



EXPLORER

- API-AIDEL
  - config
    - JS database.js
    - node\_modules
  - routes
    - JS posts.js
- api\_musik\_aidel.sql
- JS index.js
- package-lock.json
- package.json

OUTLINE

TIMELINE

```

JS posts.js
routes > JS posts.js > router.get('/:id') callback
1  const express = require('express')
2  const router = express.Router()
3
4  // import database
5  const koneksi = require('../config/database')
6
7  // insert data & validasi
8  const {body, validationResult} =require('express-validator')
9
10
11 // membaca data
12 router.get('/', function(req,res){
13     koneksi.query('SELECT * FROM posts ORDER BY id desc',
14     function(error,rows){
15         if(error){
16             return res.status(500).json({
17                 status: false,
18                 message: 'database ngga nyambung',
19             })
20         }else{
21             return res.status(200).json({
22                 status:true,
23                 message: 'Menampilkan data table posts',
24                 data:rows
25             })
26         }
27     })
28 })
29
30 // insert data
31 router.post('/store',
32 [

```

package-lock.json

package.json

Ln 92, Col 55 Spaces: 4 UTF-8 CRLF {} JavaScript

EXPLORER

- API-AIDEL
  - config
    - JS database.js
    - node\_modules
  - routes
    - JS posts.js
- api\_musik\_aidel.sql
- JS index.js
- package-lock.json
- package.json

OUTLINE

TIMELINE

```

routes > JS posts.js > router.get('/id') callback
30 // insert data
31 router.post('/store',
32 [
33   body('id').notEmpty(),
34   body('band').notEmpty(),
35   body('tahun_debut').notEmpty(),
36   body('member').notEmpty(),
37   body('vokalis').notEmpty(),
38   body('gitaris').notEmpty(),
39   body('bassist').notEmpty(),
40   body('pianist').notEmpty(),
41   body('drummer').notEmpty(),
42   body('trombone').notEmpty(),
43   body('genre').notEmpty(),
44   body('album').notEmpty(),
45   body('judul_lagu').notEmpty()
46 ],(req,res)=>{
47   const errors = validationResult(req)
48   if(!errors.isEmpty()){
49     return res.status(422).json({errors:errors.array()})
50   }
51
52   // mendefinisikan formData
53   let formData = {
54     id: req.body.id,
55     band: req.body.band,
56     tahun_debut: req.body.tahun_debut,
57     member: req.body.member,
58     vokalis: req.body.vokalis,
59     gitaris: req.body.gitaris,
60     bassist: req.body.bassist,
61     pianist: req.body.pianist,

```





EXPLORER

- API-AIDEL
  - config
    - database.js
    - node\_modules
  - routes
    - posts.js
- api\_musik\_aidel.sql
- index.js
- package-lock.json
- package.json

OUTLINE

TIMELINE

```

JS posts.js x JS database.js JS index.js {} package-lock.json {} package.json
routes > JS posts.js > router.patch('/update/:id') callback > formData
89 router.get('/:id', function(req,res){
91
92     koneksi.query(`SELECT * FROM posts WHERE ID=${id}`,
93     function(error, rows){
94         if(error){
95             return res.status(500).json({
96                 status: false,
97                 message: 'Server Error'
98             })
99         }
100
101         // pencarian posts
102         if(rows.length <= 0){
103             return res.status(404).json({
104                 status: false,
105                 message: 'Data tidak ada'
106             })
107         } else {
108             return res.status(200).json({
109                 status: true,
110                 message: 'menampilkan data posts',
111                 data: rows[0],
112             })
113         }
114     })
115 }
116
117 })
118
119 // update
120 router.patch('/update/:id',[
121     // validasi

```

Run and Debug console showing logs for the API endpoint.

```

[11/12/2024, 18:32:30] GET /:id
[11/12/2024, 18:32:30] {
  status: true,
  message: 'menampilkan data posts',
  data: {
    id: 1,
    title: 'Post 1',
    content: 'Content 1'
  }
}
[11/12/2024, 18:32:30] GET /:id
[11/12/2024, 18:32:30] {
  status: false,
  message: 'Data tidak ada'
}
[11/12/2024, 18:32:30] GET /:id
[11/12/2024, 18:32:30] {
  status: false,
  message: 'Server Error'
}

```



File

Edit

Selection

View

Go

Run

...

←

→

api-aidel

🔍

📄

🔗

⚙️

📦

82°F

Cloudy

🔍

Search

🏠

📁

📺

📞

📧

🎵

🔊

🔌

🔋

11:16

16/12/2024

EXPLOLER

...

JS posts.js

JS database.js

JS index.js

package-lock.json

package.json X

API-AIDEL

config

database.js

node\_modules

routes

posts.js

api\_musik\_aidel.sql

index.js

package-lock.json

package.json

package.json > ...

1 {

2 "name": "api\_musik\_aidel",

3 "version": "1.0.0",

4 "description": "",

5 "main": "index.js",

6 "scripts": {

7 "start": "node index.js",

8 "dev": "nodemon index.js"

9 },

10 "author": "aidel",

11 "license": "ISC",

12 "dependencies": {

13 "api\_musik\_aidel": "file:",

14 "body-parser": "^1.20.3",

15 "confiq": "^0.0.3",

16 "database": "^0.0.2",

17 "database.js": "^0.0.1",

18 "express": "^4.21.2",

19 "express-validator": "^7.2.0",

20 "mysql": "^2.18.1"

21 },

22 "devDependencies": {

23 "nodemon": "^3.1.9"

24 }

25 }

26 }

Ln 5, Col 24

Spaces: 4

UTF-8

CRLF

{ } JSON





```
JS posts.js  X  JS database.js  JS index.js  {} package-lock.json  {} package.json
routes > JS posts.js > router.get('/:id') callback
89  router.get('/:id', function(req,res){
93      function(error, rows){
112          })
113      }
114  }
115  )
116
117  })
118
119  // update
120  router.patch('/update/:id',[
121      // validasi
122      body('id').notEmpty(),
123      body('band').notEmpty(),
124      body('tahun_debut').notEmpty(),
125      body('member').notEmpty(),
126      body('vokalis').notEmpty(),
127      body('gitaris').notEmpty(),
128      body('bassist').notEmpty(),
129      body('pianist').notEmpty(),
130      body('drummer').notEmpty(),
131      body('trombone').notEmpty(),
132      body('genre').notEmpty(),
133      body('album').notEmpty(),
134      body('judul_lagu').notEmpty()
135  ],(req,res)=>{
136      const errors = validationResult (req)
137      if(!errors.isEmpty()){
138          return res.status(442).json({
139              errors:errors.array()
140          })
141      }
```

EXPLORER

- API-AIDEL
  - config
    - JS database.js
    - node\_modules
  - routes
    - JS posts.js
- api\_musik\_aidel.sql
- JS index.js
- package-lock.json
- package.json

OUTLINE

TIMELINE

JS posts.js

```

routes > JS posts.js > router.get('/:id') callback
135 ],(req,res)=>{
142
143 // id
144 let id = req.params.id
145
146 // data posts
147 let formData={
148   id: req.body.id,
149   band: req.body.band,
150   tahun_debut: req.body.tahun_debut,
151   member: req.body.member,
152   vokalis: req.body.vokalis,
153   gitaris: req.body.gitaris,
154   bassist: req.body.bassist,
155   pianist: req.body.pianist,
156   drummer: req.body.drummer,
157   trombone: req.body.trombone,
158   genre: req.body.genre,
159   album: req.body.album,
160   judul_lagu: req.body.judul_lagu,
161 }
162
163 // update query
164 koneksi.query('UPDATE posts set ? WHERE id=${id}'),
165   formData,function(error,rows){
166     if(error){
167       return res.status(500).json({
168         status: false,
169         message: 'server error',
170       })
171     } else {
172       return res.status(200).json({

```

JS database.js

```

const mysql = require('mysql');
const config = require('../config');

const db = mysql.createPool({
  host: config.host,
  user: config.user,
  password: config.password,
  database: config.database,
});

module.exports = db;

```

node\_modules

```

const express = require('express');
const router = express.Router();

const db = require('../database');

router.get('/:id', (req, res) => {
  const id = req.params.id;

  db.query('SELECT * FROM posts WHERE id = ?', [id], (err, rows) => {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error',
      });
    } else if (rows.length > 0) {
      return res.json(rows[0]);
    } else {
      return res.status(404).json({
        status: false,
        message: 'not found',
      });
    }
  });
});

module.exports = router;

```



EXPLORER

- API-AIDEL
  - config
    - JS database.js
    - node\_modules
    - routes
      - JS posts.js
      - api\_musik\_aidel.sql
      - JS index.js
      - package-lock.json
      - package.json

OUTLINE

TIMELINE

```

routes > JS posts.js > router.patch('/update/:id') callback > formData
179      })
180
181      // delete
182      router.delete('/delete/:id',
183        function(req,res){
184          let id = req.paramd.id
185
186          koneksi.query('DELETE FROM posts WHERE id=${id}') ,
187          function(error,rows){
188            if(error) {
189              return res.status(500).json({
190                status: false,
191                message: 'Server error'
192              })
193            } else {
194              return res.status(200).json({
195                status: true,
196                message: 'data sudah dihapus'
197              })
198            }
199          }
200        })
201      module.exports = router
    
```



You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.

New

Import

GET http://localhost:3000/

GET http://localhost:3000/

GET http://localhost:3000/

GET http://localhost:3000/

GET http://localhost:3000/

POST http://localhost:3000/

POST http://localhost:3000/

http://localhost:3000/api/posts/store

Save

POST



http://localhost:3000/api/posts/store

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

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

member

Fauzan Lubis, Rian Rahman aka Wihi Astanto, Hendar Dimas Anggara, Amoruso R...



vokalis

Fauzan Lubis



gitaris



bassist



pianist



drummer

Body

Cookies

Headers (7)

Test Results



Status: 201 Created

Time: 17 ms

Size: 287 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



1

2

3

4









































"status": true,

"message": "Berhasil input data"












Table structure

Relation view

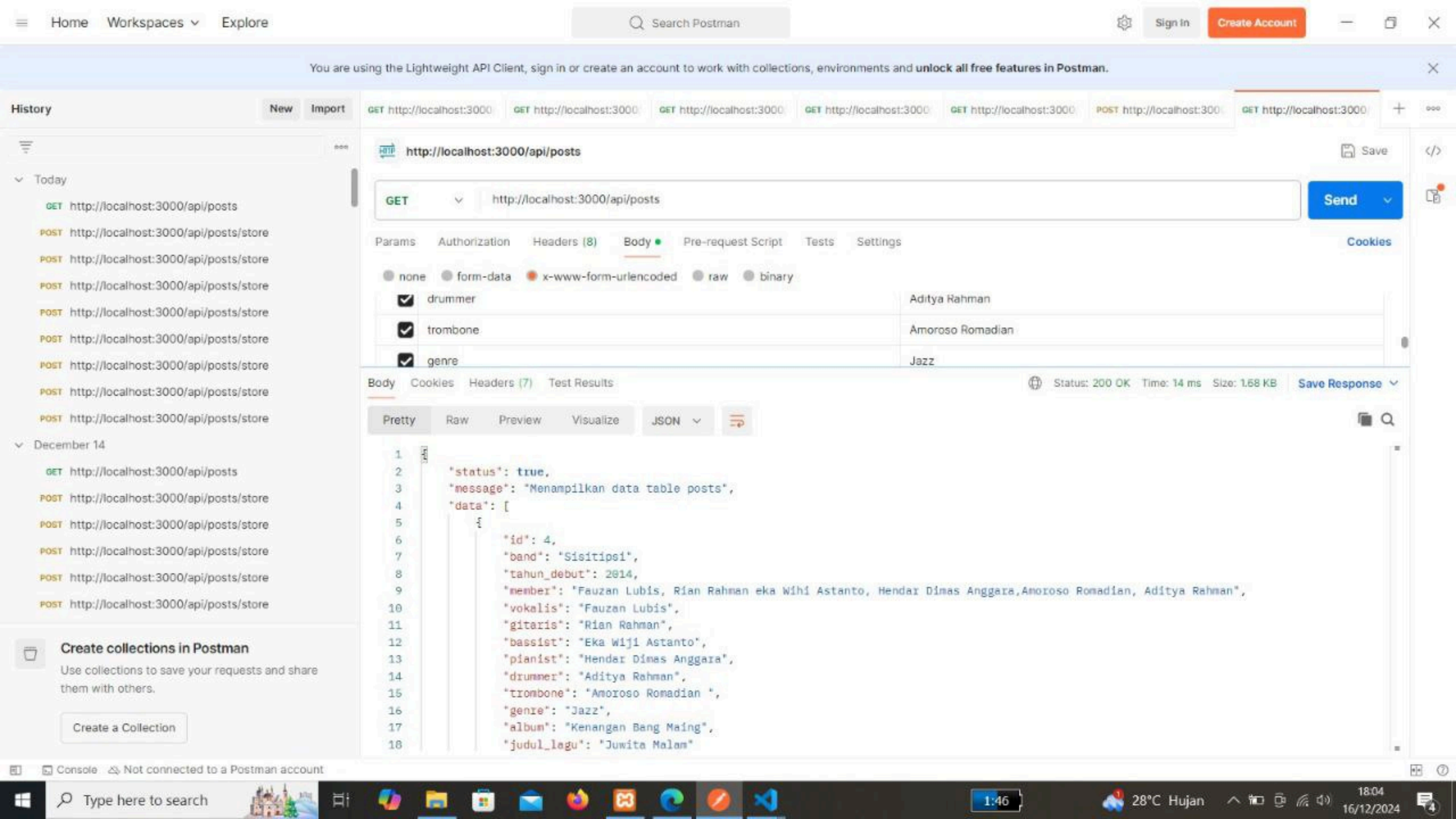
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 <b>id</b> 	int(11)			No	None		AUTO_INCREMENT	 Change  Drop  More
<input type="checkbox"/>	2 <b>band</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	3 <b>tahun_debut</b>	int(11)			Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	4 <b>member</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	5 <b>vokalis</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	6 <b>gitaris</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	7 <b>bassist</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	8 <b>pianist</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	9 <b>drummer</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	10 <b>trombone</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	11 <b>genre</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	12 <b>album</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	13 <b>judul_lagu</b>	varchar(100)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More

☐ Check all

With selected:

 Browse Change Drop Primary Unique Index Spatial Fulltext Add to center Remove from central columns





You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.

History

New Import

GET http://localhost:3000 GET http://localhost:3000 GET http://localhost:3000 GET http://localhost:3000 GET http://localhost:3000 POST http://localhost:3000 GET http://localhost:3000

Today

- GET http://localhost:3000/api/posts
- POST http://localhost:3000/api/posts/store
- POST http://localhost:3000/api/posts/store
- POST http://localhost:3000/api/posts/store
- POST http://localhost:3000/api/posts/store
- POST http://localhost:3000/api/posts/store
- POST http://localhost:3000/api/posts/store
- POST http://localhost:3000/api/posts/store
- POST http://localhost:3000/api/posts/store

December 14

- GET http://localhost:3000/api/posts
- POST http://localhost:3000/api/posts/store
- POST http://localhost:3000/api/posts/store
- POST http://localhost:3000/api/posts/store
- POST http://localhost:3000/api/posts/store
- POST http://localhost:3000/api/posts/store

Create collections in Postman  
Use collections to save your requests and share them with others.

Create a Collection

http://localhost:3000/api/posts

GET http://localhost:3000/api/posts

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

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

<input checked="" type="checkbox"/>	drummer	Aditya Rahman
<input checked="" type="checkbox"/>	trombone	Amoroso Romadian
<input checked="" type="checkbox"/>	genre	Jazz

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 14 ms Size: 1.68 KB Save Response

Pretty Raw Preview Visualize JSON

```
1
2  {
3    "status": true,
4    "message": "Menampilkan data table posts",
5    "data": [
6      {
7        "id": 4,
8        "band": "Sisitipsi",
9        "tahun_debut": 2014,
10       "member": "Fauzan Lubis, Rian Rahman eka Wihi Astanto, Hendar Dimas Anggara, Amoroso Romadian, Aditya Rahman",
11       "vokalis": "Fauzan Lubis",
12       "gitaris": "Rian Rahman",
13       "bassist": "Eka Wihi Astanto",
14       "pianist": "Hendar Dimas Anggara",
15       "drummer": "Aditya Rahman",
16       "trombone": "Amoroso Romadian",
17       "genre": "Jazz",
18       "album": "Kenangan Bang Maing",
19       "judul_lagu": "Juwita Malam"
20     }
21   ]
22 }
```