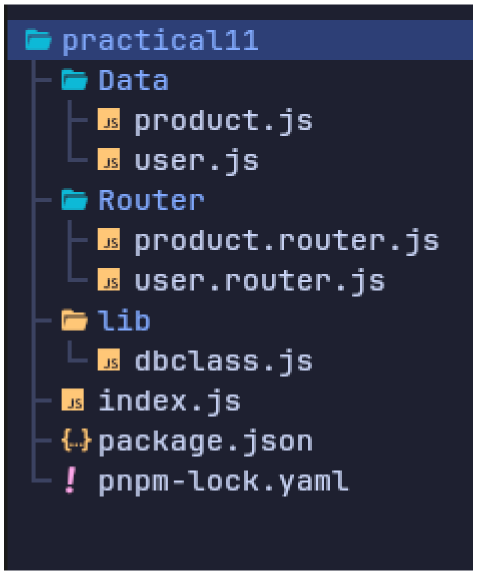
Practical No 11

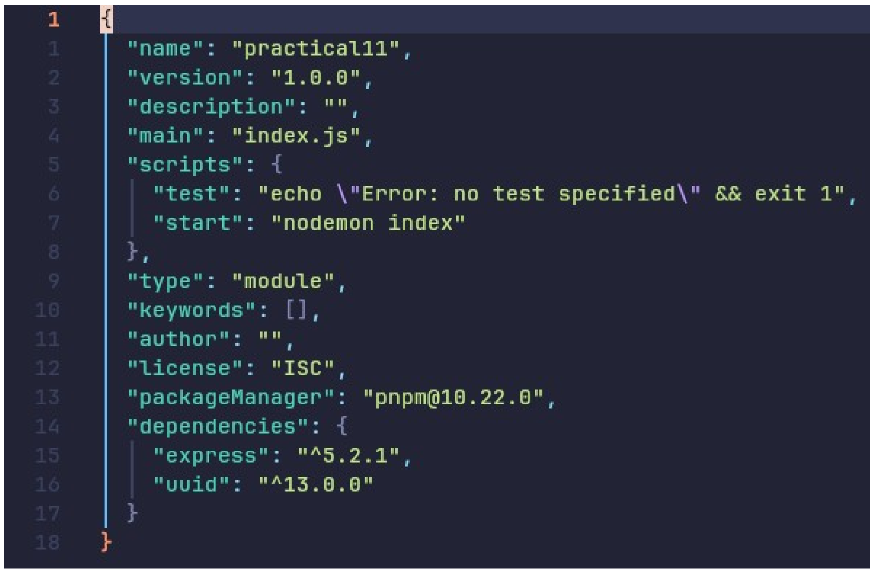
Aim : Write A Program To Demonstrate Organizing Express Application Using Express Router.

Project Structure



```
practical11
├ Data
│ ├ product.js
│ └ user.js
├ Router
│ ├ product.router.js
│ └ user.router.js
├ lib
│ └ dbclass.js
├ index.js
├ package.json
└ pnpm-lock.yaml
```

Package.json



```
{
  "name": "practical11",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "nodemon index"
  },
  "type": "module",
  "keywords": [],
  "author": "",
  "license": "ISC",
  "packageManager": "pnpm@10.22.0",
  "dependencies": {
    "express": "^5.2.1",
    "uuid": "^13.0.0"
  }
}
```

dbclass.js

```
import { v4 as uuidv4 } from "uuid"

export class Collection {

  constructor() {
    this.datas = []
  }

  insert(data) {
    const newData = {
      _id: uuidv4(),
      ...data
    }
    this.datas.push(newData)
    return newData
  }

  findMany() {
    return this.datas
  }

  findById(id) {
    return this.datas.filter((e) => e._id === id)[0]
  }


delete(id) {
    this.datas = this.datas.filter((e) => e._id !== id)
    return true
  }

  update(id, data) {
    this.delete(id)
    const newData = this.insert(data)
    return newData
  }
}
```

product.js

```
import { Collection } from "../lib/dbclass.js";
export const Product = new Collection;
```

user.js

```
import { Collection } from "../lib/dbclass.js";
export const User = new Collection();
```

product.router.js

```
import express from "express"
import { Product } from "../Data/product.js"

export const productRouter = express.Router()

// Get All Product
productRouter.get("/all", (req, res) => {
  const products = Product.findMany()
  res.status(200).json(products)
})

// Get Single Product By id
productRouter.get("/id/:id", (req, res) => {
  const id = req.params.id
  const data = Product.findById(id)
  res.status(200).json(data)
})

// Create Product
productRouter.post("/create", (req, res) => {
  const product = req.body
  if (!product) return res.status(400).json({ message: "Invalid Data." })
  const data = Product.insert(product)
  res.status(201).json(data)
})

// Update Product
productRouter.put("/update/:id", (req, res) => {
  const product = req.body
  const id = req.params.id
  const data = Product.update(id, product)
  res.status(201).json(data)
})


// Delete Product
```

```
productRouter.delete("/id/:id", (req, res) => {
  const id = req.params.id
  const data = Product.delete(id)
  if (data) return res.status(200).json({ message: "Product Deleted Successfully." })
  res.status(400).json({ error: "Error Deleting Product." })
})
```

user.router.js

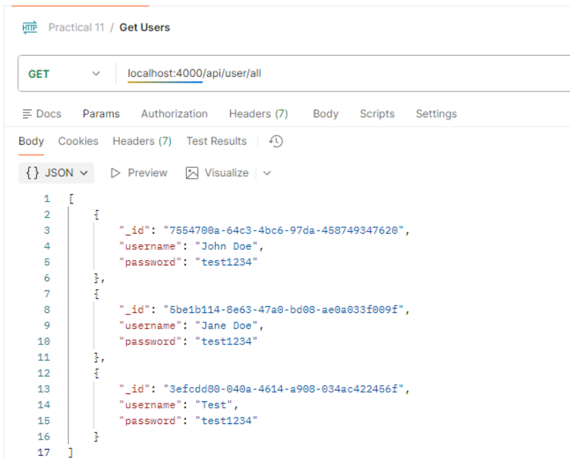
```
import express from "express"
import { User } from "../Data/user.js"

export const userRouter = express.Router()

// Get All User
userRouter.get("/all", (req, res) => {
  const users = User.findMany()
  res.status(200).json(users)
})

// Get Single User By id
userRouter.get("/id/:id", (req, res) => {
  const id = req.params.id
  const data = User.findById(id)
  res.status(200).json(data)
})

// Create User
userRouter.post("/create", (req, res) => {
  const user = req.body
  if (!user) return res.status(400).json({ message: "Invalid Data." })
  const data = User.insert(user)
  res.status(201).json(data)
})

// Update User
userRouter.put("/update/:id", (req, res) => {
  const user = req.body
  const id = req.params.id
  const data = User.update(id, user)
  res.status(201).json(data)
})

// Delete User
userRouter.delete("/id/:id", (req, res) => {
  const id = req.params.id
  const data = User.delete(id)
  if (data) return res.status(200).json({ message: "User Deleted Successfully." })
  res.status(400).json({ error: "Error Deleting User." })
})
```

index.js

```
import express from "express"; import { userRouter } from "./Router/user.router.js";
import { productRouter } from "./Router/product.router.js";

const PORT = 4000;

const app = express();

app.use(express.json())

app.use("/api/user", userRouter)
app.use("/api/product", productRouter)

app.listen(PORT, () => {
  console.log("Server Started.")
})
```

User Router

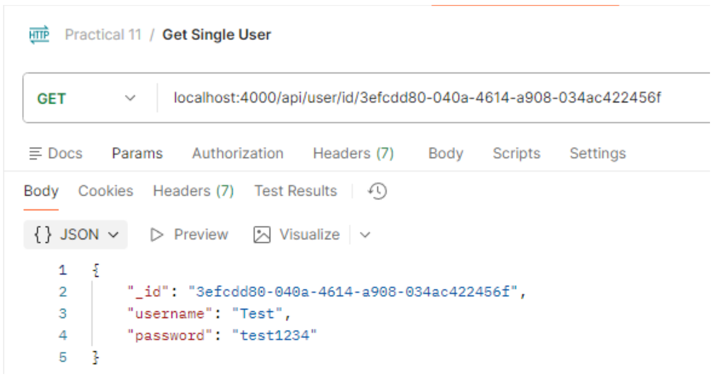
Get All User



```
GET localhost:4000/api/user/all
```

```
[
    {
        "_id": "7554700a-64c3-4bc6-97da-458749347620",
        "username": "John Doe",
        "password": "test1234"
    },
    {
        "_id": "5be1b114-8e63-47a0-bd08-ae0a033f009f",
        "username": "Jane Doe",
        "password": "test1234"
    },
    {
        "_id": "3efcdd80-040a-4614-a908-034ac422456f",
        "username": "Test",
        "password": "test1234"
    }
]
```

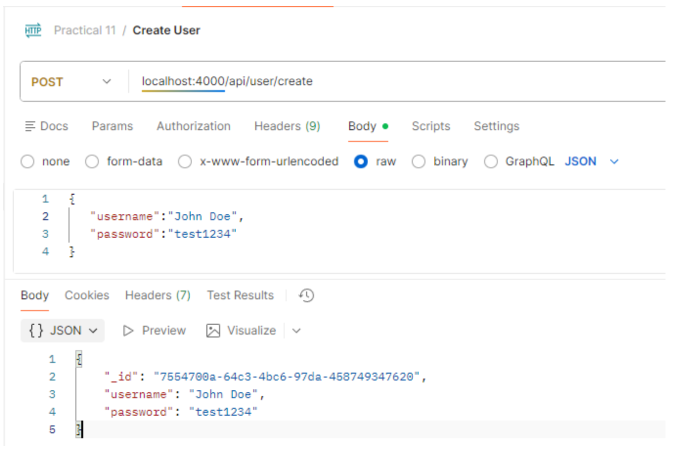
Get Single User



```
GET localhost:4000/api/user/id/3efcdd80-040a-4614-a908-034ac422456f
```

```
{
    "_id": "3efcdd80-040a-4614-a908-034ac422456f",
    "username": "Test",
    "password": "test1234"
}
```

Create User

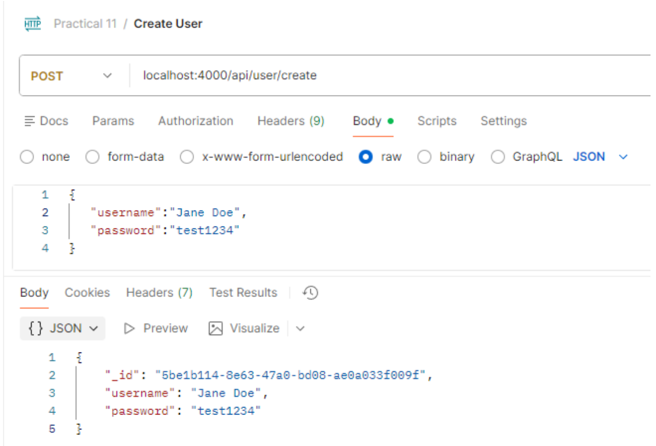


```
POST localhost:4000/api/user/create
```

Body: raw JSON

```
{
    "username":"John Doe",
    "password":"test1234"
}
```

```
{
    "_id": "7554700a-64c3-4bc6-97da-458749347620",
    "username": "John Doe",
    "password": "test1234"
}
```



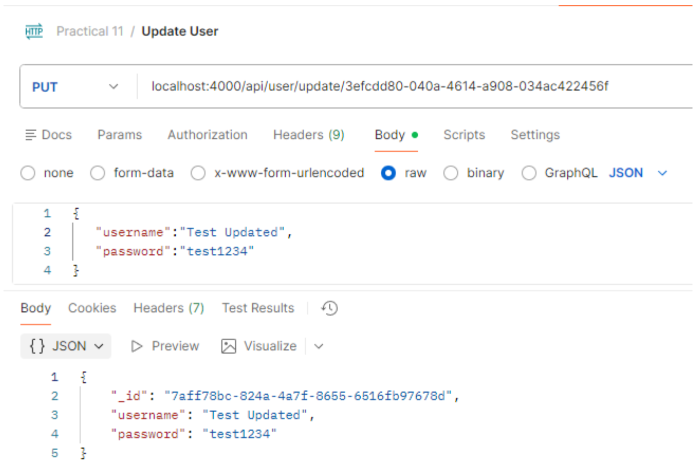
```
POST localhost:4000/api/user/create
```

Body: raw JSON

```
{
    "username":"Jane Doe",
    "password":"test1234"
}
```

```
{
    "_id": "5be1b114-8e63-47a0-bd08-ae0a033f009f",
    "username": "Jane Doe",
    "password": "test1234"
}
```

Update User



Practical 11 / Update User

PUT localhost:4000/api/user/update/3efcdd80-040a-4614-a908-034ac422456f

Docs | Params | Authorization | Headers (9) | Body ● | Scripts | Settings

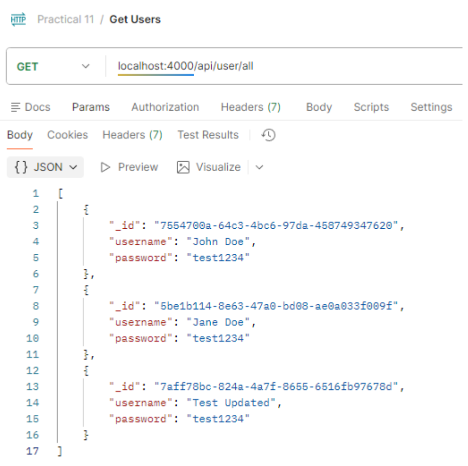
none | form-data | x-www-form-urlencoded | raw [✓] | binary | GraphQL | JSON

```
{
    "username":"Test Updated",
    "password":"test1234"
}
```

Body | Cookies | Headers (7) | Test Results

{} JSON | Preview | Visualize

```
{
    "_id": "7aff78bc-824a-4a7f-8655-6516fb97678d",
    "username": "Test Updated",
    "password": "test1234"
}
```



Practical 11 / Get Users

GET localhost:4000/api/user/all

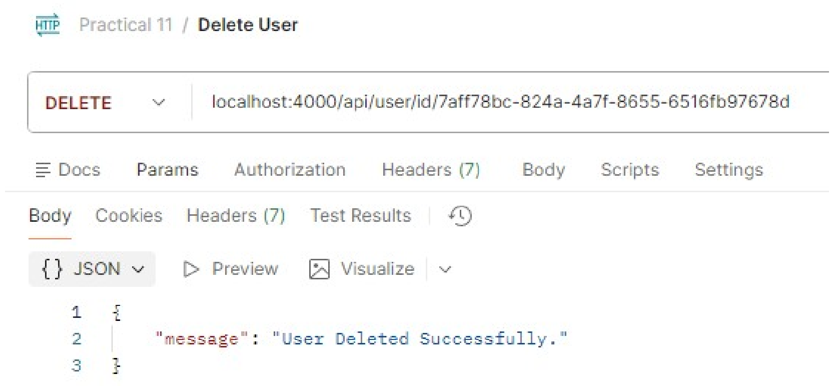
Docs | Params | Authorization | Headers (7) | Body | Scripts | Settings

Body | Cookies | Headers (7) | Test Results

{} JSON | Preview | Visualize

```
[
    {
        "_id": "7554700a-64c3-4bc6-97da-458749347620",
        "username": "John Doe",
        "password": "test1234"
    },
    {
        "_id": "5be1b114-8e63-47a0-bd08-ae0a033f009f",
        "username": "Jane Doe",
        "password": "test1234"
    },
    {
        "_id": "7aff78bc-824a-4a7f-8655-6516fb97678d",
        "username": "Test Updated",
        "password": "test1234"
    }
]
```

Delete User



Practical 11 / Delete User

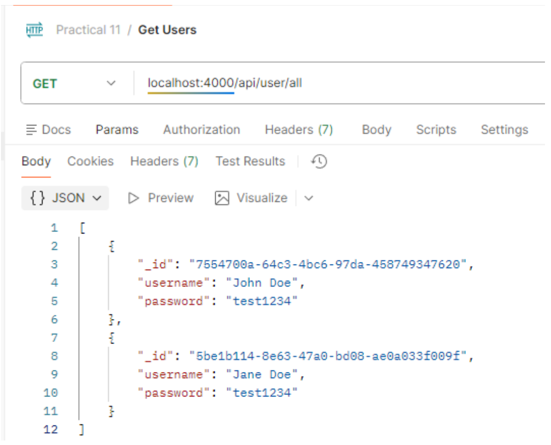
DELETE localhost:4000/api/user/id/7aff78bc-824a-4a7f-8655-6516fb97678d

Docs | Params | Authorization | Headers (7) | Body | Scripts | Settings

Body | Cookies | Headers (7) | Test Results

{} JSON | Preview | Visualize

```
{
    "message": "User Deleted Successfully."
}
```



Practical 11 / Get Users

GET localhost:4000/api/user/all

Docs | Params | Authorization | Headers (7) | Body | Scripts | Settings

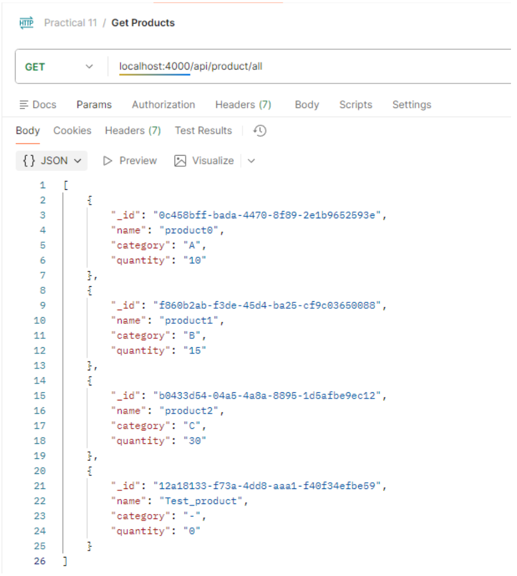
Body | Cookies | Headers (7) | Test Results

{} JSON | Preview | Visualize

```
[
    {
        "_id": "7554700a-64c3-4bc6-97da-458749347620",
        "username": "John Doe",
        "password": "test1234"
    },
    {
        "_id": "5be1b114-8e63-47a0-bd08-ae0a033f009f",
        "username": "Jane Doe",
        "password": "test1234"
    }
]
```

Product Router

Get All Product



Practical 11 / Get Products

GET localhost:4000/api/product/all

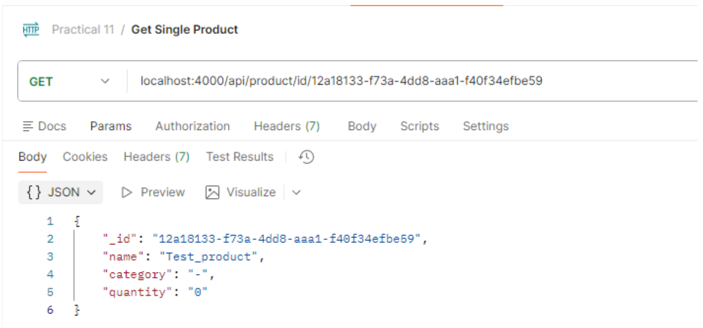
Docs | Params | Authorization | Headers (7) | Body | Scripts | Settings

Body | Cookies | Headers (7) | Test Results

{} JSON | Preview | Visualize

```
[
    {
        "_id": "0c458bff-bada-4470-8f89-2e1b9652593e",
        "name": "product0",
        "category": "A",
        "quantity": "10"
    },
    {
        "_id": "f860b2ab-f3de-45d4-ba25-cf9c03650088",
        "name": "product1",
        "category": "B",
        "quantity": "15"
    },
    {
        "_id": "b0433d54-04a5-4a8a-8895-1d5afbe9ec12",
        "name": "product2",
        "category": "C",
        "quantity": "30"
    },
    {
        "_id": "12a18133-f73a-4dd8-aaa1-f40f34efbe59",
        "name": "Test_product",
        "category": "-",
        "quantity": "0"
    }
]
```

Get Single Product



Practical 11 / Get Single Product

GET localhost:4000/api/product/id/12a18133-f73a-4dd8-aaa1-f40f34efbe59

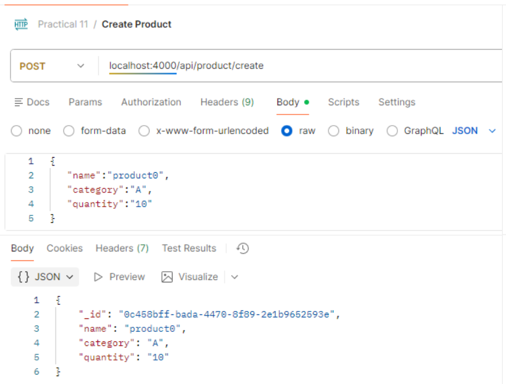
Docs | Params | Authorization | Headers (7) | Body | Scripts | Settings

Body | Cookies | Headers (7) | Test Results

{} JSON | Preview | Visualize

```
{
    "_id": "12a18133-f73a-4dd8-aaa1-f40f34efbe59",
    "name": "Test_product",
    "category": "-",
    "quantity": "0"
}
```

Create Product



Practical 11 / Create Product

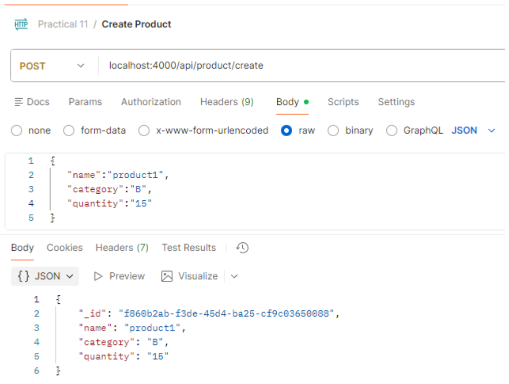
```
POST  localhost:4000/api/product/create
```

Body: raw [✓ raw] JSON

```
{
    "name":"product0",
    "category":"A",
    "quantity":"10"
}
```

Response Body (JSON):

```
{
    "_id": "0c458bff-bada-4470-8f89-2e1b9652593e",
    "name": "product0",
    "category": "A",
    "quantity": "10"
}
```

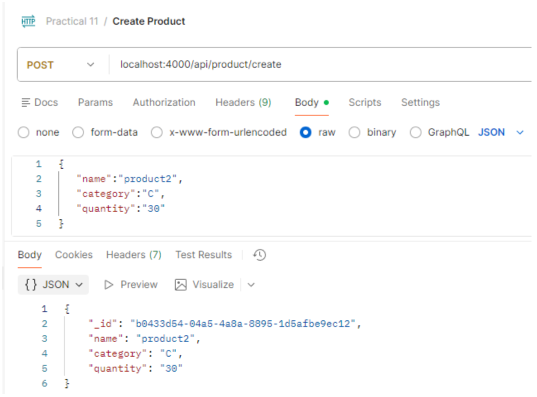


Practical 11 / Create Product

```
POST  localhost:4000/api/product/create
```

```
{
    "name":"product1",
    "category":"B",
    "quantity":"15"
}
```

```
{
    "_id": "f860b2ab-f3de-45d4-ba25-cf9c03650088",
    "name": "product1",
    "category": "B",
    "quantity": "15"
}
```



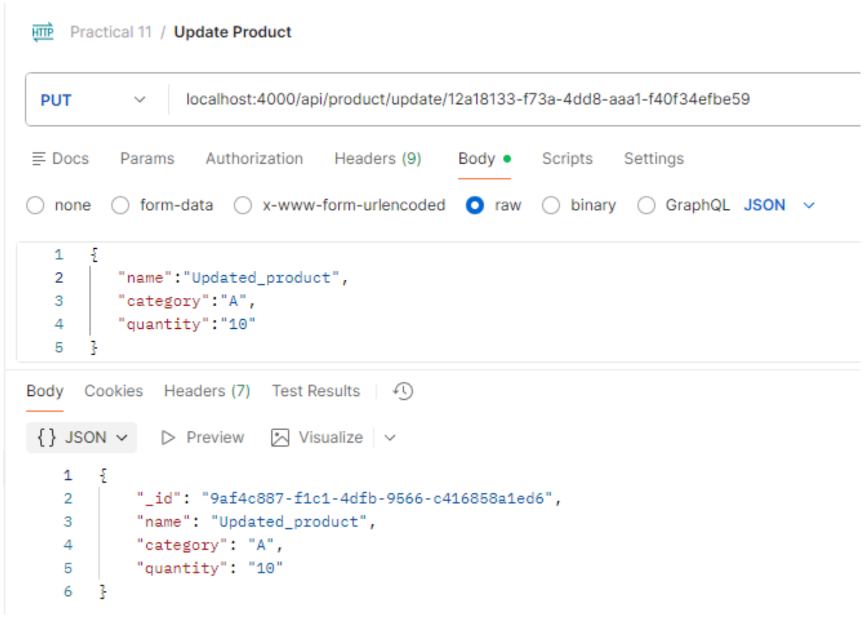
Practical 11 / Create Product

```
POST  localhost:4000/api/product/create
```

```
{
    "name":"product2",
    "category":"C",
    "quantity":"30"
}
```

```
{
    "_id": "b0433d54-04a5-4a8a-8895-1d5afbe9ec12",
    "name": "product2",
    "category": "C",
    "quantity": "30"
}
```

Update Product



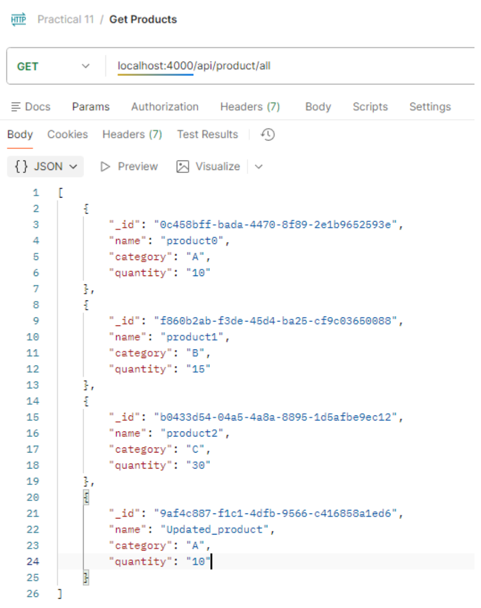
Practical 11 / Update Product

```
PUT  localhost:4000/api/product/update/12a18133-f73a-4dd8-aaa1-f40f34efbe59
```

Body: [✓ raw] JSON

```
{
    "name":"Updated_product",
    "category":"A",
    "quantity":"10"
}
```

```
{
    "_id": "9af4c887-f1c1-4dfb-9566-c416858a1ed6",
    "name": "Updated_product",
    "category": "A",
    "quantity": "10"
}
```

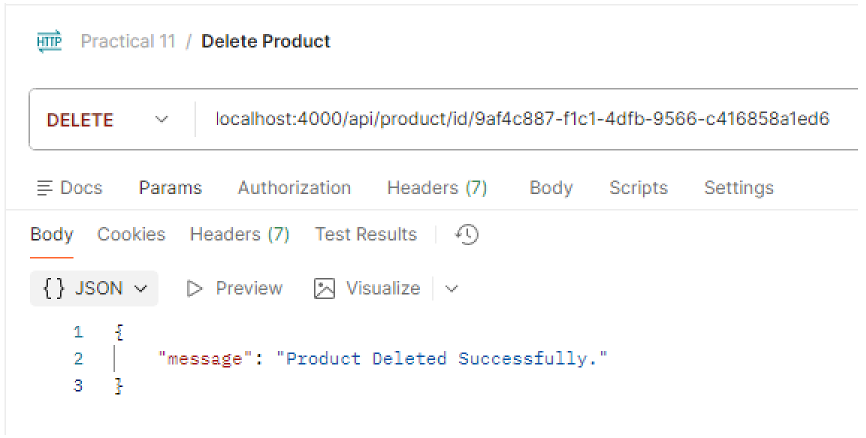


Practical 11 / Get Products

```
GET  localhost:4000/api/product/all
```

```
[
    {
        "_id": "0c458bff-bada-4470-8f89-2e1b9652593e",
        "name": "product0",
        "category": "A",
        "quantity": "10"
    },
    {
        "_id": "f860b2ab-f3de-45d4-ba25-cf9c03650088",
        "name": "product1",
        "category": "B",
        "quantity": "15"
    },
    {
        "_id": "b0433d54-04a5-4a8a-8895-1d5afbe9ec12",
        "name": "product2",
        "category": "C",
        "quantity": "30"
    },
    {
        "_id": "9af4c887-f1c1-4dfb-9566-c416858a1ed6",
        "name": "Updated_product",
        "category": "A",
        "quantity": "10"
    }
]
```

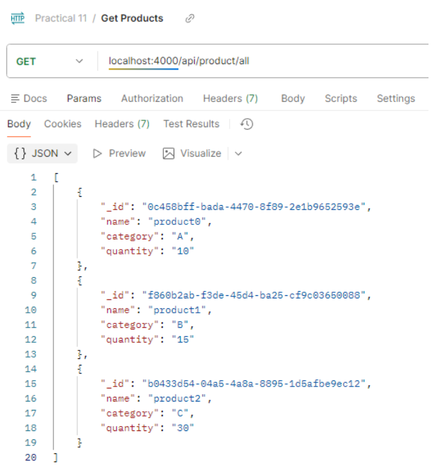
Delete Product



Practical 11 / Delete Product

```
DELETE  localhost:4000/api/product/id/9af4c887-f1c1-4dfb-9566-c416858a1ed6
```

```
{
    "message": "Product Deleted Successfully."
}
```



Practical 11 / Get Products

```
GET  localhost:4000/api/product/all
```

```
[
    {
        "_id": "0c458bff-bada-4470-8f89-2e1b9652593e",
        "name": "product0",
        "category": "A",
        "quantity": "10"
    },
    {
        "_id": "f860b2ab-f3de-45d4-ba25-cf9c03650088",
        "name": "product1",
        "category": "B",
        "quantity": "15"
    },
    {
        "_id": "b0433d54-04a5-4a8a-8895-1d5afbe9ec12",
        "name": "product2",
        "category": "C",
        "quantity": "30"
    }
]
```