

Anggota

Create

Untitled Request Comments (0)

POST ▼ perpus/api/simpan Send Save

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies

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

	KEY	VALUE	DESCRIPTION	...
<input checked="" type="checkbox"/>	nama_anggota	janet		
<input checked="" type="checkbox"/>	alamat	jalan g		
<input checked="" type="checkbox"/>	no_telp	6666		
	Key	Value	Description	

Body Cookies Headers (11) Test Results Status: 200 OK Time: 231ms Size: 370 B Save Response

Pretty Raw Preview Visualize BETA JSON ≡

```
1 {  
2   "status": "berhasil tambah data"  
3 }
```

Update

Untitled Request Comments (0)

POST ▼ perpus/api/update/7 Send Save ▼

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies Code

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

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	nama_anggota	mamak			
<input checked="" type="checkbox"/>	no_telp	099966			
<input checked="" type="checkbox"/>	alamat	jalan jalan			
	Key	Value	Description		

Body Cookies Headers (11) Test Results Status: 200 OK Time: 451ms Size: 370 B Save Response

Pretty Raw Preview Visualize BETA JSON ≡ 📄 🔍

```
1 {  
2   "status": "berhasil update data"  
3 }
```

Delete

POST perpus/api/delete/2

Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

▼ Headers (1)

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3Mi...				
Key	Value	Description			

► Temporary Headers (8) ⓘ

Body Cookies Headers (11) Test Results Status: 200 OK Time: 269ms Size: 358 B Save Response ▼

Pretty Raw Preview Visualize BETA JSON ▼

```
1 {
2   "status": "berhasil"
3 }
```

Read

GET perpus/api/show

Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA

This request does not have a body

Body Cookies Headers (11) Test Results Status: 200 OK Time: 458ms Size: 720 B Save Response ▼

Pretty Raw Preview Visualize BETA JSON ▼

```
1 {
2   {
3     "nama_anggota": "abel",
4     "alamat": "jalan bapakmu",
5     "no_telp": 666666
6   },
7   {
8     "nama_anggota": "xo",
9     "alamat": "jalan oke",
10    "no_telp": 666666
11  },
12  {
13    "nama_anggota": "roko",
14    "alamat": "jalan halu",
15    "no_telp": 999999
16  },
17  {
```

Buku

Create

The screenshot shows the Postman interface for an "Untitled Request". The request method is POST and the URL is `perpus/api/tambah`. The "Body" tab is selected, showing form-data with three fields: `judul` (value: `aku`), `pengarang` (value: `sip`), and `penerbit` (value: `juair`). The response status is 200 OK, with a time of 369ms and a size of 370 B. The response body is displayed in JSON format: `{"status": "berhasil tambah data"}`.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> judul	aku	
<input checked="" type="checkbox"/> pengarang	sip	
<input checked="" type="checkbox"/> penerbit	juair	
Key	Value	Description

```
1 {  
2   "status": "berhasil tambah data"  
3 }
```

Update

The screenshot shows the Postman interface for an "Untitled Request". The request method is POST and the URL is `perpus/api/updatebuku/5`. The "Body" tab is selected, showing form-data with three fields: `judul` (value: `jalu`), `pengarang` (value: `popo`), and `penerbit` (value: `siak`). The response status is 200 OK, with a time of 330ms and a size of 370 B. The response body is displayed in JSON format: `{"status": "berhasil update data"}`.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> judul	jalu	
<input checked="" type="checkbox"/> pengarang	popo	
<input checked="" type="checkbox"/> penerbit	siak	
Key	Value	Description

```
1 {  
2   "status": "berhasil update data"  
3 }
```

Delete

My Workspace

POST ... POST ... POST ... POST ... GET ... POST ... POST ... GET ... POST ...

No Environment

Untitled Request

Comments (0)

POST perpus/api/deletebuku/1

Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

Headers (1)

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	bearer eyj0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3Mi...				
Key	Value	Description			

Temporary Headers (8)

Body Cookies Headers (11) Test Results

Status: 200 OK Time: 414ms Size: 358 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "status": "berhasil"
3 }
```

Read

My Workspace

POST ... POST ... POST ... POST ... GET ... POST ... POST ... GET ... POST ...

No Environment

Untitled Request

Comments (0)

GET perpus/api/showw

Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

Headers (1)

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	bearer eyj0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3Mi...				
Key	Value	Description			

Temporary Headers (7)

Body Cookies Headers (11) Test Results

Status: 200 OK Time: 286ms Size: 669 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   {
3     "judul": "hantu",
4     "pengarang": "juice",
5     "penerbit": "kita"
6   },
7   {
8     "judul": "aku",
9     "pengarang": "kuro",
10    "penerbit": "tebe"
11  },
12 }
```

Script

Model anggota

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class anggota extends Model
{
    protected $table="table_anggota";
    protected $primaryKey="id";
    protected $fillable=['nama_anggota','alamat','no_telp'];
}
```

Anggotacontroller

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Auth;
use App\anggota;
use Illuminate\Support\Facades\Validator;
class anggotacontroller extends Controller
{
    public function aku(Request $req){
        if(Auth::user()->levels=="admin"){
            $validator=Validator::make($req->all(),
            [
                'nama_anggota'=>'required',
                'alamat'=>'required',
                'no_telp'=>'required',
            ]
        );
        if($validator->fails()){
            return Response()->json($validator->errors());
        }
        $simpan=anggota::create([
            'nama_anggota'=>$req->nama_anggota,
```

```

        'alamat'=>$req->alamat,
        'no_telp'=>$req->no_telp,

    ]);
    if($simpan){
        return response ()->json(['status'=>'berhasil tambah data']);
    }
    else{
        return response ()->json(['status'=>'gagal']);
    }
    }
    else{
        return response()-
>json(['status'=>'gagal :(']);
    }
}

public function update($id,Request $req){
    $validator=Validator::make($req->all(),
    [
        'nama_anggota'=>'required',
        'alamat'=>'required',
        'no_telp'=>'required',

    ]

    );
    if($validator->fails()){
        return Response()->json($validator->errors());
    }
    $ubah=anggota::where('id',$id)->update([
        'nama_anggota'=>$req->nama_anggota,
        'alamat'=>$req->alamat,
        'no_telp'=>$req->no_telp,

    ]);
    if($ubah){
        return Response()->json(['status'=>'berhasil update data']);
    }
    else{
        return Response()->json(['status'=>'gagal deh :(']);
    }
}

public function destroy($id){
    $hapus=anggota::where('id',$id)->delete();
    if($hapus){

```

```

        return Response()->json(['status'=>'berhasil']);
    }
    else{
        return Response()->json(['status'=>'gagal']);
    }
}

public function show(){
    $data_anggota = anggota::get();
    $arr_data = array();
    foreach($data_anggota as $data) {
        $arr_data[] = array(
            'nama_anggota'=>$data->nama_anggota,
            'alamat'=>$data->alamat,
            'no_telp'=>$data->no_telp,

        );

    }

    return Response()->json($arr_data);
}
}

```

Model buku

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class buku extends Model
{
    protected $table="table_buku";
    protected $primaryKey="id";
    protected $fillable=['judul','penerbit','pengarang'];
}

```

Bukucontrolller

```

<?php

```

```

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Auth;
use App\buku;
use Illuminate\Support\Facades\Validator;
class bukucontroller extends Controller
{
    public function tambah(Request $req){
        if(Auth::user()->levels=="kasir"){
            $validator=Validator::make($req->all(),
            [
                'judul'=>'required',
                'pengarang'=>'required',
                'penerbit'=>'required',

            ]
        );
        if($validator->fails()){
            return Response()->json($validator->errors());
        }
        $simpan=buku::create([
            'judul'=>$req->judul,
            'pengarang'=>$req->pengarang,
            'penerbit'=>$req->penerbit,

        ]);
        if($simpan){
            return response ()->json(['status'=>'berhasil tambah data']);
        }
        else{
            return response ()->json(['status'=>'gagal']);
        }
        }
        else{
            return response()-
            >json(['status'=>'gagal :(']);
        }
    }
    public function update($id,Request $req){
        $validator=Validator::make($req->all(),
        [
            'judul'=>'required',
            'pengarang'=>'required',
            'penerbit'=>'required',

```



```

    ]
);
if($validator->fails()){
    return Response()->json($validator->errors());
}
$ubah=buku::where('id',$id)->update([
    'judul'=>$req->judul,
    'pengarang'=>$req->pengarang,
    'penerbit'=>$req->penerbit,
]);
if($ubah){
    return Response()->json(['status'=>'berhasil update data']);
}
else{
    return Response()->json(['status'=>'gagal deh :(']);
}
}

public function destroy($id){
    $hapus=buku::where('id',$id)->delete();
    if($hapus){
        return Response()->json(['status'=>'berhasil']);
    }
    else{
        return Response()->json(['status'=>'gogol']);
    }
}

public function show(){
    $data_buku = buku::get();
    $arr_data = array();
    foreach($data_buku as $data) {
        $arr_data[] = array(
            'judul'=>$data->judul,
            'pengarang'=>$data->pengarang,
            'penerbit'=>$data->penerbit,

        );

    }

    return Response()->json($arr_data);
}
}

```

Route Api

Sengaja saya coba untuk edit dan delete memakai post

```
Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});
// anggota
Route::post('register', 'UserController@register');
Route::post('login', 'UserController@login');
Route::post('simpan', 'anggotacontroller@aku')->middleware('jwt.verify');
Route::post('update/{id}', 'anggotacontroller@update')->middleware('jwt.verify');
Route::post('delete/{id}', 'anggotacontroller@destroy')->middleware('jwt.verify');
Route::get('show', 'anggotacontroller@show')->middleware('jwt.verify');
//buku
Route::post('tambah', 'bukucontroller@tambah')->middleware('jwt.verify');
Route::post('updatebuku/{id}', 'bukucontroller@update')->middleware('jwt.verify');
Route::post('deletebuku/{id}', 'bukucontroller@destroy')->
    middleware('jwt.verify');
Route::get('showw', 'bukucontroller@show')->middleware('jwt.verify');
```