BE NodeJS

Tugas: Service Pattern

Leonardho R. Sitanggang

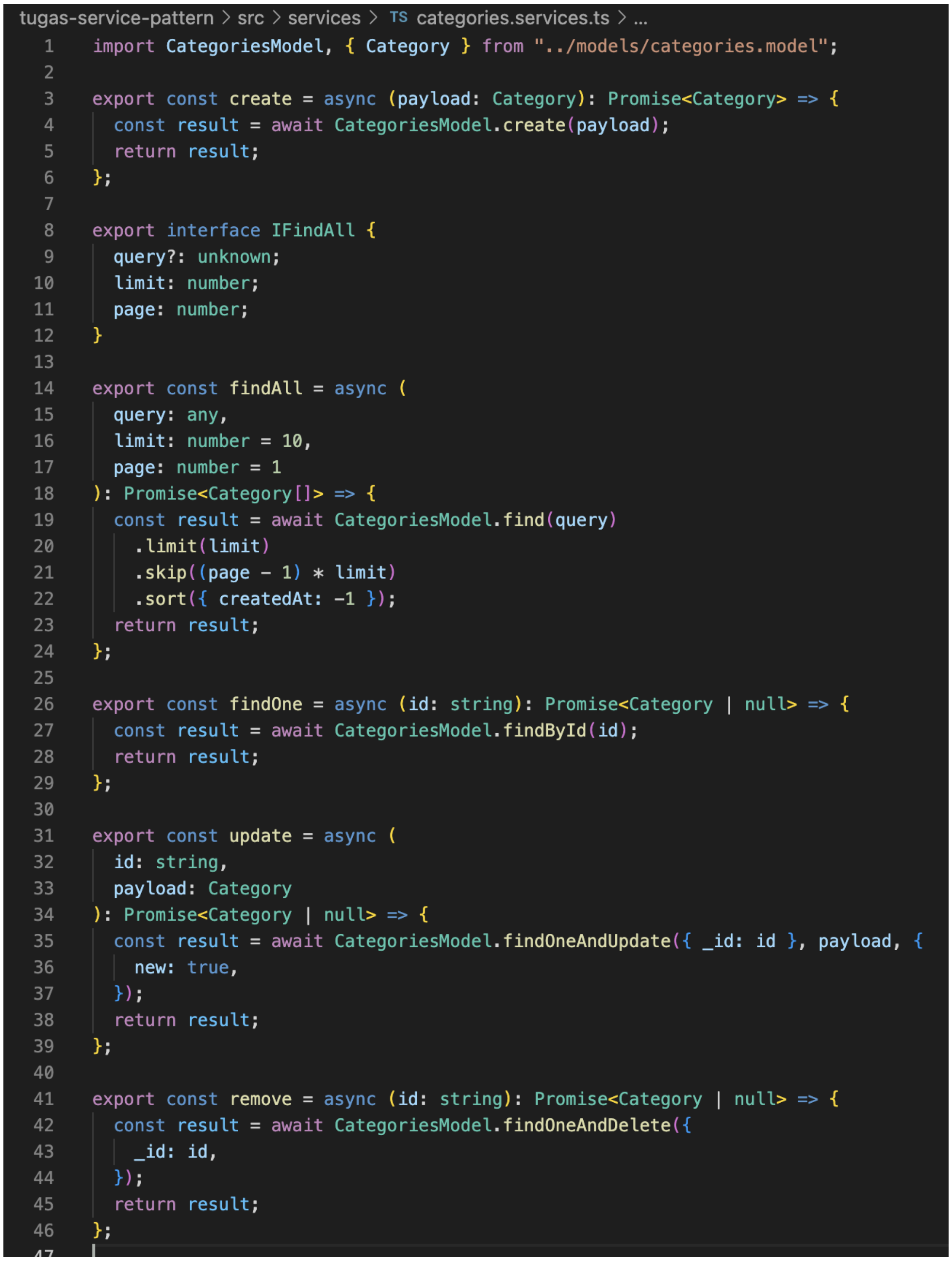
Soal :

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 :

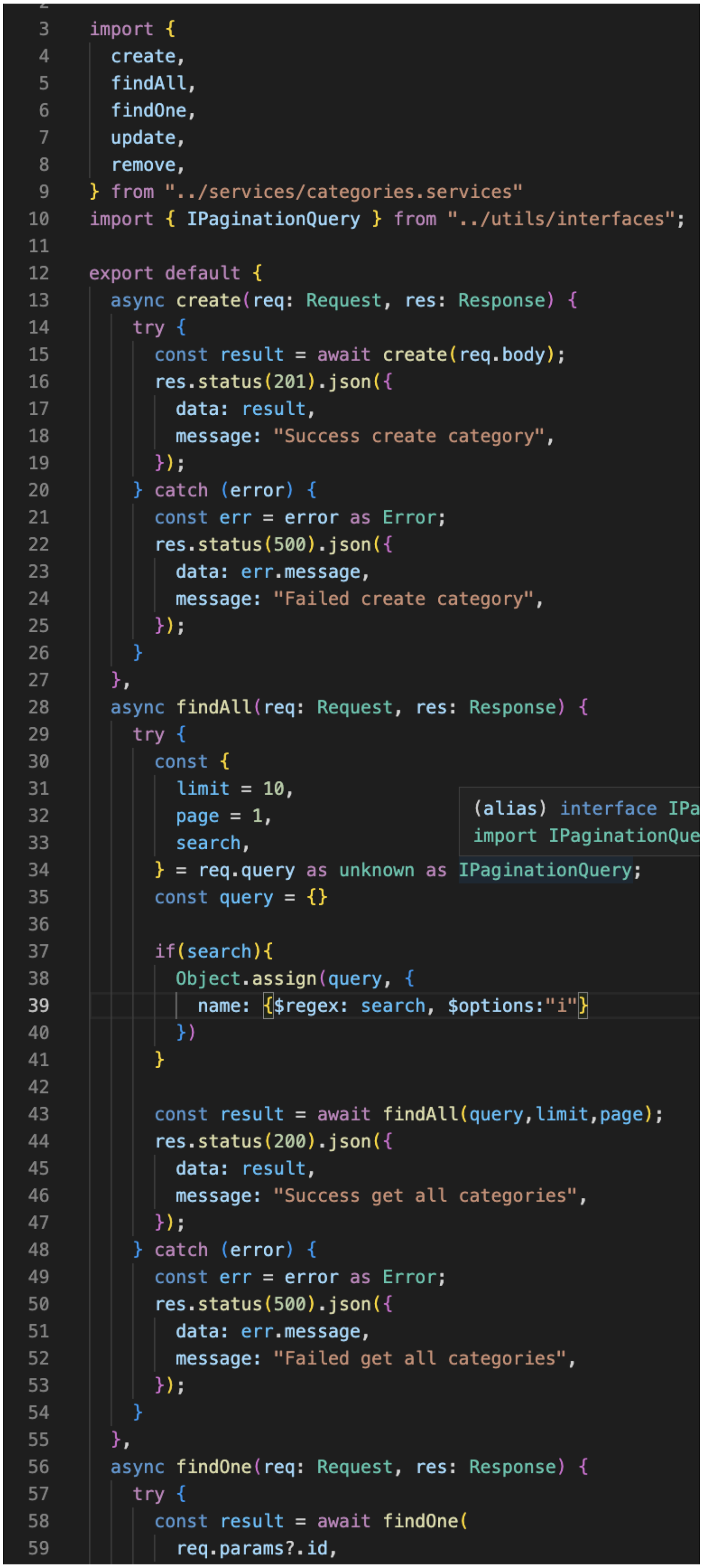
Source Code Service



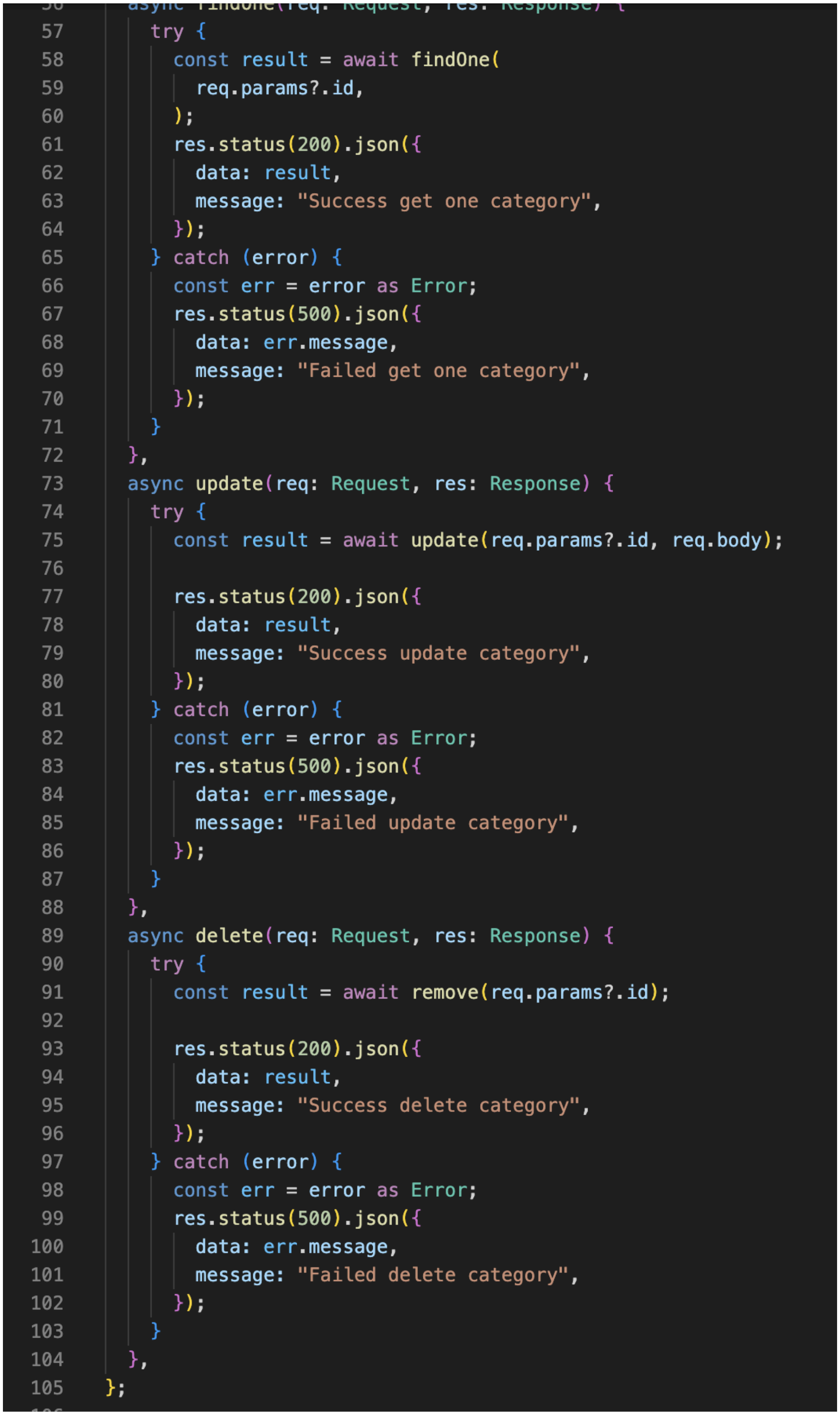
tugas-service-pattern > src > services > TS categories.services.ts > ...

```typescript
import CategoriesModel, { Category } from "../models/categories.model";

export const create = async (payload: Category): Promise<Category> => {
  const result = await CategoriesModel.create(payload);
  return result;
};

export interface IFindAll {
  query?: unknown;
  limit: number;
  page: number;
}

export const findAll = async (
  query: any,
  limit: number = 10,
  page: number = 1
): Promise<Category[]> => {
  const result = await CategoriesModel.find(query)
    .limit(limit)
    .skip((page - 1) * limit)
    .sort({ createdAt: -1 });
  return result;
};

export const findOne = async (id: string): Promise<Category | null> => {
  const result = await CategoriesModel.findById(id);
  return result;
};

export const update = async (
  id: string,
  payload: Category
): Promise<Category | null> => {
  const result = await CategoriesModel.findOneAndUpdate({ _id: id }, payload, {
    new: true,
  });
  return result;
};

export const remove = async (id: string): Promise<Category | null> => {
  const result = await CategoriesModel.findOneAndDelete({
    _id: id,
  });
  return result;
};
```

Source Code Controller

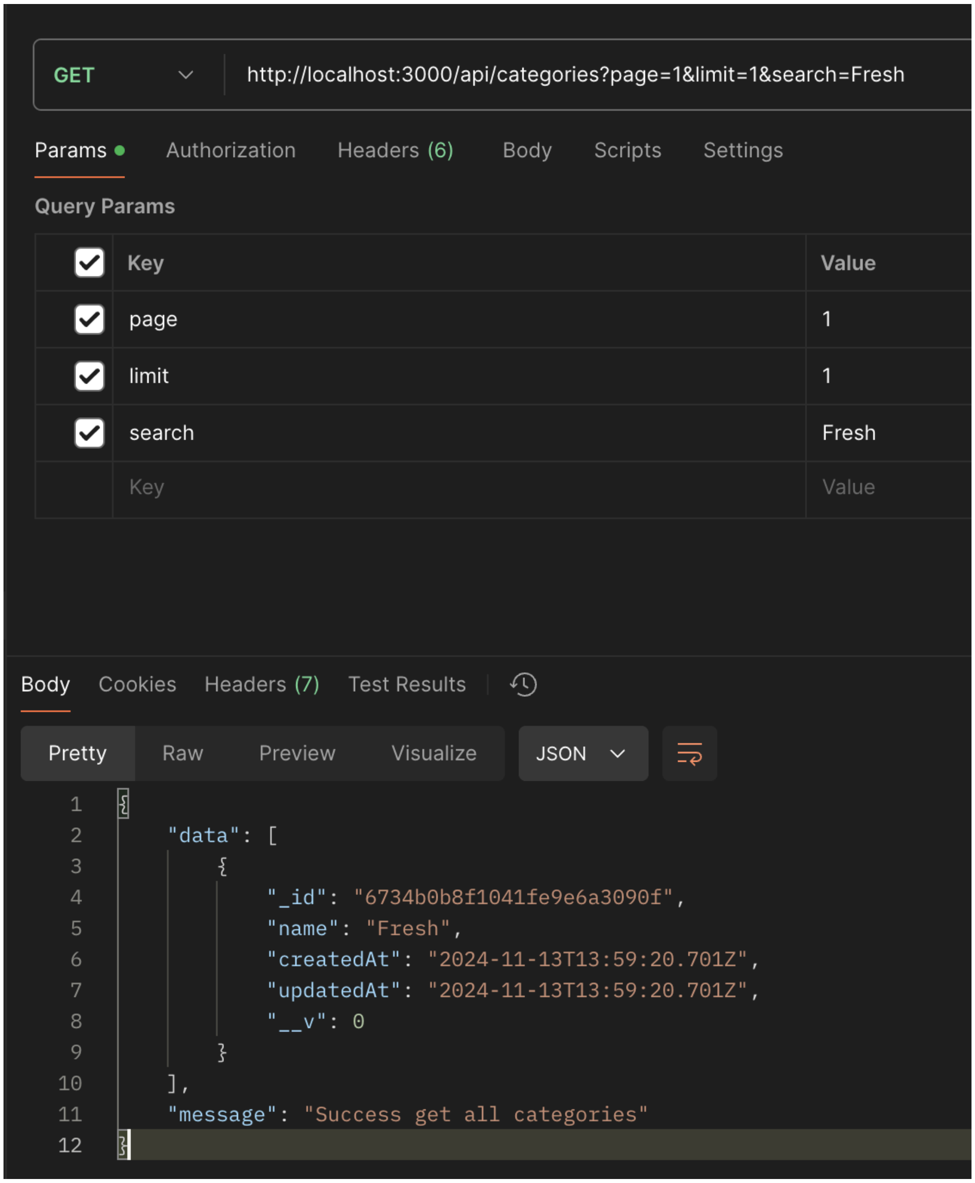


```
import {
  create,
  findAll,
  findOne,
  update,
  remove,
} from "../services/categories.services"
import { IPaginationQuery } from "../utils/interfaces";

export default {
  async create(req: Request, res: Response) {
    try {
      const result = await create(req.body);
      res.status(201).json({
        data: result,
        message: "Success create category",
      });
    } catch (error) {
      const err = error as Error;
      res.status(500).json({
        data: err.message,
        message: "Failed create category",
      });
    }
  },
  async findAll(req: Request, res: Response) {
    try {
      const {
        limit = 10,
        page = 1,
        search,
      } = req.query as unknown as IPaginationQuery;
      const query = {}

      if(search){
        Object.assign(query, {
          name: {$regex: search, $options:"i"}
        })
      }

      const result = await findAll(query,limit,page);
      res.status(200).json({
        data: result,
        message: "Success get all categories",
      });
    } catch (error) {
      const err = error as Error;
      res.status(500).json({
        data: err.message,
        message: "Failed get all categories",
      });
    }
  },
  async findOne(req: Request, res: Response) {
    try {
      const result = await findOne(
        req.params?.id,
```



```
  async findOne(req: Request, res: Response) {
    try {
      const result = await findOne(
        req.params?.id,
      );
      res.status(200).json({
        data: result,
        message: "Success get one category",
      });
    } catch (error) {
      const err = error as Error;
      res.status(500).json({
        data: err.message,
        message: "Failed get one category",
      });
    }
  },
  async update(req: Request, res: Response) {
    try {
      const result = await update(req.params?.id, req.body);

      res.status(200).json({
        data: result,
        message: "Success update category",
      });
    } catch (error) {
      const err = error as Error;
      res.status(500).json({
        data: err.message,
        message: "Failed update category",
      });
    }
  },
  async delete(req: Request, res: Response) {
    try {
      const result = await remove(req.params?.id);

      res.status(200).json({
        data: result,
        message: "Success delete category",
      });
    } catch (error) {
      const err = error as Error;
      res.status(500).json({
        data: err.message,
        message: "Failed delete category",
      });
    }
  },
};
```

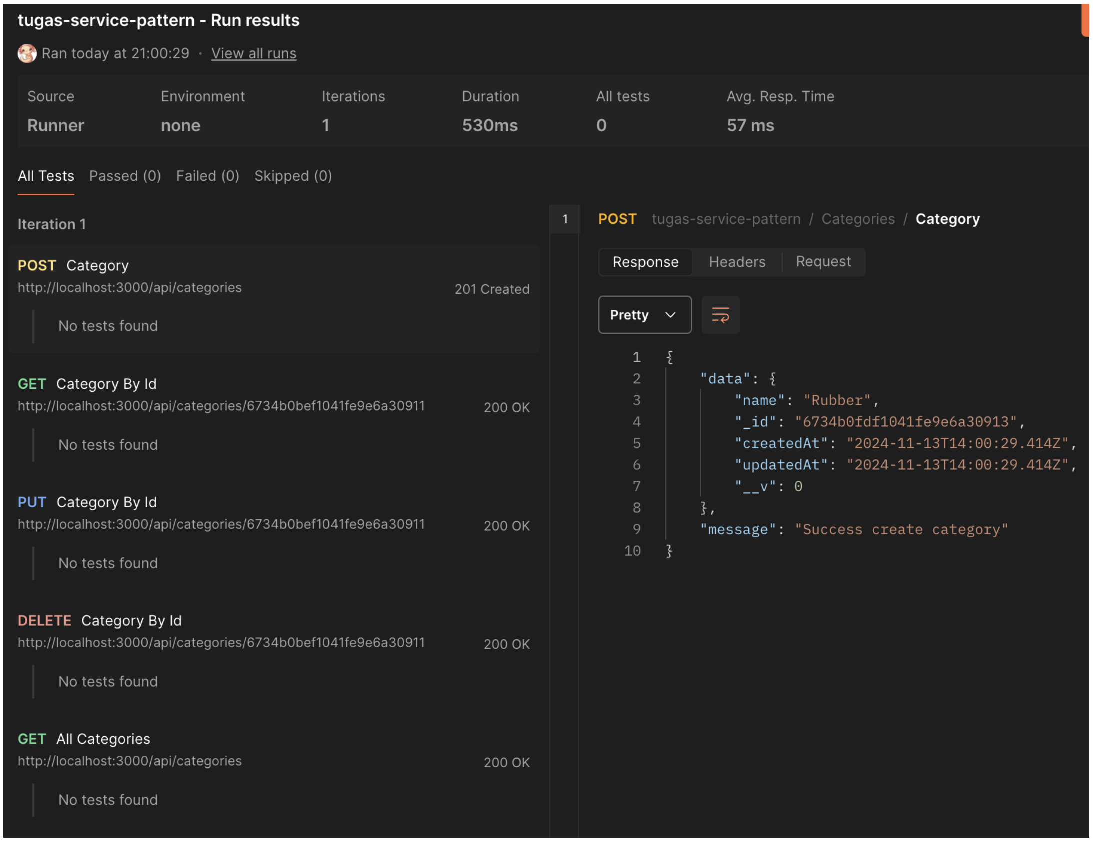
Output



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

☑	Key	Value
☑	page	1
☑	limit	1
☑	search	Fresh
	Key	Value

```
{
    "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

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

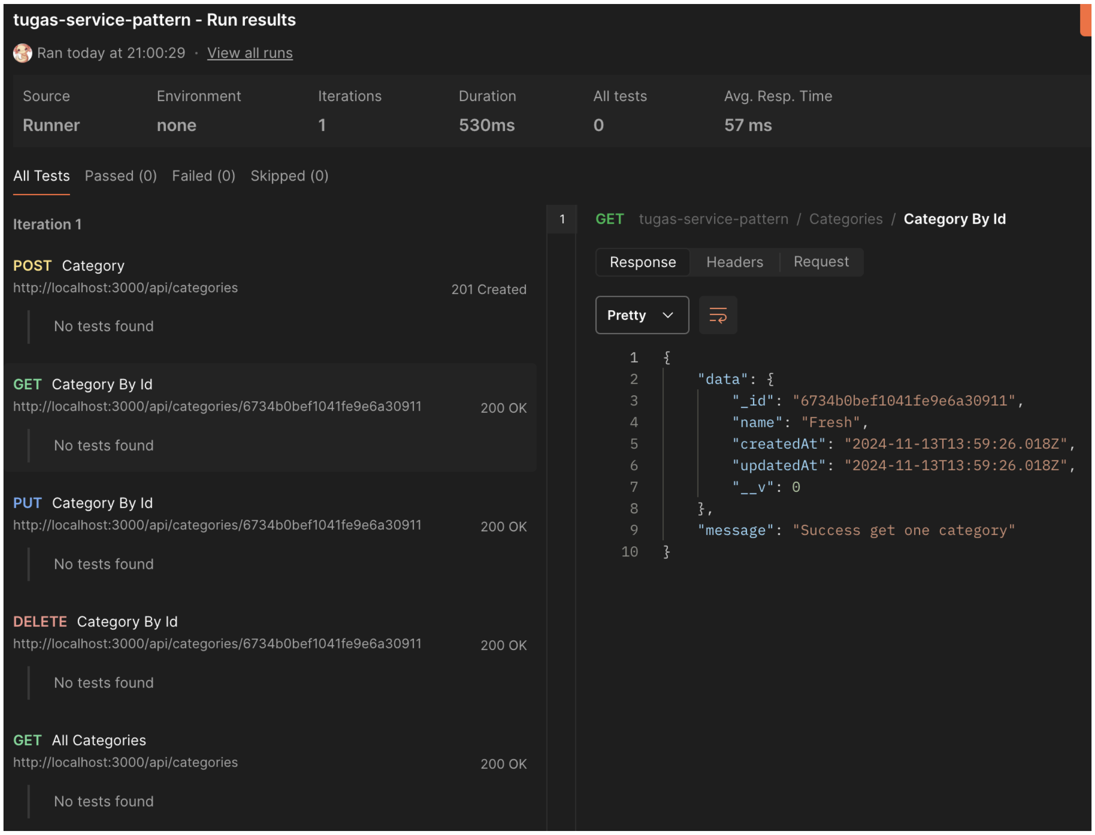
PUT Category By Id
http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911 200 OK
No tests found

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**

```
{
    "data": {
        "name": "Rubber",
        "_id": "6734b0fdf1041fe9e6a30913",
        "createdAt": "2024-11-13T14:00:29.414Z",
        "updatedAt": "2024-11-13T14:00:29.414Z",
        "__v": 0
    },
    "message": "Success create category"
}
```



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

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

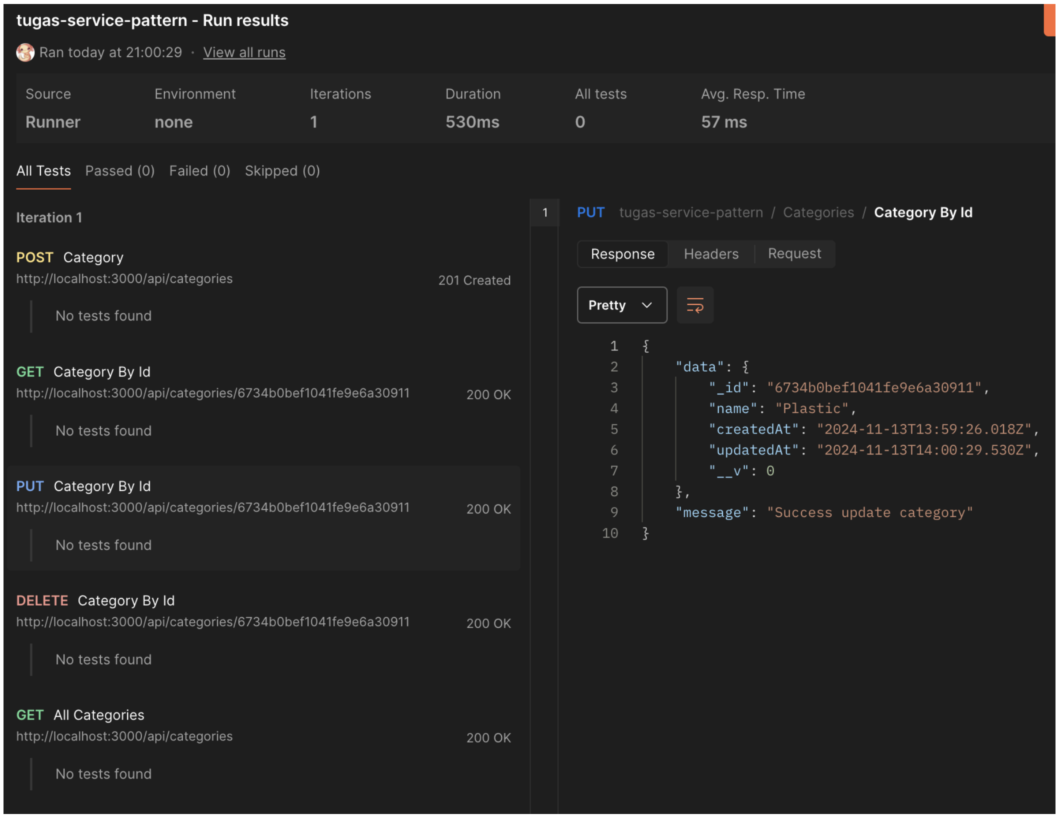
PUT Category By Id
http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911 200 OK
No tests found

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**

```
{
    "data": {
        "_id": "6734b0bef1041fe9e6a30911",
        "name": "Fresh",
        "createdAt": "2024-11-13T13:59:26.018Z",
        "updatedAt": "2024-11-13T13:59:26.018Z",
        "__v": 0
    },
    "message": "Success get one category"
}
```



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

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

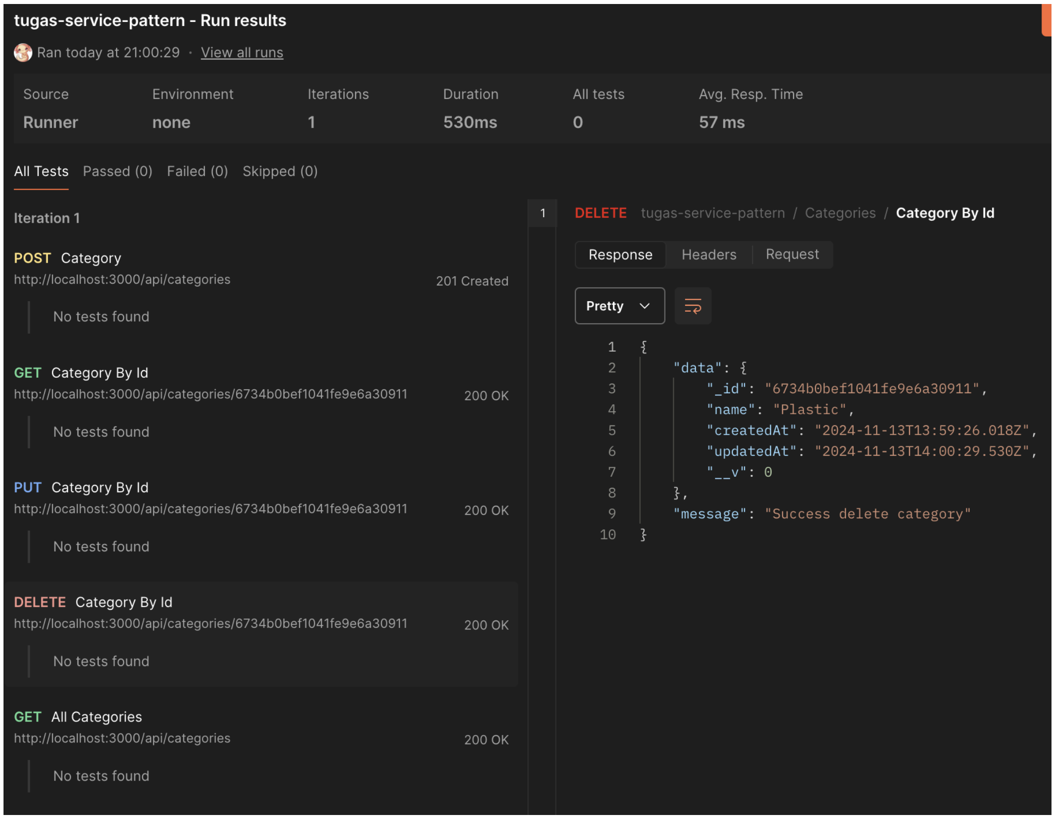
PUT Category By Id
http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911 200 OK
No tests found

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

PUT tugas-service-pattern / Categories / **Category By Id**

```
{
    "data": {
        "_id": "6734b0bef1041fe9e6a30911",
        "name": "Plastic",
        "createdAt": "2024-11-13T13:59:26.018Z",
        "updatedAt": "2024-11-13T14:00:29.530Z",
        "__v": 0
    },
    "message": "Success update category"
}
```



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

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

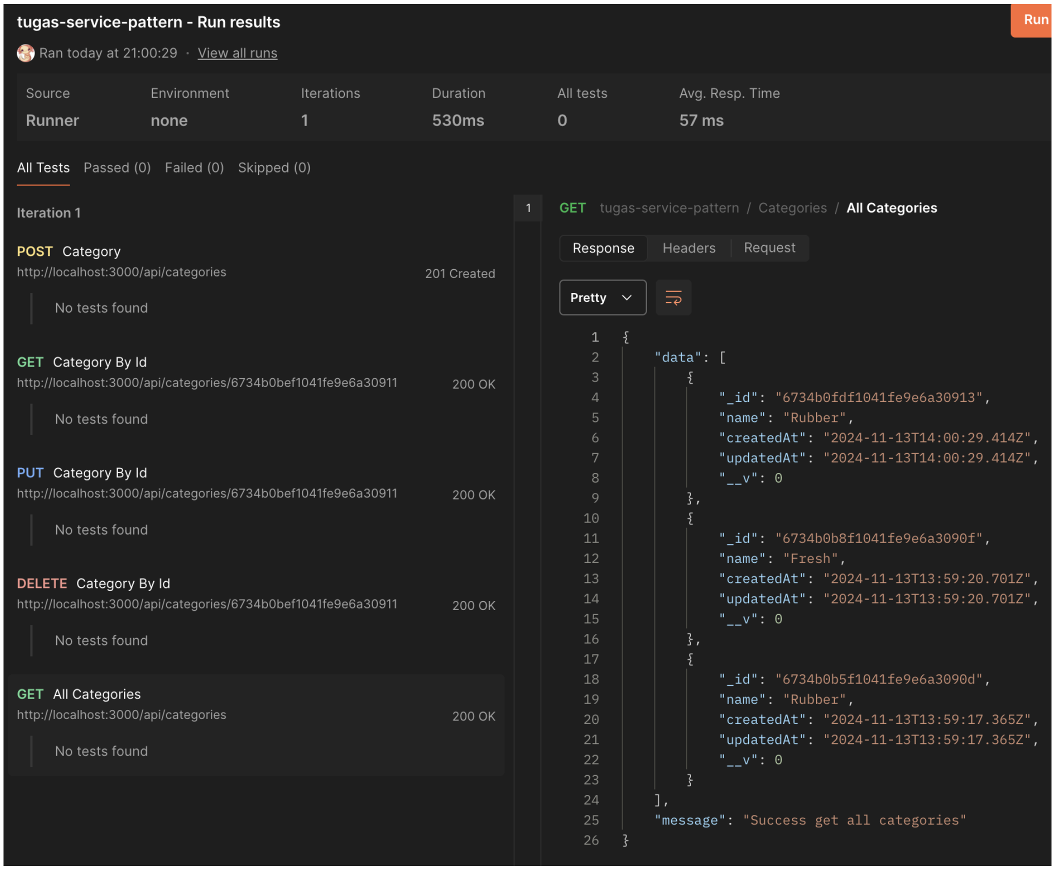
PUT Category By Id
http://localhost:3000/api/categories/6734b0bef1041fe9e6a30911 200 OK
No tests found

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

DELETE tugas-service-pattern / Categories / **Category By Id**

```
{
    "data": {
        "_id": "6734b0bef1041fe9e6a30911",
        "name": "Plastic",
        "createdAt": "2024-11-13T13:59:26.018Z",
        "updatedAt": "2024-11-13T14:00:29.530Z",
        "__v": 0
    },
    "message": "Success delete category"
}
```



tugas-service-pattern - Run results

Ran today at 21:00:29 · View all runs

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

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

```
{
    "data": [
        {
            "_id": "6734b0fdf1041fe9e6a30913",
            "name": "Rubber",
            "createdAt": "2024-11-13T14:00:29.414Z",
            "updatedAt": "2024-11-13T14:00:29.414Z",
            "__v": 0
        },
        {
            "_id": "6734b0b8f1041fe9e6a3090f",
            "name": "Fresh",
            "createdAt": "2024-11-13T13:59:20.701Z",
            "updatedAt": "2024-11-13T13:59:20.701Z",
            "__v": 0
        },
        {
            "_id": "6734b0b5f1041fe9e6a3090d",
            "name": "Rubber",
            "createdAt": "2024-11-13T13:59:17.365Z",
            "updatedAt": "2024-11-13T13:59:17.365Z",
            "__v": 0
        }
    ],
    "message": "Success get all categories"
}
```

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-lXhAQ5X7zp6JOJlwcTiFgzfP6e_sGM/edit?tab=t.0*