 ... {
34 ...     Schema::dropIfExists('products');
35 ... }
36 ... };
```


Create Model



Edit file /app/Models/Products.php
Tambahkan \$fillable

- Name
- Image
- Description
- Type
- Price

```
AppServiceProvider.php M 2023_05_23_205448_create_products_table.php U Product.php U X
app > Models > Product.php > Product > $fillable
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Product extends Model
9  {
10     use HasFactory;
11
12     /**
13      * The attributes that are mass assignable.
14      *
15      * @var array<int, string>
16      */
17     protected $fillable = [
18         'name',
19         'image',
20         'description',
21         'type',
22         'price',
23     ];
24 }
25
```

Create Resource

Buka kembali terminal, jalankan command
“php artisan make:resource ProductResource”

Edit file /app/Http/Resources/ProductResource.php

```
ProductResource.php U X
app > Http > Resources > ProductResource.php > ProductResource
1  <?php
2
3  namespace App\Http\Resources;
4
5  use Illuminate\Http\Resources\Json\JsonResource;
6
7  class ProductResource extends JsonResource
8  {
9
10     ... public $status;
11     ... public $message;
12     ... public $pagination;
13
14     > ... /** ...
22     ... public function __construct($status, $message, $resource, $pagination)
23     ... {
24         ... parent::__construct($resource);
25         ... $this->status = $status;
26         ... $this->message = $message;
27         ... $this->pagination = $pagination;
28     ... }
29
30     > ... /** ...
36     ... public function toArray($request)
37     ... {
38         ... return [
39             ... 'success' => $this->status,
40             ... 'message' => $this->message,
41             ... 'data' => $this->resource,
42             ... 'pagination' => $this->pagination,
43         ... ];
44     ... }
45 }
46
```

Create Controller

Buka kembali terminal, jalankan command
“php artisan make:controller
Api/ProductController”

Edit file
/app/Http/Controllers/ProductController.php

ProductController.php U X

app > Http > Controllers > Api > ProductController.php > ProductController > index

```
1  <?php
2
3  namespace App\Http\Controllers\Api;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7  use App\Models\Product;
8  use App\Http\Resources\ProductResource;
9
10 class ProductController extends Controller
11 {
12     /**
13      *
14      * @param Request $request
15      * @return ProductResource
16      */
17     public function index(Request $request)
18     {
19         $query = Product::get()->toArray();
20
21         $datas = [];
22         foreach($query as $qry){
23             $temp = $qry;
24
25             $created_at_indo = \Carbon\Carbon::parse($temp['created_at']);
26             $created_at_indo->locale('id')->settings(['formatFunction'=> 'translatedFormat']);
27
28             $temp['created_date_indo'] = $created_at_indo->format('l, d-F-Y H:i:s');
29             array_push($datas, $temp);
30         };
31
32         return new ProductResource(true, 'Get Data Successfull', $datas, []);
33     }
34
35     public function show(Request $request){}
36     public function store(Request $request){}
37     public function update(Request $request){}
38     public function destroy(Request $request){}
39 }
40
```

Create Route



Edit file /routes/api.php

```
api.php M X
routes > api.php > ...
1  <?php
2
3  use Illuminate\Http\Request;
4  use Illuminate\Support\Facades\Route;
5
6  /** ...
18 Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
19     ...return $request->user();
20 });
21
22
23
24 Route::apiResource('/product', App\Http\Controllers\Api\ProductController::class);
25
```

Postman Response



Buka postman

Gunakan method GET

Ketikan url "localhost:8000/api/product"

A screenshot of the Postman application interface. The top bar shows the method "GET" and the URL "localhost:8000/api/product". Below the bar, the "Body" tab is selected, and the response is displayed in "Pretty" format. The JSON response is as follows:

```
{
  "success": true,
  "message": "Get Data Successfull",
  "data": [
    {
      "id": 1,
      "name": "kertas a4",
      "image": "",
      "description": "1 rim",
      "type": "alat kantor",
      "price": 40000,
      "created_at": "2023-05-24T04:28:36.000000Z",
      "updated_at": "2023-05-24T04:28:38.000000Z",
      "created_date_indo": "Rabu, 24 Mei 2023 04:28:36"
    },
    {
      "id": 2,
      "name": "pensil 2b",
      "image": "",
      "description": "1 buah",
      "type": "alat kantor",
      "price": 2000,
      "created_at": "2023-05-24T04:49:06.000000Z",
      "updated_at": "2023-05-24T04:49:08.000000Z",
      "created_date_indo": "Rabu, 24 Mei 2023 04:49:06"
    }
  ],
  "pagination": []
}
```