

BE NodeJS

Tugas: Service Pattern

Leonardho R. Sitanggang

Soal :

Buatlah pola struktur kode file dan folder yang sama untuk modul Categories
Simpan seluruh latihan ini dalam satu folder

1. Kemudian, masukan folder tersebut ke [repository GitHub](#)
2. Boleh simpan folder latihan ini di dalam repository latihan sebelumnya
3. Nama folder-nya adalah **tugas-service-pattern**
4. Kumpulkan link repository nya saja

Jawab :

Source Code Service

```
tugas-service-pattern > src > services > TS categories.services.ts > ...
1  import CategoriesModel, { Category } from "../models/categories.model";
2
3  export const create = async (payload: Category): Promise<Category> => {
4      const result = await CategoriesModel.create(payload);
5      return result;
6  };
7
8  export interface IFindAll {
9      query?: unknown;
10     limit: number;
11     page: number;
12 }
13
14 export const findAll = async (
15     query: any,
16     limit: number = 10,
17     page: number = 1
18 ): Promise<Category[]> => {
19     const result = await CategoriesModel.find(query)
20         .limit(limit)
21         .skip((page - 1) * limit)
22         .sort({ createdAt: -1 });
23     return result;
24 };
25
26 export const findOne = async (id: string): Promise<Category | null> => {
27     const result = await CategoriesModel.findById(id);
28     return result;
29 };
30
31 export const update = async (
32     id: string,
33     payload: Category
34 ): Promise<Category | null> => {
35     const result = await CategoriesModel.findOneAndUpdate({ _id: id }, payload, {
36         new: true,
37     });
38     return result;
39 };
40
41 export const remove = async (id: string): Promise<Category | null> => {
42     const result = await CategoriesModel.findOneAndDelete({
43         _id: id,
44     });
45     return result;
46 };
47
```

Source Code Controller

```

2
3 import {
4   create,
5   findAll,
6   findOne,
7   update,
8   remove,
9 } from "../services/categories.services"
10 import { IPaginationQuery } from "../utils/interfaces";
11
12 export default {
13   async create(req: Request, res: Response) {
14     try {
15       const result = await create(req.body);
16       res.status(201).json({
17         data: result,
18         message: "Success create category",
19       });
20     } catch (error) {
21       const err = error as Error;
22       res.status(500).json({
23         data: err.message,
24         message: "Failed create category",
25       });
26     }
27   },
28   async findAll(req: Request, res: Response) {
29     try {
30       const {
31         limit = 10,
32         page = 1,
33         search,
34       } = req.query as unknown as IPaginationQuery;
35       const query = {}
36
37       if(search){
38         Object.assign(query, {
39           name: { $regex: search, $options: "i" }
40         })
41       }
42
43       const result = await findAll(query, limit, page);
44       res.status(200).json({
45         data: result,
46         message: "Success get all categories",
47       });
48     } catch (error) {
49       const err = error as Error;
50       res.status(500).json({
51         data: err.message,
52         message: "Failed get all categories",
53       });
54     }
55   },
56   async findOne(req: Request, res: Response) {
57     try {
58       const result = await findOne(
59         req.params?.id,

```

```

56     async findOne(req: Request, res: Response) {
57         try {
58             const result = await findOne(
59                 req.params?.id,
60             );
61             res.status(200).json({
62                 data: result,
63                 message: "Success get one category",
64             });
65         } catch (error) {
66             const err = error as Error;
67             res.status(500).json({
68                 data: err.message,
69                 message: "Failed get one category",
70             });
71         }
72     },
73     async update(req: Request, res: Response) {
74         try {
75             const result = await update(req.params?.id, req.body);
76
77             res.status(200).json({
78                 data: result,
79                 message: "Success update category",
80             });
81         } catch (error) {
82             const err = error as Error;
83             res.status(500).json({
84                 data: err.message,
85                 message: "Failed update category",
86             });
87         }
88     },
89     async delete(req: Request, res: Response) {
90         try {
91             const result = await remove(req.params?.id);
92
93             res.status(200).json({
94                 data: result,
95                 message: "Success delete category",
96             });
97         } catch (error) {
98             const err = error as Error;
99             res.status(500).json({
100                 data: err.message,
101                 message: "Failed delete category",
102             });
103         }
104     },
105 };

```

Output

GET

▼

http://localhost:3000/api/categories?page=1&limit=1&search=Fresh

Params

Authorization

Headers (6)

Body

Scripts

Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	page	1
<input checked="" type="checkbox"/>	limit	1
<input checked="" type="checkbox"/>	search	Fresh
	Key	Value

Body

Cookies

Headers (7)

Test Results

↺

Pretty

Raw

Preview

Visualize

JSON ▼

⌵

1

2

3

4

5

6

7

8

9

10

11

12

{

"data": [

{

"_id": "6734b0b8f1041fe9e6a3090f",

"name": "Fresh",

"createdAt": "2024-11-13T13:59:20.701Z",

"updatedAt": "2024-11-13T13:59:20.701Z",

"__v": 0

}

],

"message": "Success get all categories"

}

tugas-service-pattern - Run results

Ran today at 21:00:29 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	530ms	0	57 ms

All Tests Passed (0) Failed (0) Skipped (0)

Iteration 1

POST Category

http://localhost:3000/api/categories

201 Created

No tests found

GET Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

PUT Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

DELETE Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

GET All Categories

http://localhost:3000/api/categories

200 OK

No tests found

1

POST tugas-service-pattern / Categories / Category

Response

Headers

Request

Pretty

1

{

2

"data": {

3

"name": "Rubber",

4

"_id": "6734b0dfd1041fe9e6a30913",

5

"createdAt": "2024-11-13T14:00:29.414Z",

6

"updatedAt": "2024-11-13T14:00:29.414Z",

7

"__v": 0

8

},

9

"message": "Success create category"

10

}

tugas-service-pattern - Run results

Ran today at 21:00:29 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	530ms	0	57 ms

All Tests Passed (0) Failed (0) Skipped (0)

Iteration 1

POST Category

http://localhost:3000/api/categories

201 Created

No tests found

GET Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

PUT Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

DELETE Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

GET All Categories

http://localhost:3000/api/categories

200 OK

No tests found

1

GET tugas-service-pattern / Categories / Category By Id

Response

Headers

Request

Pretty

1

{

2

"data": {

3

"_id": "6734b0bef1041fe9e6a30911",

4

"name": "Fresh",

5

"createdAt": "2024-11-13T13:59:26.018Z",

6

"updatedAt": "2024-11-13T13:59:26.018Z",

7

"__v": 0

8

},

9

"message": "Success get one category"

10

}

tugas-service-pattern - Run results

Ran today at 21:00:29

[View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	530ms	0	57 ms

All Tests

Passed (0)

Failed (0)

Skipped (0)

Iteration 1

1

PUT

tugas-service-pattern / Categories / Category By Id

Response

Headers

Request

Pretty

1

{

2

"data": {

3

"_id": "6734b0bef1041fe9e6a30911",

4

"name": "Plastic",

5

"createdAt": "2024-11-13T13:59:26.018Z",

6

"updatedAt": "2024-11-13T14:00:29.530Z",

7

"_v": 0

8

},

9

"message": "Success update category"

10

}

POST

Category

http://localhost:3000/api/categories

201 Created

No tests found

GET

Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

PUT

Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

DELETE

Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

GET

All Categories

http://localhost:3000/api/categories

200 OK

No tests found

tugas-service-pattern - Run results

Ran today at 21:00:29

[View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	530ms	0	57 ms

All Tests

Passed (0)

Failed (0)

Skipped (0)

Iteration 1

1

DELETE

tugas-service-pattern / Categories / Category By Id

Response

Headers

Request

Pretty

1

{

2

"data": {

3

"_id": "6734b0bef1041fe9e6a30911",

4

"name": "Plastic",

5

"createdAt": "2024-11-13T13:59:26.018Z",

6

"updatedAt": "2024-11-13T14:00:29.530Z",

7

"_v": 0

8

},

9

"message": "Success delete category"

10

}

POST

Category

http://localhost:3000/api/categories

201 Created

No tests found

GET

Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

PUT

Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

DELETE

Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

GET

All Categories

http://localhost:3000/api/categories

200 OK

No tests found

tugas-service-pattern - Run results
Run

Ran today at 21:00:29 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	530ms	0	57 ms

All Tests
Passed (0)
Failed (0)
Skipped (0)

Iteration 1

POST Category

http://localhost:3000/api/categories

201 Created

No tests found

GET Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

PUT Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

DELETE Category By Id

http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911

200 OK

No tests found

GET All Categories

http://localhost:3000/api/categories

200 OK

No tests found

1
GET
tugas-service-pattern / Categories / All Categories

Response
Headers
Request

Pretty

```

1  {
2    "data": [
3      {
4        "_id": "6734b0fdf1041fe9e6a30913",
5        "name": "Rubber",
6        "createdAt": "2024-11-13T14:00:29.414Z",
7        "updatedAt": "2024-11-13T14:00:29.414Z",
8        "__v": 0
9      },
10     {
11       "_id": "6734b0b8f1041fe9e6a3090f",
12       "name": "Fresh",
13       "createdAt": "2024-11-13T13:59:20.701Z",
14       "updatedAt": "2024-11-13T13:59:20.701Z",
15       "__v": 0
16     },
17     {
18       "_id": "6734b0b5f1041fe9e6a3090d",
19       "name": "Rubber",
20       "createdAt": "2024-11-13T13:59:17.365Z",
21       "updatedAt": "2024-11-13T13:59:17.365Z",
22       "__v": 0
23     }
24   ],
25   "message": "Success get all categories"
26 }

```

Repository :

- Github <https://github.com/FlazeFy/sanbercode-nodejs-xpress/tree/main/tugas-service-pattern>
- Postman Documentation <https://documenter.getpostman.com/view/18882129/2sAY55ZHm1>

Refrensi :

- Materi Service Pattern https://docs.google.com/document/d/1wR2tuWPOfQG7-IXhAQ5X7zp6JOJlwcTiFgzfP6e_sGM/edit?tab=t.0