

Nama : Mukhamad Rico Ramada

Kelas : XIRPL6

Absen : 23

API

```
1  <?php
2
3  use Illuminate\Http\Request;
4
5  /*
6   |-----
7   | API Routes
8   |-----
9   |
10  | Here is where you can register API routes for your application. These
11  | routes are loaded by the RouteServiceProvider within a group which
12  | is assigned the "api" middleware group. Enjoy building your API!
13  |
14  */
15
16  Route::middleware('auth:api')->get('/user', function (Request $request) {
17      return $request->user();
18  });
19  //USER
20  Route::post('register', 'UserController@register');
21  Route::post('login', 'UserController@login');
22  Route::get('book', 'BookController@book');
23
24  Route::get('bookall', 'BookController@bookAuth')->middleware('jwt.verify');
25  Route::get('user', 'UserController@getAuthenticatedUser')->middleware('jwt.verify');
26
27  //ANGGOTA
28  Route::get('/cek_anggota', 'anggotacontroller@cek')->middleware('jwt.verify');
29  Route::post('/save_anggota', 'anggotacontroller@store')->middleware('jwt.verify');
30  Route::put('/ubah_anggota/{id}', 'anggotacontroller@update')->middleware('jwt.verify');
31  Route::delete('/hapus_anggota/{id}', 'anggotacontroller@delete')->middleware('jwt.verify');
32  Route::get('tampil_anggota', 'anggotacontroller@tampil')->middleware('jwt.verify');
33
34  //BUKU
35  Route::get('/cek_buku', 'bukucontroller@cek')->middleware('jwt.verify');
36  Route::post('/save_buku', 'bukucontroller@store')->middleware('jwt.verify');
37  Route::put('/ubah_buku/{id}', 'bukucontroller@update')->middleware('jwt.verify');
38  Route::delete('/hapus_buku/{id}', 'bukucontroller@delete')->middleware('jwt.verify');
39  Route::get('tampil_buku', 'bukucontroller@tampil')->middleware('jwt.verify');
40
```

## Controller Anggota

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\anggota;
7  use Validator;
8  use Auth;
9
10 class anggotacontroller extends Controller
11 {
12     public function cek()
13     {
14         if(Auth::user()->levels=='admin'){
15             $dt_anggota=anggota::get();
16             return response()->json($dt_anggota);
17         }else{
18             return response()->json(['status'=>'anda bukan admin']);
19         }
20     }
21     public function store(Request $req)
22     {
23         if (Auth::user()->levels=='admin') {
24             $valid=Validator::make($req->all(),
25                 [
26                     'nama_anggota'=>'required',
27                     'alamat'=>'required',
28                     'nomer_telepon'=>'required',
29                     'username'=>'required',
30                     'password'=>'required'
31                 ]
32             );
33             if($valid->fails()){
34                 return Response()->json($valid->errors());
35             }
36         }
```

```
36     $save=anggota::create([
37         'nama_anggota'=>$req->nama_anggota,
38         'alamat'=>$req->alamat,
39         'nomer_telepon'=>$req->nomer_telepon,
40         'username'=>$req->username,
41         'password'=>$req->password
42     ]);
43     if($save){
44         return Response()->json(['Berhasil Di Tambahkan']);
45     }else {
46         return Response()->json(['Gagal Di Tambahkan']);
47     }
48 }
49 }
50 public function update($id,Request $req)
51 {
52     if (Auth::user()->levels=='admin') {
53         $valid=Validator::make($req->all(),
54             [
55                 'nama_anggota'=>'required',
56                 'alamat'=>'required',
57                 'nomer_telepon'=>'required',
58                 'username'=>'required',
59                 'password'=>'required'
60             ]
61         );
62         if($valid->fails()){
63             return Response()->json($valid->errors());
64         }
65     }
```

```

65     $ubah=anggota::where('id',$id)->update([
66         'nama_anggota'=>$req->nama_anggota,
67         'alamat'=>$req->alamat,
68         'nomer_telepon'=>$req->nomer_telepon,
69         'username'=>$req->username,
70         'password'=>$req->password
71     ]);
72     if($ubah){
73         return Response()->json(['Berhasil Di Update']);
74     }else {
75         return Response()->json(['Gagal Di Update']);
76     }
77 }
78 }
79 public function delete($id)
80 {
81     if (Auth::user()->level=='admin') {
82         $hapus=anggota::where('id',$id)->delete();
83         if($hapus){
84             return Response()->json(['Telah dihapus']);
85         }else{
86             return Response()->json(['Gagal dihapus']);
87         }
88     }
89 }
90 public function tampil()
91 {
92     if (Auth::user()->level=='admin') {
93         $data=anggota::all();
94         $hitung=anggota::count();
95         return Response()->json(['Jumlah'=>$hitung,['Data'=>$data]);
96     }
97 }
98 }
99

```

## Controler Buku

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\buku;
7  use Validator;
8  use Auth;
9
10 class bukucontroller extends Controller
11 {
12     public function cek()
13     {
14         if(Auth::user()->levels=='admin'){
15             $dt_buku=buku::get();
16             return response()->json($dt_buku);
17         }else{
18             return response()->json(['status'=>'anda bukan admin']);
19         }
20     }
21
22     public function store(Request $req)
23     {
24         if (Auth::user()->levels=='admin') {
25             // code...
26             $valid=Validator::make($req->all(),
27                 [
28                     'judul'=>'required',
29                     'penerbit'=>'required',
30                     'pengarang'=>'required'
31                 ]
32             );
33             if($valid->fails()){
34                 return Response()->json($valid->errors());
35             }
36         }
37     }
38 }
```

```

35     $save=buku::create([
36         'judul'=>$req->judul,
37         'penerbit'=>$req->penerbit,
38         'pengarang'=>$req->pengarang
39     ]);
40     if($save){
41         return Response()->json(['Berhasil Di Tambahkan']);
42     }else {
43         return Response()->json(['Gagal Di Tambahkan']);
44     }
45 }
46 }

public function update($id,Request $req)
{
    if (Auth::user()->levels=='admin') {
        // code...
51     $valid=Validator::make($req->all(),
52         [
53             'judul'=>'required',
54             'penerbit'=>'required',
55             'pengarang'=>'required'
56         ]
57     );
58     if($valid->fails()){
59         return Response()->json($valid->errors());
60     }
}

```

```

60     }
61     $ubah=buku::where('id',$id)->update([
62         'judul'=>$req->judul,
63         'penerbit'=>$req->penerbit,
64         'pengarang'=>$req->pengarang
65     ]);
66     if($ubah){
67         return Response()->json(['Berhasil Di Update']);
68     }else {
69         return Response()->json(['Gagal Di Update']);
70     }
71 }
72 }
73 public function delete($id)
74 {
75     if (Auth::user()->levels=='admin') {
76         // code...
77         $hapus=buku::where('id',$id)->delete();
78         if($hapus){
79             return Response()->json(['Telah dihapus']);
80         }else{
81             return Response()->json(['Gagal dihapus']);
82         }
83     }
84 }
85 public function tampil()
86 {
87     if (Auth::user()->levels=='admin') {
88         // code...
89         $data=buku::all();
90         $hitung=buku::count();
91         return Response()->json(['Jumlah'=>$hitung],['Data'=>$data]);
92     }
93 }
94 }
95

```

## Login

[CONFLICT] POST http://127.0.0.1:80...

POST http://localhost/book/public/...

+...

No Environment

POSThttp://localhost/AnyTemplate/public/api/login

Send

Save

<input checked="" type="checkbox"/>	email	rafiy12@gmail.com	
<input checked="" type="checkbox"/>	password	rafiy123456	
<input type="checkbox"/>	password_confirmation	rafiy123456	
<input type="checkbox"/>	nama_anggota	Rafiy	
<input type="checkbox"/>	alamat	jl durian	
<input type="checkbox"/>	nomer_telepon	098128312736	
<input type="checkbox"/>	username	rafiy@gmail.com	
<input type="checkbox"/>	password	apasaja	
<input type="checkbox"/>	levels	admin	
	Key	Value	Description

BodyCookiesHeaders (11)Test Results

Status: 200 OKTime: 816msSize: 703 B

Save Response

PrettyRawPreviewVisualize BETAJSON

1

2

3

{

"token":

"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOiwwXC9sb2NhOGhvc3RcL0FueVR1bXBsYXR1XC9wdWJsaWNCe2FwaVwvYm9naW4iLCJpYXQiOiJE1ODAyMjA3NDQsImV4cCI6MTU4MDIyNDM0NCwiImIjozNTgwIiwiaWF0Ij0aU9laWJ3bV14OUhaSjVLIiwic3ViIjozLCJwc2V0Ii4N2UwYWVxZWY5ZmQxNTgxImZkZW5NzE1M2ExNGUwYjA0NzU0NmFhIn0.nqqNqCcGRWtK0j2mc9V-1yZvecjfyNAh-UFYhOMpR9k"

}



save anggota

POST

http://localhost/AnyTemplate/public/api/save\_anggota

<input type="checkbox"/>		
<input type="checkbox"/>	email	rico323@gmail.com
<input type="checkbox"/>	password	rico123456
<input type="checkbox"/>	password_confirmation	rico123456
<input checked="" type="checkbox"/>	nama_anggota	Rafly
<input checked="" type="checkbox"/>	alamat	jl durian
<input checked="" type="checkbox"/>	nomer_telepon	098128312736
<input checked="" type="checkbox"/>	username	rafly@gmail.com
<input checked="" type="checkbox"/>	password	apasaja
	Key	Value

Body

Cookies

Headers (11)

Test Results

Pretty

Raw

Preview

Visualize BETA

JSON

```
1  [
2    "Berhasil Di Tambahkan"
3  ]
```

## MODEL

```
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class anggota extends Model
8  {
9      protected $table='table_anggota';
10     protected $primaryKey='id';
11     public $timestamps=false;
12
13     protected $fillable=[
14         'nama_anggota',
15         'alamat',
16         'nomer_telepon',
17         'username',
18         'password'
19     ];
20 }
21
```

```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class buku extends Model
8 {
9     protected $table='table_buku';
10    protected $primaryKey='id';
11    public $timestamps=false;
12
13    protected $fillable=[
14        'judul',
15        'penerbit',
16        'pengarang'
17    ];
18 }
19
```

The screenshot shows the REST Client interface with a POST request to `http://localhost/AnyTemplate/public/api/register`. The request body is a JSON object with the following fields: `alamat`, `nomer_telepon`, `username`, `password`, and `levels` (checked). The response is a JSON object with the following fields: `name`, `email`, `levels`, `updated_at`, `created_at`, `id`, and `token`. The status is 201 Created, and the time taken is 922ms.