

LAPORAN TUGAS
MATA KULIAH APPLICATION PROGRAMMING INTERFACE (API)
DOSEN PENGAMPU
GELAR ADITYA PRATAMA, S.T., M.KOM.



Nama : Luthfi Nur Ramadhan

Kelas : IF – B

NPM : 2142430

PROGRAM STUDI 'TEKNIK INFORMATIKA'
STMIK AMIK BANDUNG

Buatlah API menggunakan framework Laravel dan database MySQL

dengan topik bebas, asalkan bukan seperti yang dicontohkan di kelas (mahasiswa, produk, pendaftaran, dsb)

Membuat endpoint di bawah ini :

1. Endpoint / (berisi swagger documentation)
2. Endpoint register
3. Endpoint login (mendapatkan token)
4. Endpoint <sesuaitopik>
 1. GET all data
 2. GET data by id
 3. POST data (mengharuskan pakai token)
 4. PUT data by id (mengharuskan pakai token)
 5. DELETE data by id (mengharuskan pakai token)

Dalam project laravel, harus memiliki :

1. Controller Auth
2. Controller <sesuai topik>
3. Resources <sesuai topik>
4. Model User
5. Model <sesuai topik>
6. Migration <sesuai topik>
7. Route API auth
8. Route API <sesuai topik>

Yang harus dikumpulkan :

1. Screenshot seluruh code (controller, model, migration, route)
2. Screenshot swagger documentation
3. Screenshot seluruh hasil api (boleh dari swagger / dari postman)
4. Rename nama file pdf dengan format "UAS API - <NIM> - <NAMA>"

Dikumpulkan paling lambat tanggal 5 Juni 2023 jam 23:59

Topik sama dengan mahasiswa lain tidak apa apa, asalkan kolom-kolom databasenya beda

Tapi kalau codingan sama dengan mahasiswa lain, nilai = 0

Screenshot Code :

.env

```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:ozzGemyrysGvwXcMLq2y3lwF0KJtN4cs202we/7Xk2Y=
4 APP_DEBUG=true
5 APP_URL=http://uas_api_laravel.test
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=uas_api_laravel
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
```

AppServiceProvider.php

```
app > Providers > AppServiceProvider.php > AppServiceProvider > boot
1  <?php
2
3  namespace App\Providers;
4
5  use Illuminate\Support\Facades\Schema;
6  use Illuminate\Support\ServiceProvider;
7
8  class AppServiceProvider extends ServiceProvider
9  {
10 >     /** ...
13     public function register(): void
14     {
15         //
16     }
17
18 >     /** ...
21     // jumlah string 255
22     public function boot(): void
23     {
24         Schema::defaultStringLength(Length: 255);
25     }
26 }
27
```

2023_06_29_034851_create_obats_table.php (Database Migration)

```
database > migrations > 2023_06_29_034851_create_obats_table.php > class > up > Closure
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      /** ...
12     public function up(): void
13     {
14         Schema::create( table: 'obats', callback: function (Blueprint $table) {
15             $table->id();
16             $table->string( column: 'nama' );
17             $table->string( column: 'gambar' );
18             $table->string( column: 'tipe' );
19             $table->text( column: 'deskripsi' );
20             $table->string( column: 'produksi' );
21             $table->double( column: 'harga' );
22             $table->date( column: 'kadaluarsa' );
23             $table->timestamps();
24         });
25     }
26
27     /** ...
30     public function down(): void
31     {
32         Schema::dropIfExists( table: 'obats' );
33     }
34 };
35
```

Obat.php (Model)

```
app > Models > Obat.php > $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 Obat extends Model
9  {
10     use HasFactory;
11
12     protected $guarded = ['id'];
13
14     protected $fillable = [
15         'nama',
16         'tipe',
17         'gambar',
18         'deskripsi',
19         'produksi',
20         'harga',
21         'kadaluarsa',
22     ];
23 }
24
```

ObatResource.php (Resources)

```
app > Http > Resources > ObatResource.php > ObatResource > toArray
1  <?php
2
3  namespace App\Http\Resources;
4
5  use Illuminate\Http\Resources\Json\JsonResource;
6
7  class ObatResource extends JsonResource
8  {
9      // Membuat Variabel Public
10     public $status;
11     public $message;
12     public $pagination;
13
14     // Membuat Constructor
15     public function __construct($status, $message, $resource, $pagination)
16     {
17         parent::__construct(resource: $resource);
18         $this->status = $status;
19         $this->message = $message;
20         $this->pagination = $pagination;
21     }
22
23     /** ...
24
25     public function toArray($request): array
26     {
27         // return parent::toArray($request);
28         return [
29             'success' => $this->status,
30             'message' => $this->message,
31             'data' => $this->resource,
32             'pagination' => $this->pagination
33         ];
34     }
35 }
```

ObatController.php (Controller)

index

```
app > Http > Controllers > Api > ObatController.php > ObatController > index
10
11 class ObatController extends Controller
12 {
13     # auth
14     public function __construct()
15     {
16         $this->middleware( middleware: 'auth:api', options: ['except' => ['index', 'show']]);
17     }
18
19 > /**...
86 # Index / Showing ALL Data
87 public function index(Request $request)
88 {
89     # parameter
90     $where = $request->has( key: 'where' ) ? $request->get( key: 'where' ) : '{}';
91     $sort = $request->has( key: 'sort' ) ? $request->get( key: 'sort' ) : 'id:asc';
92     $per_page = $request->has( key: 'per_page' ) ? $request->get( key: 'per_page' ) : 2;
93     $page = $request->has( key: 'page' ) ? $request->get( key: 'page' ) : 1;
94     $count = $request->has( key: 'count' ) ? $request->get( key: 'count' ) : false;
95     $search = $request->has( key: 'search' ) ? $request->get( key: 'search' ) : '';
96
97     # prepare parameter
98     $sort = explode( separator: ':', string: $sort);
99     $where = str_replace( search: '"', replace: '\"', subject: $where);
100     $where = json_decode( json: $where, associative: true);
101
102     # query get
103     $query = Obat::where([[ 'id', '>', '0' ]]);
104
105     # query where
106     if($where) {
107         foreach($where as $key => $value) {
108             $query = $query->where([[ $key, "=", $value ]]);
109         }
110     }
111 }
```


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

```
112     # query search
113     if($search) {
114         $query = $query->where([[ 'nama', 'like', '% ' . $search . '% ']]);
115         $query = $query->orWhere([[ 'deskripsi', 'like', '% ' . $search . '% ']]);
116         $query = $query->orWhere([[ 'harga', 'like', '% ' . $search . '% ']]);
117     }
118
119     # variabel data
120     $datas = [];
121
122     # pagination
123     $pagination = [];
124     $pagination['page'] = (int)$page;
125     $pagination['per_page'] = (int)$per_page;
126     $pagination['total_data'] = $query->count( mixed: 'id');
127     $pagination['total_page'] = ceil( num: $pagination['total_data'] / $pagination['per_page']);
128
129     # count
130     if($count == true) {
131         $query = $query->count( mixed: 'id');
132         $datas['count'] = $query;
133     } else { # Get Data
134         $query = $query
135             ->orderBy($sort[0], $sort[1])
136             ->limit($per_page)
137             ->offset(($page - 1) * $per_page)
138             ->get()
139             ->toArray();
140
141         foreach($query as $qry) {
142             $temp = $qry;
143
144             $created_at_indo = Carbon::parse( time: $temp['created_at']);
145             $created_at_indo->locale( locale: 'id')->settings( settings: ['formatFunction' => 'translatedFormat'])
```

```
146             ->toArray();
147
148             foreach($query as $qry) {
149                 $temp = $qry;
150
151                 $created_at_indo = Carbon::parse( time: $temp['created_at']);
152                 $created_at_indo->locale( locale: 'id')->settings( settings: ['formatFunction' => 'translatedFormat']);
153
154                 $temp['created_date_indo'] = $created_at_indo->format( format: 'l, d F Y H:i:s');
155                 array_push( &array: $datas, values[0]: $temp);
156             }
157
158         }
159
160     }
161
162     return new ObatResource( status: true, message: 'Get Data Successfull', resource: $datas, pagination: $pagination)
163 }
```

Show data by id

```
app > Http > Controllers > Api > ObatController.php > ObatController > show
155 > /** ...
186 # Show Data By ID
187 public function show($id)
188 {
189     # query get by id
190     $query = Obat::find($id);
191
192     # variabel data
193     $datas = $query;
194
195     # pagination
196     $pagination = [];
197
198     return new ObatResource( status: true, message: "Get Data By Id Successfull", resource: $datas, pagination: $p
199 }
```

Insert Data

```
app > Http > Controllers > Api > ObatController.php > ObatController > store
259 # Insert Data
260 public function store(Request $request)
261 {
262     # query insert
263     $query = Obat::create($request->all());
264
265     # variabel data
266     $datas = $query;
267
268     # pagination
269     $pagination = [];
270
271     return new ObatResource( status: true, message: "Insert Data Successfull", resource: $datas, pagination: $pagi
272 }
```

Update Data by Id

```
app > Http > Controllers > Api > ObatController.php > ObatController > update
274 > /** ...
341 # Put By Id / Update Data By Id
342 public function update(Request $request, $id)
343 {
344     # query insert
345     $query = Obat::findOrFail($id);
346     $query = $query->update($request->all());
347
348     # get after update
349     $query = Obat::findOrFail($id);
350
351     # variabel data
352     $datas = $query;
353
354     # pagination
355     $pagination = [];
356
357     return new ObatResource( status: true, message: "Update Data Successfull", resource: $datas, pagination: $pagi
358 }
```

Destroy by id

```
app > Http > Controllers > Api > ObatController.php > ObatController > destroy
360 > /** ...
392 # Delete By Id
393 public function destroy($id)
394 {
395     # query insert
396     $query = Obat::findOrFail($id);
397     $query = $query->delete();
398
399     # variabel data
400     $datas = $query;
401
402     # pagination
403     $pagination = [];
404
405     return new ObatResource( status: true, message: "Delete Data Successfull", resource: $datas, pagination: $pagi
406 }
407
408
```

Api.php (Route)

```
routes > api.php >
19 Route::middleware( middleware: 'auth:sanctum' )->get( uri: '/user', action: function (Request $request) {
20     return $request->user();
21 });
22
23 Route::controller( controller: AuthController::class )->group( callback: function() {
24     Route::post( uri: 'login', action: 'login' );
25     Route::post( uri: 'register', action: 'register' );
26     Route::post( uri: 'logout', action: 'logout' );
27     Route::post( uri: 'refresh', action: 'refresh' );
28 });
29
30 // Route::apiResource( '/obat', ObatController::class );
31 Route::controller( controller: ObatController::class )->group( callback: function() {
32     Route::get( uri: 'obat', action: 'index' );
33     Route::get( uri: 'obat/{id}', action: 'show' );
34     Route::post( uri: 'obat', action: 'store' );
35     Route::put( uri: 'obat/{id}', action: 'update' );
36     Route::delete( uri: 'obat/{id}', action: 'destroy' );
37 });
38
39 Route::fallback( action: function() {
40     return response()->json( data: [
41         'status' => false,
42         'message' => 'Page Not Found'
43     ], status: 404 );
44 });
```

Composer.json (Import library l5-swagger & jwt-auth)

```
{> composer.json > {} require
1  {
2      "name": "laravel/laravel",
3      "type": "project",
4      "description": "The skeleton application for the Laravel framework.",
5      "keywords": ["laravel", "framework"],
6      "license": "MIT",
7      "require": {
8          "php": "^8.1",
9          "darkaonline/l5-swagger": "^8.5",
10         "guzzlehttp/guzzle": "^7.2",
11         "laravel/framework": "^10.10",
12         "laravel/sanctum": "^3.2",
13         "laravel/tinker": "^2.8",
14         "php-open-source-saver/jwt-auth": "^2.1"
15     },
16     "require-dev": {
```

jwt.php

```
{> composer.json  jwt.php X
config > jwt.php
17
18     'secret' => env('key: 'JWT_SECRET'),
19
20     /*
21     /-----
22     / JWT Authentication Keys
```

.env

```
{> composer.json  jwt.php .env X
.env
56 VITE_PUSHER_PORT="${PUSHER_PORT}"
57 VITE_PUSHER_SCHEME="${PUSHER_SCHEME}"
58 VITE_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
59
60 JWT_SECRET=uECS0Z2MNCh0ceEUVq6tkgwAtsqSdg8DSZgI4U9DawHPScyFoZbf2K5XephtEeep
61
62 JWT_ALGO=HS256
63
```

auth.php (config)

```
composer.json  jwt.php  .env  auth.php X
config > auth.php
15
16     'defaults' => [
17         'guard' => 'api',
18         'passwords' => 'users',
19     ],
20
21     /*
22     |-----
23     | Authentication Guards
24     |-----
25     |
26     | Next, you may define every authentication guard for your application.
27     | Of course, a great default configuration has been defined for you
28     | here which uses session storage and the Eloquent user provider.
29     |
30     | All authentication drivers have a user provider. This defines how the
31     | users are actually retrieved out of your database or other storage
32     | mechanisms used by this application to persist your user's data.
33     |
34     | Supported: "session"
35     |
36     */
37
38     'guards' => [
39         'web' => [
40             'driver' => 'session',
41             'provider' => 'users',
42         ],
43
44         'api' => [
45             'driver' => 'jwt',
46             'provider' => 'users',
47         ],
48     ],
```

User.php (Model)

app > Models > User.php > User

```
12 class User extends Authenticatable implements JWTSubject
13 {
14     use HasApiTokens, HasFactory, Notifiable;
15
16     /** ...
21     protected $fillable = [
22         'name',
23         'email',
24         'password',
25     ];
26
27     /** ...
32     protected $hidden = [
33         'password',
34         'remember_token',
35     ];
36
37     /** ...
42     protected $casts = [
43         'email_verified_at' => 'datetime',
44     ];
45
46     public function getJWTIdentifier()
47     {
48         return $this->getKey();
49     }
50
51     public function getJWTCustomClaims()
52     {
53         return [];
54     }
55 }
```

AuthController.php

Login

```
app > Http > Controllers > Api > AuthController.php > AuthController > login
11 class AuthController extends Controller
12 {
13     public function __construct()
14     {
15         $this->middleware( middleware: 'auth:api', options: ['except' => ['login', 'register']]);
16     }
17
18     /** ...
56     public function login(Request $request)
57     {
58         $request->validate( rules: [
59             'email' => 'required|string|email',
60             'password' => 'required|string',
61         ]);
62         $credentials = $request->only( keys: 'email', 'password');
63
64         $token = Auth::attempt( credentials: $credentials);
65         if(!$token) {
66             return response()->json( data: [
67                 'status' => 'error',
68                 'message' => 'Unauthorized',
69             ], status: 401);
70         }
71
72         $user = Auth::user();
73         return response()->json( data: [
74             'status' => "success",
75             'data' => $user,
76             'auth' => [
77                 'token' => $token,
78                 'type' => 'bearer',
79             ]
80         ]);
81     }
```

Register

```
app > Http > Controllers > Api > AuthController.php > AuthController > register
83 > /** ...
92 > *          mediaType="multipart/form-data", ...
124 // Register
125 public function register(Request $request)
126 {
127     $request->validate( rules: [
128         'name' => 'required|string|max:255',
129         'email' => 'required|string|email|max:255|unique:users',
130         'password' => 'required|string|min:8',
131     ]);
132
133     $user = User::create([
134         'name' => $request->name,
135         'email' => $request->email,
136         'password' => Hash::make( value: $request->password),
137     ]);
138
139     $token = Auth::login( user: $user);
140     return response()->json( data: [
141         'status' => 'success',
142         'message' => 'User Created Successfull',
143         'data' => $user,
144         'auth' => [
145             'token' => $token,
146             'type' => 'bearer',
147         ]
148     ]);
149 }
```

logout

```
app > Http > Controllers > Api > AuthController.php > AuthController > logout
152 public function logout()
153 {
154     Auth::logout();
155     return response()->json( data: [
156         'status' => 'success',
157         'message' => 'Successfully Logged out',
158     ]);
159 }
```


Refresh token

```
app > Http > Controllers > Api > AuthController.php > AuthController > refresh
161 // refresh token
162 public function refresh()
163 {
164     return response()->json( data: [
165         'status' => 'success',
166         'message' => 'Token Refresh',
167         'data' => Auth::user(),
168         'auth' => [
169             'token' => Auth::refresh(),
170             'type' => 'bearer',
171         ]
172     ]);
173 }
174 }
```

Authenticate.php

```
User.php AuthController.php Authenticate.php X
app > Http > Middleware > Authenticate.php > Authenticate
1 <?php
2
3 namespace App\Http\Middleware;
4
5 use Illuminate\Auth\Middleware\Authenticate as Middleware;
6 use Illuminate\Http\Request;
7
8 class Authenticate extends Middleware
9 {
10     /**
11      * Get the path the user should be redirected to when they are not authenticated.
12      */
13
14 }
15 }
```

Handler.php

```
User.php X AuthController.php Authenticate.php Handler.php X
app > Exceptions > Handler.php > Handler > render
30
31 public function render($request, Throwable $exception)
32 {
33     return response()->json( data: [
34         'status' => 'false',
35         'message' => '(Handler)Error::' . $exception->getMessage(),
36         'data' => []
37     ]);
38 }
39
40
```

Web.php

```
web.php X
routes > web.php
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 /*
6 |-----
7 | Web Routes
8 |-----
9 |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider and all of them will
12 | be assigned to the "web" middleware group. Make something great!
13 |
14 */
15
16 // Route::get('/', function () {
17 //     return view('welcome');
18 // });
19
```

Swagger :

The image shows two screenshots of the Swagger UI interface. The top screenshot shows the initial state where the 'auth' and 'Obat' tags are collapsed. The bottom screenshot shows the 'auth' and 'Obat' tags expanded, revealing the API endpoints for each.

UAS Laravel API Obat Documentation 1.0.0 OAuth 3.0

http://uas_api_laravel.test/docs/api-docs.json

Dibuat oleh Luthfi Nur Ramadhan 🌟 - 2142430

[Contact the developer](#)

Apache 2.0

[Authorize](#)

Filter by tag

auth

- POST** /api/login
- POST** /api/register

Obat

- GET** /api/obat
- POST** /api/obat
- GET** /api/obat/{id}
- PUT** /api/obat/{id}
- DELETE** /api/obat/{id}

```
/api/register
```

[illegible]

<div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div>											id	name	email	email_verified_at	password	remember_token	created_at	updated_at
<div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div>	<div><div></div><div></div></div> Edit	<div><div></div><div></div></div> Copy	<div><div></div><div></div></div> Delete	1	admin	admin123@gmail.com	NULL	\$2y\$10\$X3YsDzuxm/yjE73ZN/TumPmM/x/Qkrvg4vtY7e3e...	NULL	2023-06-30 02:25:30	2023-06-30 02:25:30							
<div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div>	<div><div></div><div></div></div> Edit	<div><div></div><div></div></div> Copy	<div><div></div><div></div></div> Delete	2	Luthfi Nur Ramadhan	luthfiramadhan.lr55@gmail.com	NULL	\$2y\$10\$QWdXcPinOgZPFWKG/8toUek9XJov/q6BjCERS30px...	NULL	2023-06-30 03:24:31	2023-06-30 03:24:31							
<div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></div> <div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div></div><div></div></div></</div>																		

```
/api/login
```

Login Ketika Salah

Code	Details
401	<p>Error: Unauthorized</p> <p>Response body</p> <pre>{ "status": "error", "message": "Unauthorized" }</pre> <p>Response headers</p> <pre>access-control-allow-origin: * cache-control: no-cache,private connection: keep-alive content-type: application/json date: Fri, 30 Jun 2023 03:27:12 GMT server: nginx/1.19.5 transfer-encoding: chunked x-powered-by: PHP/8.2.4 x-ratelimit-limit: 60 x-ratelimit-remaining: 59</pre>

Login Ketika Benar (*akan mendapatkan token)

The screenshot shows a REST client interface with a 'Details' tab selected. The status code is 200. The response body is a JSON object indicating a successful login. The response headers include standard web headers like 'access-control-allow-origin', 'cache-control', and 'content-type'.

```
Code    Details
200
Response body
{
  "status": "success",
  "data": {
    "id": 1,
    "name": "Luthfi Nur Ramadhan",
    "email": "luthfiramadhan.lf55@gmail.com",
    "email_verified_at": null,
    "created_at": "2023-06-30T03:24:31.000000Z",
    "updated_at": "2023-06-30T03:24:31.000000Z"
  },
  "auth": {
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWQ9PSY3ZjhmYStlbnRlc3QyYXNpL2xvZ2UuIiwiaWF0IjoxNjg4MDk1NjcxLjE2eHA1OjE2ODg0OTkxNzEsIm51Zi11GHTY4ODASNTY3MSwianRpIjo1MWhGUlZlMGxBODRlQzV3OS51SiN1Y1IiG1j1i1CwcnY1O1YpM2JkMmM4OTQ5ZjYhNGFkyjMSZTc0MmM0MmM4MzJkYjd0NTk3NmY3In0.Q70w-qtFpVCAGi0WkQ9VFQRSPS6ZRFbCLpKpfsvD01",
    "type": "bearer"
  }
}
Response headers
access-control-allow-origin: *
cache-control: no-cache,private
connection: keep-alive
content-encoding: gzip
content-type: application/json
date: Fri, 30 Jun 2023 03:27:51 GMT
server: nginx/1.19.5
transfer-encoding: chunked
vary: Accept-Encoding
x-powered-by: PHP/8.2.4
x-ratelimit-limit: 60
x-ratelimit-remaining: 58
Responses
```

Copy & paste token login disini

The screenshot shows a dialog box titled 'Available authorizations'. It lists 'bearerAuth (http, Bearer)' with a 'Value' field containing the token 'QR5PSk6ZRFbCLpKpfsvD01'. There are 'Authorize' and 'Close' buttons.

```
Available authorizations
x
bearerAuth (http, Bearer)
Value:
QR5PSk6ZRFbCLpKpfsvD01
Authorize Close
```

Jika mencoba melakukan Post tanpa memasukkan token login

The screenshot shows a REST client interface with a 'Details' tab selected. The status code is 200. The response body is a JSON object indicating an unauthorized access. The response headers are identical to the previous screenshot.

```
Request URL
http://uas_api_laravel.test/api/obat
Server response
Code    Details
200
Response body
{
  "status": "false",
  "message": "(Handler)Error::unauthenticated.",
  "data": []
}
Response headers
access-control-allow-origin: *
cache-control: no-cache,private
connection: keep-alive
content-encoding: gzip
content-type: application/json
date: Fri, 30 Jun 2023 03:30:30 GMT
server: nginx/1.19.5
transfer-encoding: chunked
vary: Accept-Encoding
x-powered-by: PHP/8.2.4
```

Data di Database

The screenshot shows the phpMyAdmin interface for the 'uas_api_laravel' database. The 'obats' table is selected, and the query results are displayed. The table contains 3 rows of data.

	id	nama	gambar	type	deskripsi	produksi	harga	kadaluarsa	created_at	updated_at
<input type="checkbox"/>	1	Hansaplast (1 Box)	hansaplast-box.jpg	Obat Luka	Hansaplast merupakan plaster penutup luka untuk me...	PT. Beiersdorf Indonesia	12900	2024-12-30	2023-06-30 02:28:21	2023-06-30 02:28:21
<input type="checkbox"/>	2	Insto 7.5ml	insto75ml.jpg	Obat Mata	Hansaplast merupakan plaster penutup luka untuk me...	PT PHARMA HEALTH CARE	16500	2024-12-30	2023-06-30 02:31:07	2023-06-30 02:31:07
<input type="checkbox"/>	3	Siladex	Siladex.jpg	Obat Batuk	SILADEX ANTITUSSIVE SIRUP merupakan obat yang meng...	PT Konimex	12500	2024-12-30	2023-06-30 02:33:05	2023-06-30 02:33:05

Jika mencoba melakukan Put tanpa memasukkan token login

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

- Request URL:** `http://uas_api_laravel.test/api/obat/1`
- Server response:** 200
- Response body:**

```
{
  "status": "false",
  "message": "(Handler)Error::Unauthenticated.",
  "data": []
}
```
- Response headers:**

```
access-control-allow-origin: *
cache-control: no-cache, private
connection: keep-alive
content-encoding: gzip
content-type: application/json
date: Fri, 30 Jun 2023 03:32:09 GMT
server: nginx/1.19.5
transfer-encoding: chunked
vary: Accept-Encoding
x-powered-by: PHP/8.2.4
```

Jika mencoba melakukan Delete tanpa memasukkan token login

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

- Request URL:** `http://uas_api_laravel.test/api/obat/1`
- Server response:** 200
- Response body:**

```
{
  "status": "false",
  "message": "(Handler)Error::Unauthenticated.",
  "data": []
}
```
- Response headers:**

```
access-control-allow-origin: *
cache-control: no-cache, private
connection: keep-alive
content-encoding: gzip
content-type: application/json
date: Fri, 30 Jun 2023 03:38:38 GMT
server: nginx/1.19.5
transfer-encoding: chunked
vary: Accept-Encoding
x-powered-by: PHP/8.2.4
```

Jika Tanpa memasukkan token login kita hanya bisa melakukan Get All Data & Get Data by Id

Get All Data

Request URL

http://uas_api_laravel.test/api/obat?per_page=5

Server response

Code

Details

200

Response body

```
{
  "success": true,
  "message": "Get Data Successfull",
  "data": [
    {
      "id": 1,
      "nama": "Hansaplast (1 Box)",
      "gambar": "hansaplast-box.jpg",
      "type": "Obat luka",
      "deskripsi": "Hansaplast merupakan plaster penutup luka untuk mencegah infeksi pada luka selama proses penyembuhan. Produk ini dibuat dari bahan elastis yang akan mengikuti tiap lekukan kulit sehingga menutup luka dari kotoran dan bakteri, kedap air, steril, dan tidak melekat pada luka.",
      "produksi": "PT. Beiersdorf Indonesia",
      "harga": 12000,
      "kadaluarsa": "2024-12-30",
      "created_at": "2023-06-30T02:28:21.000000Z",
      "updated_at": "2023-06-30T02:28:21.000000Z",
      "created_date_indo": "Jumat, 30 Juni 2023 02:28:21"
    },
    {
      "id": 2,
      "nama": "Insto 7.5ml",
      "gambar": "insto75ml.jpg",
      "type": "Obat Mata",
      "deskripsi": "Hansaplast merupakan plaster penutup luka untuk mencegah infeksi pada luka selama proses penyembuhan. Produk ini dibuat dari bahan elastis yang akan mengikuti tiap lekukan kulit sehingga menutup luka dari kotoran dan bakteri, kedap air, steril, dan tidak melekat pada luka.",
      "produksi": "PT PHARMA HEALTH CARE",
      "harga": 16500,
      "kadaluarsa": "2024-12-30",
      "created_at": "2023-06-30T02:28:21.000000Z",
      "updated_at": "2023-06-30T02:28:21.000000Z",
      "created_date_indo": "Jumat, 30 Juni 2023 02:28:21"
    }
  ]
}
```

Download

Response headers

```
access-control-allow-origin: *
cache-control: no-cache,private
connection: keep-alive
content-encoding: gzip
content-type: application/json
date: Fri, 30 Jun 2023 03:40:21 GMT
server: nginx/1.19.5
transfer-encoding: chunked
vary: Accept-Encoding
```

Get Data by Id

Request URL

http://uas_api_laravel.test/api/obat/1

Server response

Code

Details

200

Response body

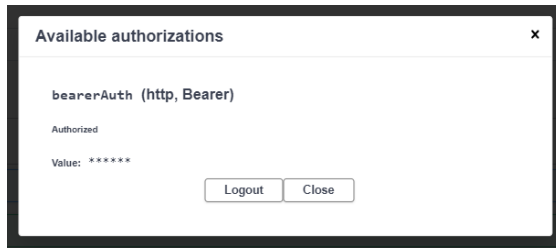
```
{
  "success": true,
  "message": "Get Data By Id Successfull",
  "data": {
    "id": 1,
    "nama": "Hansaplast (1 Box)",
    "gambar": "hansaplast-box.jpg",
    "type": "Obat luka",
    "deskripsi": "Hansaplast merupakan plaster penutup luka untuk mencegah infeksi pada luka selama proses penyembuhan. Produk ini dibuat dari bahan elastis yang akan mengikuti tiap lekukan kulit sehingga menutup luka dari kotoran dan bakteri, kedap air, steril, dan tidak melekat pada luka.",
    "produksi": "PT. Beiersdorf Indonesia",
    "harga": 12000,
    "kadaluarsa": "2024-12-30",
    "created_at": "2023-06-30T02:28:21.000000Z",
    "updated_at": "2023-06-30T02:28:21.000000Z"
  },
  "pagination": []
}
```

Download

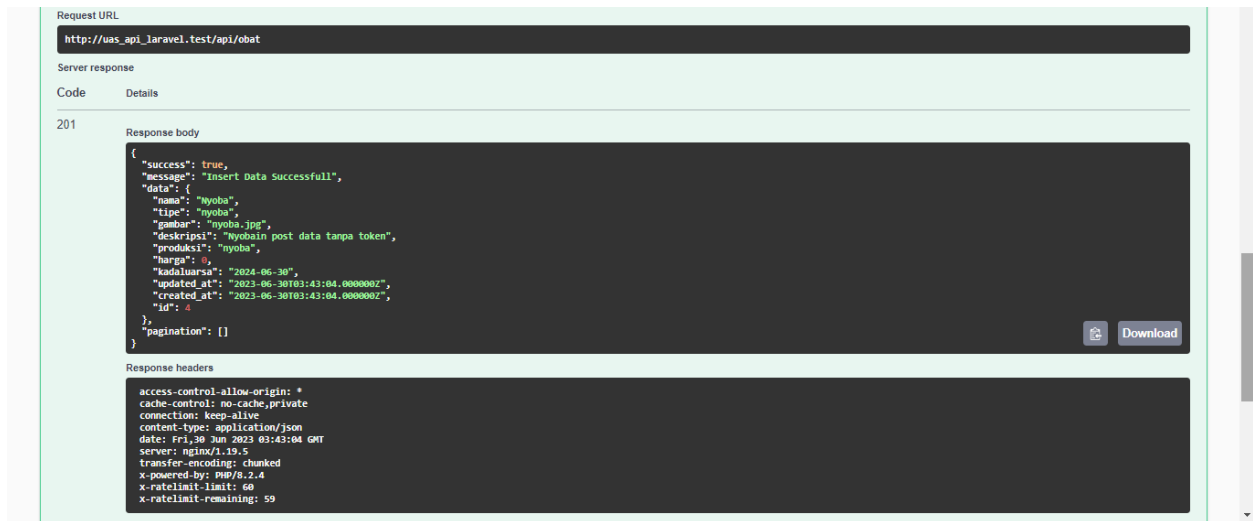
Response headers

```
access-control-allow-origin: *
cache-control: no-cache,private
connection: keep-alive
content-encoding: gzip
content-type: application/json
date: Fri, 30 Jun 2023 03:41:02 GMT
server: nginx/1.19.5
transfer-encoding: chunked
vary: Accept-Encoding
x-powered-by: PHP/8.2.4
x-ratelimit-limit: 60
x-ratelimit-remaining: 58
```

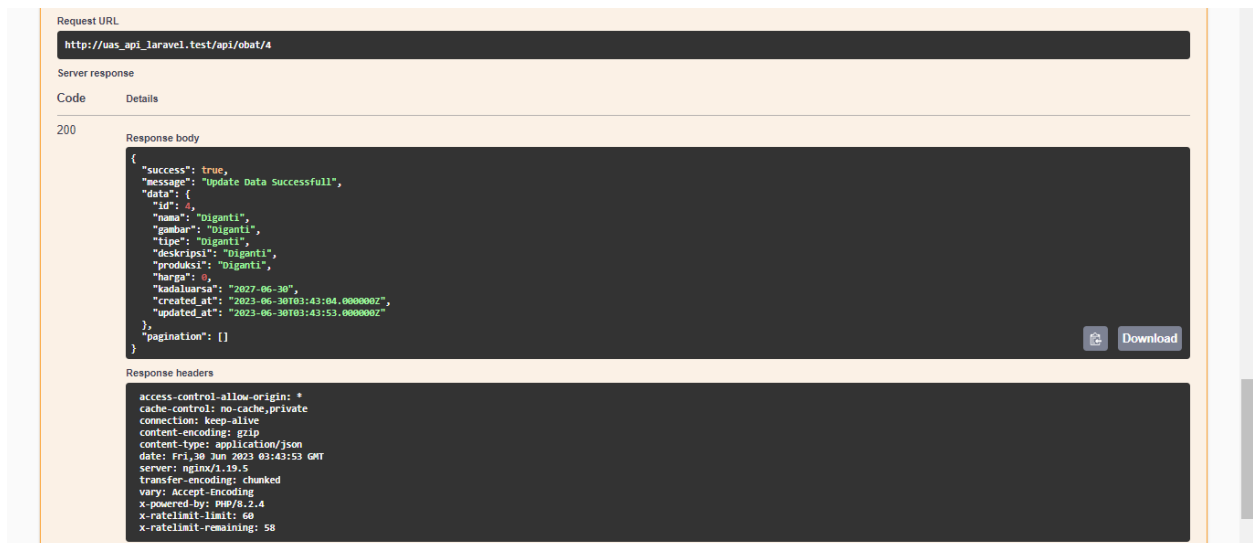
Jika User Memasukkan Token Login Maka User Akan bisa melakukan Post, Put, Delete



Post - Insert Data Baru



Put – Update Data Baru {id}



Request URL
`http://uas_api_laravel.test/api/obat/per_page=5`

Server response

Code Details

200

Response body

```
{
  "harga": 13500,
  "kadaluarsa": "2024-12-30",
  "created_at": "2023-06-30T02:33:05.000000Z",
  "updated_at": "2023-06-30T02:33:05.000000Z",
  "created_date_indo": "Jumat, 30 Juni 2023 02:33:05"
},
{
  "id": 4,
  "nama": "Diganti",
  "gambar": "Diganti",
  "tipe": "Diganti",
  "deskripsi": "Diganti",
  "produksi": "Diganti",
  "harga": 0,
  "kadaluarsa": "2027-06-30",
  "created_at": "2023-06-30T03:43:04.000000Z",
  "updated_at": "2023-06-30T03:43:53.000000Z",
  "created_date_indo": "Jumat, 30 Juni 2023 03:43:04"
}
},
"pagination": {
  "page": 1,
  "per_page": 5,
  "total_data": 4,
  "total_page": 1
}
}
```

Response headers

```
access-control-allow-origin: *
cache-control: no-cache,private
connection: keep-alive
content-encoding: gzip
content-type: application/json
date: Fri, 30 Jun 2023 03:44:30 GMT
server: nginx/1.19.5
```

Sekarang Delete Data Yang telah Dibuat

Request URL
`http://uas_api_laravel.test/api/obat/4`

Server response

Code Details

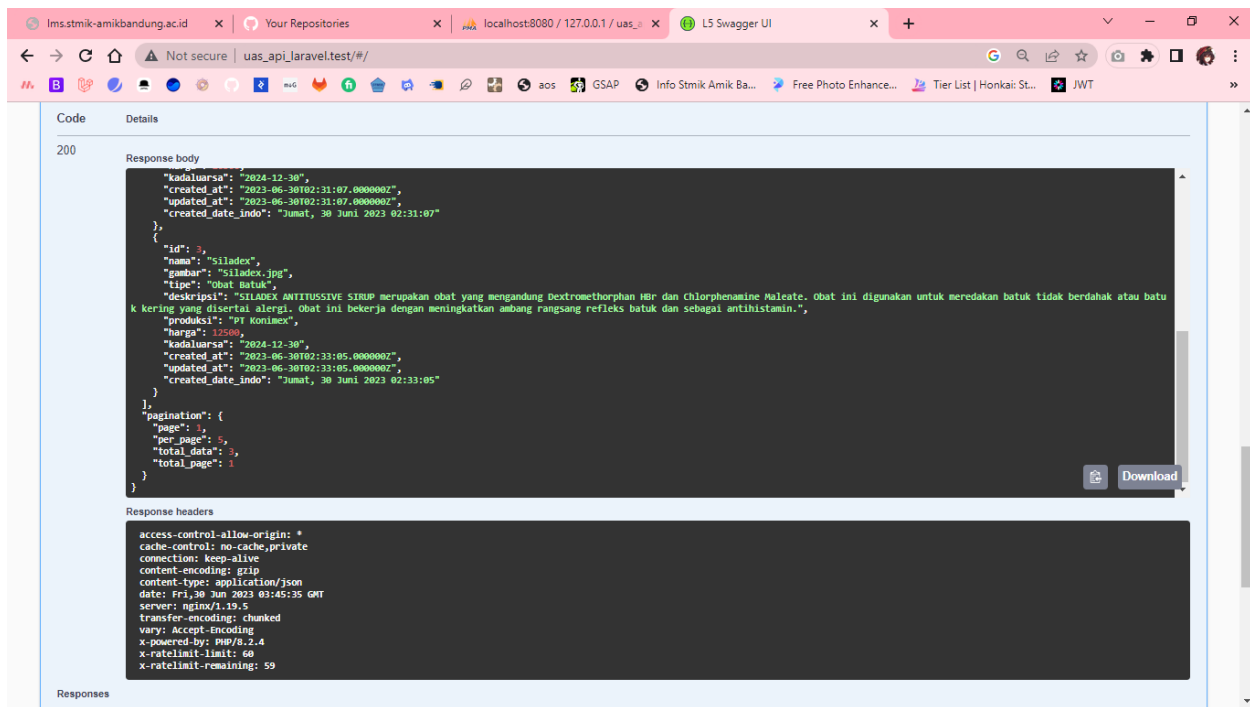
200

Response body

```
{
  "success": true,
  "message": "Delete Data Successfull",
  "data": true,
  "pagination": []
}
```

Response headers

```
access-control-allow-origin: *
cache-control: no-cache,private
connection: keep-alive
content-encoding: gzip
content-type: application/json
date: Fri, 30 Jun 2023 03:45:00 GMT
server: nginx/1.19.5
transfer-encoding: chunked
vary: Accept-Encoding
x-powered-by: PHP/8.2.4
x-ratelimit-limit: 60
x-ratelimit-remaining: 59
```



Begitu pula dengan di Postman

