

Application Programming Interface

Pertemuan 13

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



laravel api & mysql – middleware jwt & login

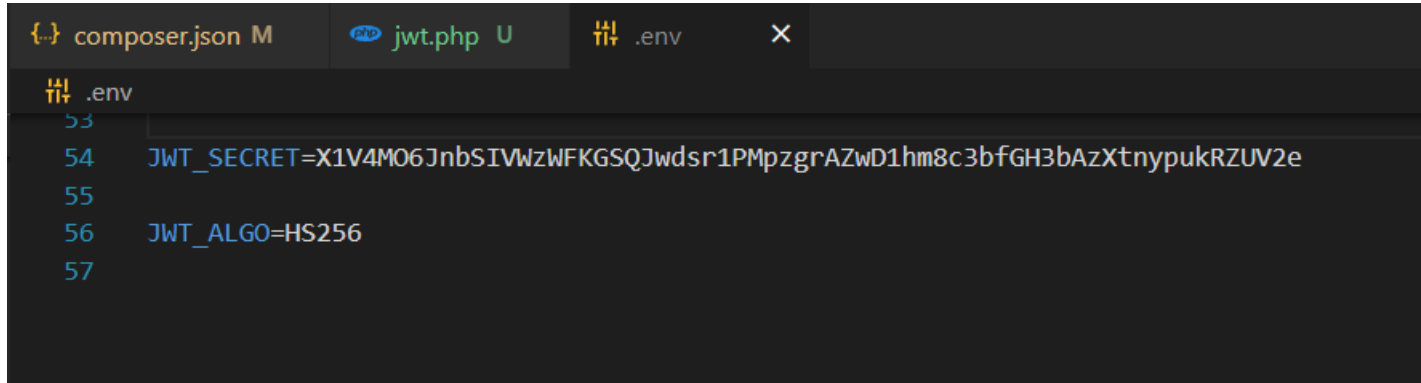
- Jalankan command
composer require php-open-source-saver/jwt-auth
- Nanti akan ada package baru tertulis di composer.json

```
{-} composer.json M X
{-} composer.json > {-} require > [add] php-open-source-saver/jwt-auth
1  {
2      .... "name": "laravel/laravel",
3      .... "type": "project",
4      .... "description": "The Laravel Framework.",
5      .... "keywords": ["framework", "laravel"],
6      .... "license": "MIT",
7      .... "require": {
8          .... "php": "^8.0",
9          .... "fruitcake/laravel-cors": "^2.0.5",
10         .... "guzzlehttp/guzzle": "^7.2",
11         .... "laravel/framework": "^9.0",
12         .... "laravel/sanctum": "^2.14",
13         .... "laravel/tinker": "^2.7",
14         .... "php-open-source-saver/jwt-auth": "^2.1"
15     },
```

- Jalankan command
php artisan vendor:publish --provider="PHPOpenSourceSaver\JWTAuth\Providers\LaravelServiceProvider"
- Nanti akan ada file baru config/jwt.php

```
composer.json M  jwt.php U X
config > jwt.php
1  <?php
2
3  return [
4      /*
5      |-----
6      | JWT Authentication Secret
7      |-----
8      |
9      | Don't forget to set this in your .env file, as it will be used to sign
10     | your tokens. A helper command is provided for this:
11     | `php artisan jwt:secret`
12     |
13     | Note: This will be used for Symmetric algorithms only (HMAC),
14     | since RSA and ECDSA use a private/public key combo (See below).
15     |
16     | */
17
18     'secret' => env('JWT_SECRET'),
```

- Jalankan command
php artisan jwt:secret
- Nanti akan ada baris config baru untuk jwt

A screenshot of a code editor with a dark theme. The top bar shows three tabs: "composer.json M", "jwt.php U", and ".env". The ".env" tab is active, showing the following configuration:

```
53  
54 JWT_SECRET=X1V4M06JnbSIVwzWFKGSQJwdsr1PMpzgrAZwD1hm8c3bfGH3bAzXtnypukRZUV2e  
55  
56 JWT_ALGO=HS256  
57
```

- Edit file config/auth.php

```
composer.json M  jwt.php A  .env  auth.php M X
config > auth.php
1  <?php
2
3  return [
4
5  >  /** ...
14  */
15
16  'defaults' => [
17      'guard' => 'api',
18      'passwords' => 'users',
19  ],
20
21  >  /** ...
36  */
37
38  'guards' => [
39      'web' => [
40          'driver' => 'session',
41          'provider' => 'users',
42      ],
43
44      'api' => [
45          'driver' => 'jwt',
46          'provider' => 'users',
47      ],
48  ],
49
```

```
app > Models > User.php > User > getJWTCustomClaims
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 }
56
```

- Jalankan command
php artisan make:controller Api/AuthController
- Edit file app/Http/Controllers/Api/AuthController.php

AuthController.php U X

app > Http > Controllers > Api > AuthController.php > {} App\Http\Controllers\Api

```
1  <?php
2
3  namespace App\Http\Controllers\Api;
4  use App\Http\Controllers\Controller;
5  use Illuminate\Http\Request;
6  use App\Models\User;
7  use Illuminate\Support\Facades\Auth;
8  use Illuminate\Support\Facades\Hash;
9
10 class AuthController extends Controller
11 {
12     ...public function __construct()
13     ...{
14     ...    ...$this->middleware('auth:api', ['except' => ['login', 'register']]);
15     ...}
16
```



app > Http > Controllers > Api > AuthController.php > {} App\Http\Controllers\Api

```
16
17 ... public function login(Request $request)
18 ... {
19 ...     $request->validate([
20 ...         'email' => 'required|string|email',
21 ...         'password' => 'required|string',
22 ...     ]);
23 ...     $credentials = $request->only('email', 'password');
24
25 ...     $token = Auth::attempt($credentials);
26 ...     if (!$token) {
27 ...         return response()->json([
28 ...             'status' => 'error',
29 ...             'message' => 'Unauthorized',
30 ...         ], 401);
31 ...     }
32
33 ...     $user = Auth::user();
34 ...     return response()->json([
35 ...         'status' => 'success',
36 ...         'data' => $user,
37 ...         'auth' => [
38 ...             'token' => $token,
39 ...             'type' => 'bearer',
40 ...         ]
41 ...     ]);
42
43 ... }
```

JWT



AuthController.php U X

app > Http > Controllers > Api > AuthController.php > {} App\Http\Controllers\Api

```
44
45     public function register(Request $request){
46         $request->validate([
47             'name' => 'required|string|max:255',
48             'email' => 'required|string|email|max:255|unique:users',
49             'password' => 'required|string|min:6',
50         ]);
51
52         $user = User::create([
53             'name' => $request->name,
54             'email' => $request->email,
55             'password' => Hash::make($request->password),
56         ]);
57
58         $token = Auth::login($user);
59         return response()->json([
60             'status' => 'success',
61             'message' => 'User created successfully',
62             'data' => $user,
63             'auth' => [
64                 'token' => $token,
65                 'type' => 'bearer',
66             ]
67         ]);
68     }
```

JWT



AuthController.php U X

app > Http > Controllers > Api > AuthController.php > {} App\Http\Controllers\Api

```
69
70     ...public function logout()
71     ...{
72     ...     Auth::logout();
73     ...     return response()->json([
74     ...         'status' => 'success',
75     ...         'message' => 'Successfully logged out',
76     ...     ]);
77     ...}
78
79     ...public function refresh()
80     ...{
81     ...     return response()->json([
82     ...         'status' => 'success',
83     ...         'message' => 'Token refresh',
84     ...         'data' => Auth::user(),
85     ...         'auth' => [
86     ...             'token' => Auth::refresh(),
87     ...             'type' => 'bearer',
88     ...         ]
89     ...     ]);
90     ...}
91 }
92
```

- Edit file app/Http/Middleware/Authenticate.php

```
AuthController.php A  Authenticate.php M 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
7  class Authenticate extends Middleware
8  {
9      /**
10       * Get the path the user should be redirected to when they are not authenticated.
11       *
12       * @param \Illuminate\Http\Request $request
13       * @return string|null
14       */
15     /**protected function redirectTo($request)
16     /**{
17     /**if (!$request->expectsJson()) {
18     /**return route('login');
19     /**}
20     /**}
21  }
22
```

- Edit file app/Exceptions/Handler.php

```
AuthController.php A  Authenticate.php M  Handler.php M X
app > Exceptions > Handler.php > Handler
41
42     public function render($request, Throwable $exception)
43     {
44         return response()->json([
45             'status' => 'false',
46             'message' => 'Error. ' . $exception->getMessage(),
47             'data' => []
48         ]);
49     }
50 }
51
```

- Edit file routes/api.php

```
AuthController.php A  Authenticate.php M  Handler.php M  api.php M X
routes > api.php > #Function#92deebf8
1  <?php
2
3  use Illuminate\Http\Request;
4  use Illuminate\Support\Facades\Route;
5  use App\Http\Controllers\Api\AuthController;
6  use App\Http\Controllers\Api\ProductController;
7
8  > /** ...
20 Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
21     ...return $request->user();
22 });
23
24 Route::controller(AuthController::class)->group(function () {
25     ...Route::post('login', 'login');
26     ...Route::post('register', 'register');
27     ...Route::post('logout', 'logout');
28     ...Route::post('refresh', 'refresh');
29 });
30
```

- Edit file routes/api.php

```
AuthController.php A  Authenticate.php M  Handler.php M  api.php M X
routes > api.php > #Function#92deebf8
31 //Route::apiResource('/product', App\Http\Controllers\Api\ProductController::class);
32 Route::controller(ProductController::class)->group(function() {
33     ...Route::get('product', 'index');
34     ...Route::get('product/{id}', 'show');
35     ...Route::post('product', 'store');
36     ...Route::put('product/{id}', 'update');
37     ...Route::delete('product/{id}', 'destroy');
38 });
39
40 Route::fallback(function(){
41     ...return response()->json(
42         ...[
43             ...'status'=>false,
44             ...'message'=>'Page Not Found'
45         ],
46         404);
47 });
48
```


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

```
ProductController.php M 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     public function __construct()
14     {
15         $this->middleware('auth:api', ['except' => ['index', 'show']]);
16     }
17
```

Postman Response



Gunakan method POST
Ketikan url "localhost:8000/api/register"

name : Nama user
email : emailuser@gmail.com
password : password

Isi body -> form-data

POST localhost:8000/api/register

Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

form-data

| | Key | Value | Description | ... | Bulk Edit |
|-------------------------------------|----------|---------------------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | name | Gelar Aditya | | | |
| <input checked="" type="checkbox"/> | email | akatgelar@gmail.com | | | |
| <input checked="" type="checkbox"/> | password | password | | | |
| | Key | Value | Description | | |

Body Cookies Headers (10) Test Results

200 OK 325 ms 876 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "success",
3   "message": "User created successfully",
4   "data": {
5     "name": "Gelar Aditya",
6     "email": "akatgelar@gmail.com",
7     "updated_at": "2023-06-06T20:04:40.000000Z",
8     "created_at": "2023-06-06T20:04:40.000000Z",
9     "id": 3
10  },
11  "auth": {
12    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ3c3MiOiJodHRwOi8vbG9jYXRob3N0bGwMDAvYXBpL3JlZ2lzdG9yIiwiaWF0IjoxNjg2MDgxODgwLCJleHAiOiJlZ20yY0U0ODAsIm5iZiI6MTY4NjA4MTg4MCIwianRpIjoicHpwGhycnk0UWdyVTV4NCIsInN1YiI6IjMiLCJwcnYiOiIyM2JkNW40TQ5ZjYwMGFkyjM5ZTcwMWM0MDA4NzJkYjdhNTk3NmY3In0.dsmG9kCq9me_Yc4407Bb-cobVLLvJEgxZkDnMKzntQ",
13    "type": "bearer"
14  }
15 }
```

Postman Response



Gunakan method POST
Ketikan url "localhost:8000/api/register"

Error Response

The image shows the Postman application interface. At the top, a POST request is configured to the URL "localhost:8000/api/register". The "Body" tab is selected, showing a "form-data" collection with three entries: "name" with value "Gelar Aditya", "email" with value "akateglar@gmail.com", and "password" with value "password". The response section on the right shows a 200 OK status with a response time of 209 ms and a body size of 389 B. The response body is displayed in the "Pretty" view as a JSON object: {"status": "false", "message": "Error. The email has already been taken.", "data": []}.

POST localhost:8000/api/register

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

form-data

| | Key | Value | Description | ... | Bulk Edit |
|-------------------------------------|----------|---------------------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | name | Gelar Aditya | | | |
| <input checked="" type="checkbox"/> | email | akateglar@gmail.com | | | |
| <input checked="" type="checkbox"/> | password | password | | | |
| | Key | Value | Description | | |

Body Cookies Headers (10) Test Results

200 OK 209 ms 389 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "false",
3   "message": "Error. The email has already been taken.",
4   "data": []
5 }
```

Postman Response



Gunakan method POST
Ketikan url "localhost:8000/api/login"

email : emailuser@gmail.com
password : password

Isi body -> form-data

POST

localhost:8000/api/login

Send

ParamsAuthHeaders (9)BodyPre-req. Tests SettingsCookies

form-data

| | Key | Value | Description | ... | Bulk Edit |
|-------------------------------------|----------|---------------------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | email | akatgelar@gmail.com | | | |
| <input checked="" type="checkbox"/> | password | password | | | |
| | Key | Value | Description | | |

BodyCookiesHeaders (10)Test Results

200 OK448 ms859 BSave as Example

PrettyRawPreviewVisualizeJSON

```
1  {
2    "status": "success",
3    "data": {
4      "id": 3,
5      "name": "Gelar Aditya",
6      "email": "akatgelar@gmail.com",
7      "email_verified_at": null,
8      "created_at": "2023-06-06T20:04:40.000000Z",
9      "updated_at": "2023-06-06T20:04:40.000000Z"
10   },
11   "auth": {
12     "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbG9jYXRob3N0OjgwMDAvYXBpL2xvZ2luIiwiaWF0IjoxNjg2MDgyMDgwLCJleHAiOiJlZ20yYw0DU200AsIm5iZiI6MTY4NjA4MCwianRpIjoiodUNEY2xiRk50MUUVuOEhUNiIsInN1YiI6IjMlLCJwcnYiOiIyM2JkNW40TQ5ZjYwMGFKYjM5T2cwMWM0MDA4NzJkYjdhNTk3NmY3In0.OlHYzs30o4zE-tAB7FyvUdMgEu-zp5GTAGuIuwJl1P8",
13     "type": "bearer"
14   }
15 }
```

Postman Response



Gunakan method POST
Ketikan url "localhost:8000/api/login"

Error Response

POST localhost:8000/api/login Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

form-data

| | Key | Value | Description | ... | Bulk Edit |
|-------------------------------------|----------|---------------------|-------------|-----|-----------|
| <input checked="" type="checkbox"/> | email | akatgelar@gmail.com | | | |
| <input checked="" type="checkbox"/> | password | passwords | | | |
| | Key | Value | Description | | |

Body Cookies Headers (10) Test Results 401 Unauthorized 236 ms 361 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "error",
3   "message": "Unauthorized"
4 }
```

Postman Response



Gunakan method GET
Ketikan url "localhost:8000/api/product"

Tidak perlu token

GET localhost:8000/api/product

Send

Params Auth Headers (9) Body Pre-req Tests Settings Cookies

Type No Auth

This request does not use any authorization. Learn more about [authorization](#)

Body Cookies Headers (10) Test Results

200 OK 221 ms 830 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": true,
3   "message": "Get Data Successfull",
4   "data": [
5     {
6       "id": 1,
7       "name": "kertas a4",
8       "image": "",
9       "description": "kertas 1 rim",
10      "type": "alat kantor",
11      "price": 40000,
12      "created_at": "2023-05-24T11:27:31.000000Z",
13      "updated_at": "2023-05-24T11:27:33.000000Z",
14      "created_date_indo": "Rabu, 24 Mei 2023 11:27:31"
15    },
16  ]
17 }
```

Postman Response



Gunakan method POST
Ketikan url "localhost:8000/api/product"

Error Response, karena perlu token

The image shows the Postman application interface. On the left, the 'Request' tab is active, showing a POST request to 'localhost:8000/api/product'. The body is a JSON object with fields: name, image, description, type, and price. On the right, the 'Response' tab is active, showing a 200 OK status with a response body of {"status": "false", "message": "Error. Unauthenticated.", "data": []}. The status bar at the top right indicates '200 OK 174 ms 322 B'.

POST localhost:8000/api/product

Params Auth Headers (9) Body Pre-req. Tests Settings

raw JSON Beautify

```
1 {
2   "name": "spidol 2",
3   "image": "http://google2.com",
4   "description": "12 warna 2",
5   "type": "alat kantor 2",
6   "price": 120002
7 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "false",
3   "message": "Error. Unauthenticated.",
4   "data": []
5 }
```

200 OK 174 ms 322 B Save as Example

Postman Response



Gunakan method POST

Ketikan url "localhost:8000/api/product"

Pada Auth -> type

Pilih Bearer Token

Paste token yang didapat dari login

POST localhost:8000/api/product **Send**

Params **Auth** Headers (10) Body Pre-req. Tests Settings **Cookies** Body Cookies Headers (10) Test Results 201 Created 253 ms 597 B Save as Example

Type **Bearer Token**

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)

Token eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1...

Body Pretty Raw Preview Visualize JSON

```
1  {
2    "success": true,
3    "message": "Insert Data Successfull",
4    "data": {
5      "name": "spidol 2",
6      "image": "http://google2.com",
7      "description": "12 warna 2",
8      "type": "alat kantor 2",
9      "price": 120002,
10     "updated_at": "2023-06-06T20:10:56.000000Z",
11     "created_at": "2023-06-06T20:10:56.000000Z",
12     "id": 7
13   },
14   "pagination": []
15 }
```