

Rafif Tri Hartanto

SIB 2A

26

Jobsheet 10 PWL

Praktikum 1: Membuat RESTful API Register

1.

The Postman app

Download the app to get started with the Postman API Platform.

 Windows 64-bit

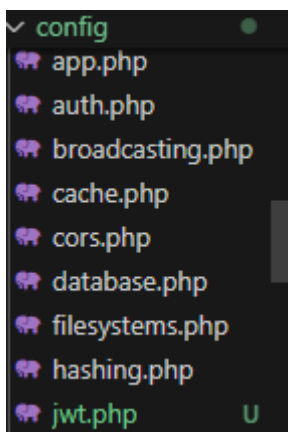
2.

```
PS C:\laragon\www\PWL_25\PWL_POS> composer require tymon/jwt-auth:2.1.1
./composer.json has been updated
Running composer update tymon/jwt-auth
Loading composer repositories with package information
Updating dependencies
Lock file operations: 4 installs, 0 updates, 0 removals
- Locking lcobucci/clock (2.3.0)
```

3.

```
PS C:\laragon\www\PWL_25\PWL_POS> php artisan vendor:publish --provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"
[INFO] Publishing assets.
Copying file [C:\laragon\www\PWL_25\PWL_POS\vendor\tymon\jwt-auth\config\config.php] to [C:\laragon\www\PWL_25\PWL_POS\config\jwt.php] ..... DONE
```

4.



5.

```
PS C:\laragon\www\PWL_25\PWL_POS> php artisan jwt:secret
jwt-auth secret [XtmQtskNAKTV7NYqLwmEH4YthgumQwDd01B2VFTJOeLr1UB38qANeJkky82pEh8g] set successfully.

PWL_POS > .env

54 VITE_APP_NAME="${APP_NAME}"
55 VITE_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
56 VITE_PUSHER_HOST="${PUSHER_HOST}"
57 VITE_PUSHER_PORT="${PUSHER_PORT}"
58 VITE_PUSHER_SCHEME="${PUSHER_SCHEME}"
59 VITE_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
60
61 JWT_SECRET=XtmQtskNAKTV7NYqLwmEH4YthgumQwDd01B2VFTJOeLr1UB38qANeJkky82pEh8g
62
```

6.

```
38     'guards' => [
39         'web' => [
40             'driver' => 'session',
41             'provider' => 'users',
42         ],
43         'api' => [
44             'driver' => 'jwt',
45             'provider' => 'users',
46         ],
47     ],
48 ],
```

7.

```
PWL_POS > app > Models > UserModel.php > PHP Intelephense > UserModel

1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Casts\Attribute;
6  use Illuminate\Database\Eloquent\Model;
7  use Tymon\JWTAuth\Contracts\JWTSubject;
8  use Illuminate\Database\Eloquent\Relations\BelongsTo;
9  use Illuminate\Foundation\Auth\User as Authenticatable;
10
11  class UserModel extends Authenticatable implements JWTSubject
12  {
13      public function getJWTIdentifier() {
14          return $this->getKey();
15      }
16
17      public function getJWTCustomClaims() {
18          return [];
19      }
20
21      protected $table = 'm_user';
22      protected $primaryKey = 'user_id';
```

8.

```
PS C:\laragon\www\PWL_25\PWL_POS> php artisan make:controller Api/RegisterController

INFO Controller [C:\laragon\www\PWL_25\PWL_POS\app\Http\Controllers\Api\RegisterController.php] created successfully.
```

9.

```
PWL_POS > app > Http > Controllers > Api > RegisterController.php > PHP Intelephense > RegisterController.php

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\UserModel;
8  use Illuminate\Support\Facades\Validator;
9
10 class RegisterController extends Controller
11 {
12     public function __invoke(Request $request)
13     {
14         //set validation
15         $validator = Validator::make($request->all(), [
16             'username' => 'required',
17             'nama' => 'required',
18             'password' => 'required|min:5|confirmed',
19             'level_id' => 'required',
20             'image' => 'required|image|mimes:jpeg,png,jpg,gif|max:2048',
21         ]);
22
23         //if validation fails
24         if($validator->fails()){
25             return response()->json($validator->errors(), 422);
26         }
27
28         //create user
29         $user = UserModel::create([
30             'username' => $request->username,
31             'nama' => $request->nama,
32             'password' => bcrypt($request->password),
33             'level id' => $request->level id,
```

10.

```
PWL_POS > routes > api.php
1  <?php
2
3  use App\Http\Controllers\Api\RegisterController;
4  use Illuminate\Http\Request;
5  use Illuminate\Support\Facades\Route;
6
7  /*
8  |-----
9  | API Routes
10 |-----
11 |
12 | Here is where you can register API routes for your application. These
13 | routes are loaded by the RouteServiceProvider and all of them will
14 | be assigned to the "api" middleware group. Make something great!
15 |
16 */
17
18 Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');
19
```

11.

POST localhost/PWL_25/PWL_POS/public/api/register Send

```

1  {
2      "username": [
3          "The username field is required."
4      ],
5      "nama": [
6          "The nama field is required."
7      ],
8      "password": [
9          "The password field is required."
10     ],
11     "level_id": [
12         "The level id field is required."
13     ],
14     "image": [
15         "The image field is required."
16     ]
17 }
```

12.

POST localhost/PWL_25/PWL_POS/public/api/register Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key		Value	Description	*** Bulk Edit
<input checked="" type="checkbox"/>	username	Text	penggunasatu		
<input checked="" type="checkbox"/>	nama	Text	Pengguna1		
<input checked="" type="checkbox"/>	password	Text	12345		
<input checked="" type="checkbox"/>	password_confirmation	Text	12345		
<input checked="" type="checkbox"/>	level_id	Text	2		

```

1  {
2      "success": true,
3      "user": {
4          "username": "penggunasatu",
5          "nama": "Pengguna1",
6          "level_id": "2",
7          "updated_at": "2025-07-10T04:54:12.000000Z",
8          "created_at": "2025-07-10T04:54:12.000000Z",
9          "user_id": 30
10     }
11 }

```

Praktikum 2: Membuat RESTful API Login

1.

```

PS C:\laragon\www\PWL_25\PWL_POS> php artisan make:controller Api/LoginController

[INFO] Controller [C:\laragon\www\PWL_25\PWL_POS\app\Http\Controllers\Api>LoginController.php] created successfully.

```

2.

```

PWL_POS > app > Http > Controllers > Api > LoginController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Api;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Validator;
8
9  class LoginController extends Controller
10 {
11
12     public function __invoke(Request $request){
13         // set validation
14         $validator = Validator::make($request->all(), [
15             'username' => 'required',
16             'password' => 'required'
17         ]);
18
19         // if validation fails
20         if ($validator->fails()) {
21             return response()->json($validator->errors(), 422);
22         }
23
24         // get credentials from request
25         $credentials = $request->only('username', 'password');
26
27         // if auth failed
28         if (!$token = auth()->guard('api')->attempt($credentials)) {
29             return response()->json([
30                 'success' => false,
31                 'message' => 'Username atau Password Anda salah'
32             ], 401);
33         }

```


6.

localhost/PWL_25/PWL_POS/public/api/login?username=penggunasatu&password=12346

POST localhost/PWL_25/PWL_POS/public/api/login?username=penggunasatu&password=12346

Params Authorization Headers (7) Body Scripts Settings Cookies

Key	Value	Description
username	penggunasatu	
password	12346	

Body Cookies Headers (11) Test Results 401 Unauthorized 634 ms 444 B

```

1 {
2   "success": false,
3   "message": "Username atau Password Anda salah"
4 }
  
```

7.

localhost/PWL_25/PWL_POS/public/api/login?username=penggunasatu&password=12345

GET localhost/PWL_25/PWL_POS/public/api/login?username=penggunasatu&password=12345

Params Authorization Headers (6) Body Scripts Settings Cookies

Key	Value	Description
username	penggunasatu	
password	12345	

Body Cookies Headers (10) Test Results 405 Method Not Allowed 678 ms 1020.13 KB

```

1 <!DOCTYPE html>
2 <html lang="en" class="auto">
3 <!--
4 Symfony\Component\HttpKernel\Exception\MethodNotAllowedHttpException: The GET method is not supported for route
  api/login. Supported methods: POST. in file
  C:\laragon\www\PWL_25\PWL_POS\vendor\laravel\framework\src\Illuminate\Routing\AbstractRouteCollection.php on
  line 122
5
6 #0 C:\laragon\www\PWL_25\PWL_POS\vendor\laravel\framework\src\Illuminate\Routing\AbstractRouteCollection.php(107):
  Illuminate\Routing\AbstractRouteCollection->requestMethodNotAllowed(Object(Illuminate\Http\Request),
  Array, &#039;GET&#039;);
7 #1 C:\laragon\www\PWL_25\PWL_POS\vendor\laravel\framework\src\Illuminate\Routing\AbstractRouteCollection.php(41):
  Illuminate\Routing\AbstractRouteCollection->getRouteForMethods(Object(Illuminate\Http\Request), Array)
8 #2 C:\laragon\www\PWL_25\PWL_POS\vendor\laravel\framework\src\Illuminate\Routing\RouteCollection.php(162):
  Illuminate\Routing\AbstractRouteCollection->handleMatchedRoute(Object(Illuminate\Http\Request), NULL)
9 #3 C:\laragon\www\PWL_25\PWL_POS\vendor\laravel\framework\src\Illuminate\Routing\Router.php(761):
  Illuminate\Routing\RouteCollection->match(Object(Illuminate\Http\Request))
  
```

Pesan error 405 Method Not Allowed menunjukkan bahwa endpoint /api/login tidak menerima permintaan dengan metode GET karena rute tersebut hanya diatur untuk menangani permintaan POST.

Praktikum 3: Membuat RESTful API Logout

1.

```
61 JWT_SECRET=XtmQtskNAKTV7NYqLwmEH4YthgumQWdD0lB2VFTJOeLr1UB38qANeJkky82pEh8g
62 JWT_SHOW_BLACKLIST_EXCEPTION=true
63
```

2.

```
PS C:\laragon\www\PWL_25\PWL_POS> php artisan make:controller Api/LogoutController

INFO Controller [C:\laragon\www\PWL_25\PWL_POS\app\Http\Controllers\Api\LogoutController.php] created successfully.
```

3.

```
PWL_POS > app > Http > Controllers > Api > LogoutController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Api;
4  use App\Http\Controllers\Controller;
5  use Illuminate\Http\Request;
6  use Tymon\JWTAuth\Facades\JWTAuth;
7  use Tymon\JWTAuth\Exceptions\JWTAuthException;
8  use Tymon\JWTAuth\Exceptions\TokenExpiredException;
9  use Tymon\JWTAuth\Exceptions\TokenInvalidException;
10
11 class LogoutController extends Controller
12 {
13     public function __invoke(Request $request)
14     {
15         // remove token
16         $removeToken = JWTAuth::invalidate(JWTAuth::getToken());
17
18         if ($removeToken) {
19             // return response JSON
20             return response()->json([
21                 'success' => true,
22                 'message' => 'Logout Berhasil!',
23             ]);
24         }
25     }
26 }
```

4.

```
18 Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');
19 Route::post('/login', App\Http\Controllers\Api>LoginController::class)->name('login');
20 Route::post('/logout', App\Http\Controllers\Api\LogoutController::class)->name('logout');
21 Route::middleware('auth:api')->get('/user', function (Request $request) {
22     return $request->user();
23 });
```


5.

localhost/PWL_25/PWL_POS/public/api/logout Save Share

POST localhost/PWL_25/PWL_POS/public/api/logout Send

6.

"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3MiOiJodHRwOi8vbG9jYXRob3N0L1BXTF8yNS9QV0xfUE9TL3B1YmtpYy9hcGkvbG9naW4iLCJpYXQ10jeE3NTIxMjQ2ODQsImV4cCI6MTc1MjEyODI4NCwibmJmIjoxNzUyMTI0Njg0LCJqdGkiOiJhbnB3MXNtZzJmMERnM2hsIiwic3ViIjoimZAILCJwcnYiOiI0MWRmODgzNGYxYjk4ZjcwZWZhbnJhYVWVkbWY0MjM0MTM3MDA2OTBjIn0.XvsCA-ddNsw2kcoJ0rGoyfLKIKeLuzY10xxYRTeipOo"

localhost/PWL_25/PWL_POS/public/api/logout Save Share

POST localhost/PWL_25/PWL_POS/public/api/logout Send

Params **Authorization** Headers (8) Body Scripts Settings Cookies

Auth Type
Bearer Token

Token
.....

The authorization header will be automatically generated when you

Body Cookies Headers (12) Test Results 🕒

{ } JSON Preview Visualize 🔍 📄 🔗

```
1 {  
2   "success": true,  
3   "message": "Logout Berhasil!"  
4 }
```

200 OK • 412 ms • 437 B • 🌐 • ⋮

Praktikum 4: Implementasi CRUD dalam RESTful API

1.

```
PS C:\laragon\www\PWL_25\PWL_POS> php artisan make:controller Api/LevelController
```

```
INFO Controller [C:\laragon\www\PWL_25\PWL_POS\app\Http\Controllers\Api\LevelController.php] created successfully.
```

2.

```
PWL_POS > app > Http > Controllers > Api > LevelController.php > PHP Intelephense > LevelController

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\LevelModel;
8
9  class LevelController extends Controller
10 {
11     public function index(){
12         return LevelModel::all();
13     }
14
15     public function store(Request $request){
16         $level = LevelModel::create($request->all());
17         return response()->json($level, 201);
18     }
19
20     public function show(LevelModel $level){
21         return LevelModel::find($level);
22     }
23
24     public function update(Request $request, LevelModel $level){
25         $level->update($request->all());
26         return LevelModel::find($level);
27     }
28
29     public function destroy(LevelModel $level){
30         $level->delete();
31         return response()->json([
32             'success' => true,
33             'message' => 'Data terhapus',
```

3.

```
PWL_POS > routes > api.php > ...
1  <?php
2
3  use App\Http\Controllers\Api\RegisterController;
4  use Illuminate\Http\Request;
5  use Illuminate\Support\Facades\Route;
6  use App\Http\Controllers\Api\LevelController;
7
8
9  /*
10 |-----
11 | API Routes
12 |-----
13 |
14 | Here is where you can register API routes for your application. These
15 | routes are loaded by the RouteServiceProvider and all of them will
16 | be assigned to the "api" middleware group. Make something great!
17 |
18 */
19
20 Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');
21 Route::post('/login', App\Http\Controllers\Api>LoginController::class)->name('login');
22 Route::post('/logout', App\Http\Controllers\Api\LogoutController::class)->name('logout');
23 Route::middleware('auth:api')->get('/user', function (Request $request) {
24     return $request->user();
25 });
26
27 //route api level
28 Route::get('/levels', [LevelController::class, 'index']);
29 Route::post('/levels', [LevelController::class, 'store']);
30 Route::get('/levels/{level}', [LevelController::class, 'show']);
31 Route::put('/levels/{level}', [LevelController::class, 'update']);
32 Route::delete('/levels/{level}', [LevelController::class, 'destroy']);
```

4.

localhost/PWL_25/PWL_POS/public/api/levels
Save
Share

GET
localhost/PWL_25/PWL_POS/public/api/levels
Send

Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies

Body
Cookies
Headers (11)
Test Results
Visualize

200 OK
366 ms
781 B

JSON
Preview
Visualize

```

1  [
2    {
3      "level_id": 1,
4      "level_kode": "ADM",
5      "level_nama": "Administrator",
6      "created_at": null,
7      "updated_at": null
8    },
9    {
10     "level_id": 2,
11     "level_kode": "MNG",
12     "level_nama": "Manager",
13     "created_at": null,
14     "updated_at": null
15   },
16   {
17     "level_id": 3,
18     "level_kode": "STF",
19     "level_nama": "Staff/Kasir",
20     "created_at": null,
21     "updated_at": null
22   },
23 ]

```

5.

localhost/PWL_25/PWL_POS/public/api/levels

POST localhost/PWL_25/PWL_POS/public/api/levels

Params Authorization Headers (8) **Body** Scripts Settings Cookies

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key		Value	Description	Bulk Edit
<input checked="" type="checkbox"/>	level_kode	Text	SPV		
<input checked="" type="checkbox"/>	level_nama	Text	Supervisor		

Body Cookies Headers (10) Test Results 500 Internal Server Error 524 ms 1.07 MB

HTML Preview Visualize

```

1 <!DOCTYPE html>
2 <html lang="en" class="auto">
3 <!--
4 Illuminate\Database\Eloquent\MassAssignmentException: Add [level_kode] to fillable property to allow mass
  assignment on [App\Models\LevelModel]. in file
  C:\laragon\www\PWL_25\PWL_POS\vendor\laravel\framework\src\Illuminate\Database\Eloquent\Model.php on line 525

```

6.

localhost/PWL_25/PWL_POS/public/api/levels?level_kode=SPV&level_nama=Supervisor

GET localhost/PWL_25/PWL_POS/public/api/levels?level_kode=SPV&level_nama=Supervisor

Params Authorization Headers (8) **Body** Scripts Settings Cookies

Query Params

	Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/>	level_kode	SPV		
<input checked="" type="checkbox"/>	level_nama	Supervisor		

Body Cookies Headers (11) Test Results 200 OK 368 ms 902 B

{ } JSON Preview Visualize

```

24     "level_id": 5,
25     "level_kode": "CTM",
26     "level_nama": "Customer",
27     "created_at": "2025-07-09T05:39:05.000000Z",
28     "updated_at": null
29   },
30   {
31     "level_id": 7,
32     "level_kode": "SPV",
33     "level_nama": "Supervisor",
34     "created_at": "2025-07-10T05:44:57.000000Z",
35     "updated_at": null
36   }
37 ]

```

7.

PUT Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/>	level_kode	SPR		
	Key	Value	Description	

8.

		level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	CTM	Customer	2025-07-09 05:39:05	NULL
<input type="checkbox"/>	Edit Copy Delete	7	SPV	Supervisor	2025-07-10 05:44:57	NULL

localhost/PWL_25/PWL_POS/public/api/levels/7 Save Share

DELETE Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	Bulk Edit
	Key	Value	Description	

Body Cookies Headers (11) Test Results 200 OK • 413 ms • 413 B •

{ } JSON Preview Visualize

```
1 {
2   "success": true,
3   "message": "Data terhapus"
4 }
```

		level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	CTM	Customer	2025-07-09 05:39:05	NULL

Tugas

1. M_user

```
PS C:\laragon\www\PWL_25\PWL_POS> php artisan make:controller Api/UserController

INFO Controller [C:\laragon\www\PWL_25\PWL_POS\app\Http\Controllers\Api\UserController.php] created successfully.

PWL_POS > app > Http > Controllers > Api > UserController.php > PHP Intelephense > UserController

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\UserModel;
8
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         return UserModel::all();
14     }
15
16     public function store(Request $request)
17     {
18         $user = UserModel::create($request->all());
19         return response()->json($user, 201);
20     }
21
22     public function show(UserModel $user)
23     {
24         return $user;
25     }
26
27     public function update(Request $request, UserModel $user)
28     {
29         $user->update($request->all());
30         return $user;
31     }
32
33     public function destroy(UserModel $user)
34     {
35         $user->delete();
36         return response()->noContent();
37     }
38 }
39
40 Route::get('/users', [UserController::class, 'index']);
41 Route::post('/users', [UserController::class, 'store']);
42 Route::get('/users/{user}', [UserController::class, 'show']);
43 Route::put('/users/{user}', [UserController::class, 'update']);
44 Route::delete('/users/{user}', [UserController::class, 'destroy']);
```

2. M_kategori

```
PS C:\laragon\www\PWL_25\PWL_POS> php artisan make:controller Api/KategoriController
```

```
INFO Controller [C:\laragon\www\PWL_25\PWL_POS\app\Http\Controllers\Api\KategoriController.php] created successfully.
```

```
PWL_POS > app > Http > Controllers > Api > KategoriController.php > PHP Intelephense > KategoriController
```

```
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\KategoriModel;
8
9  class KategoriController extends Controller
10 {
11     public function index()
12     {
13         return KategoriModel::all();
14     }
15
16     public function store(Request $request)
17     {
18         $kategori = KategoriModel::create($request->all());
19         return response()->json($kategori, 201);
20     }
21
22     public function show(KategoriModel $kategori)
23     {
24         return $kategori;
25     }
26
27     public function update(Request $request, KategoriModel $kategori)
28     {
29         $kategori->update($request->all());
30         return $kategori;
31     }
32
33     public function destroy(KategoriModel $kategori)
34     {
35         $kategori->delete();
36         return response()->json([], 204);
37     }
38 }
```

```
44 Route::get('/kategoris', [KategoriController::class, 'index']);
45 Route::post('/kategoris', [KategoriController::class, 'store']);
46 Route::get('/kategoris/{kategori}', [KategoriController::class, 'show']);
47 Route::put('/kategoris/{kategori}', [KategoriController::class, 'update']);
48 Route::delete('/kategoris/{kategori}', [KategoriController::class, 'destroy']);
```

3. M_barang

```
PS C:\laragon\www\PWL_25\PWL_POS> php artisan make:controller Api/BarController

INFO Controller [C:\laragon\www\PWL_25\PWL_POS\app\Http\Controllers\Api\BarController.php] created successfully.

PWL_POS > app > Http > Controllers > Api > BarangController.php > PHP Intelephense > BarangController

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\BarangModel;
8
9  class BarangController extends Controller
10 {
11     public function index()
12     {
13         return BarangModel::all();
14     }
15
16     public function store(Request $request)
17     {
18         $barang = BarangModel::create($request->all());
19         return response()->json($barang, 201);
20     }
21
22     public function show(BarangModel $barang)
23     {
24         return $barang;
25     }
26
27     public function update(Request $request, BarangModel $barang)
28     {
29         $barang->update($request->all());
30         return $barang;
31     }
32 }

51 Route::get('/barangs', [BarController::class, 'index']);
52 Route::post('/barangs', [BarController::class, 'store']);
53 Route::get('/barangs/{barang}', [BarController::class, 'show']);
54 Route::put('/barangs/{barang}', [BarController::class, 'update']);
55 Route::delete('/barangs/{barang}', [BarController::class, 'destroy']);
```