Лабораторна робота №5

Зв'язування користувачів і задач

Хід роботи :

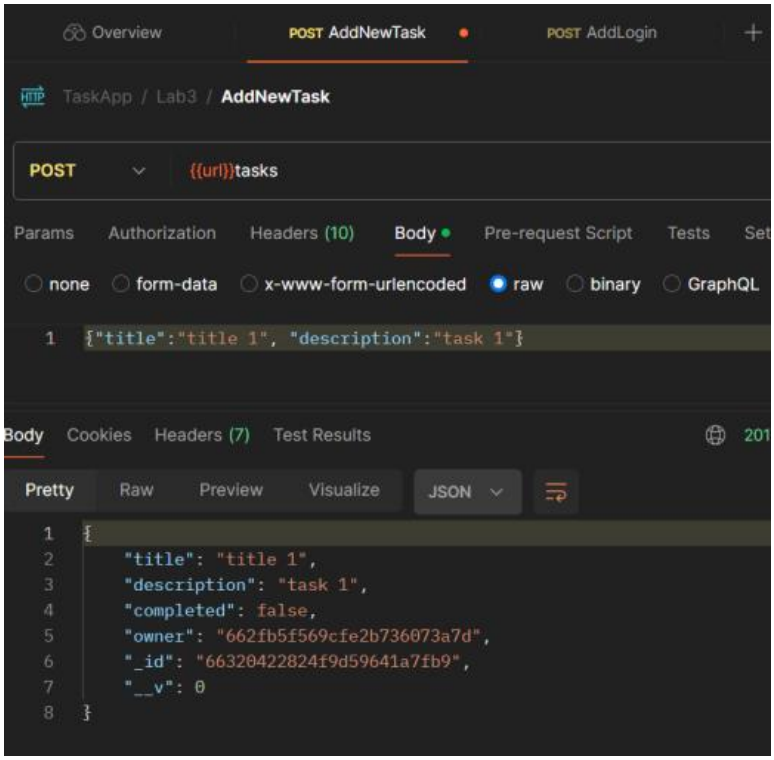
1.В моделі Task створіть поле для зберігання id користувача

```
  },
  owner: {
      type: mongoose.Schema.Types.ObjectId,
      ref: 'User',
      required: true
  }
})
```

2. В маршрутизаторі підключіть auth та модифікуйте обробник додавання нового запису

```
Aliev Omar *
router.get( path: '/tasks',auth, async (req : Request<P, ResBody, ReqBo
    try {
        const tasks : Query  = await Task.find();
        res.send(tasks);
    } catch (error) {
        res.status( code: 500).send(error);
    }
});
```

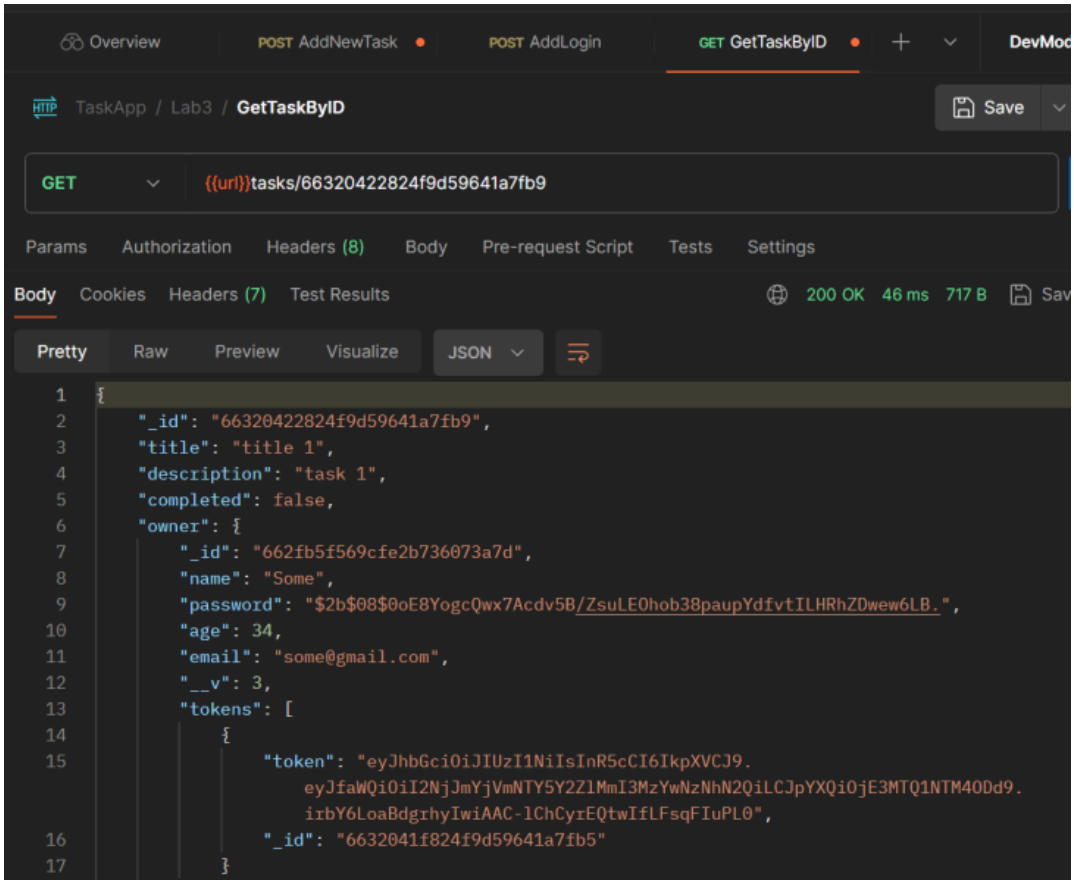
					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.21.121.01.000 – Лр.5		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Алієв О.Є			Звіт з лабораторної роботи №5	Літ.	Арк. 1 / Аркушів
Перевір.		Сидорчук В.О					
Реценз.						ФІКТ, гр.ІПЗ-22-1	
Н. Контр.							
Зав.каф.							



Для отримання запису користувача, якому належить дана задача

```
38

    new *
39  router.get( path: '/tasks/:id',auth,async (req : Request<P, ResBody, ReqBody
40      try {
41          const task : Query  = await Task.findById(req.params.id)
42          await task.populate('owner')
43          res.status( code: 200).send(task)
44      }
45      catch (error){
46          res.status(500).send(error);
47      }
48  })
49
```



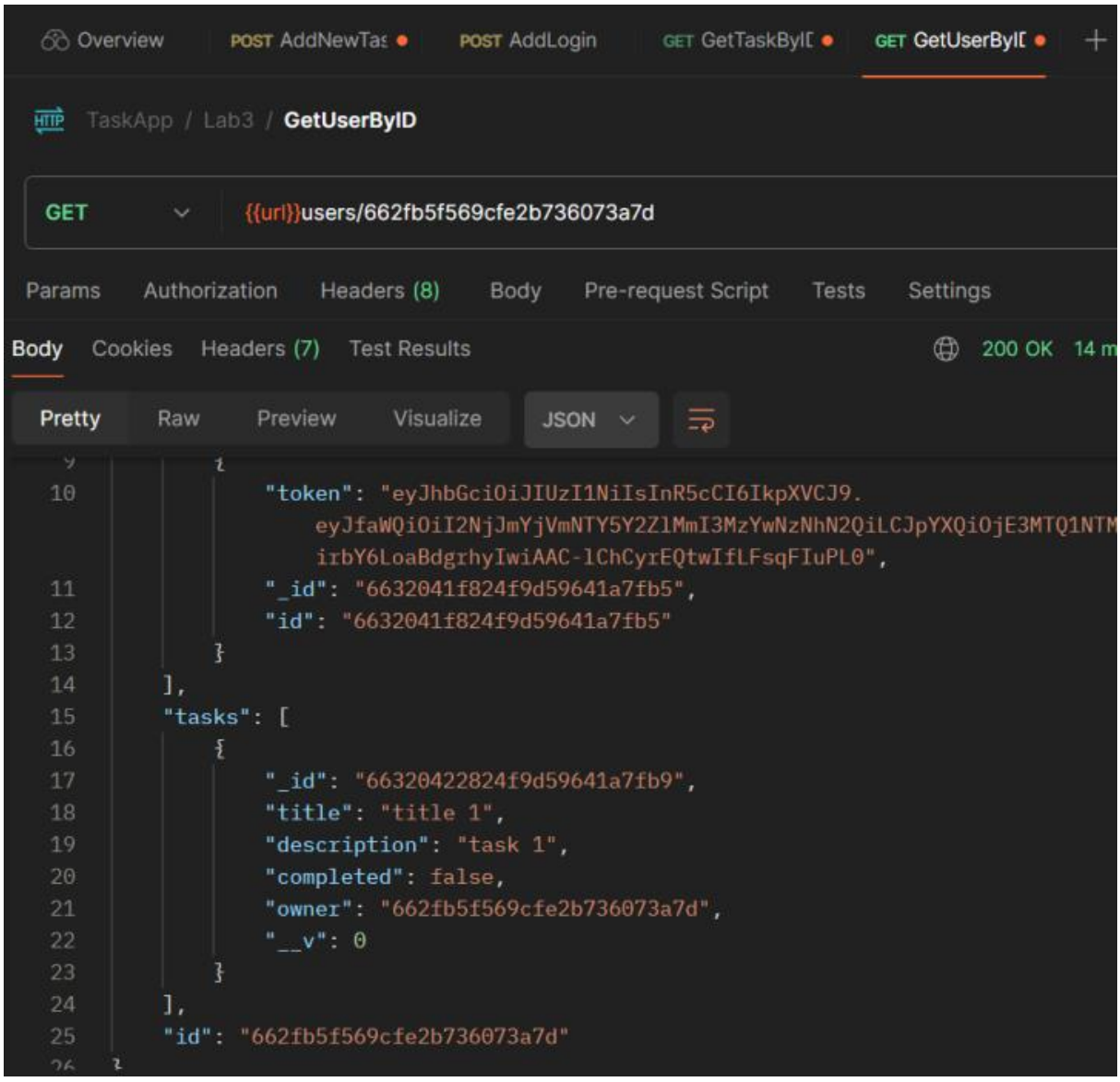
Для отримання задач, які належать даному користувачу, створіть віртуальне поле в моделі User. Відобразіть дані поля tasks в профілі користувача (запит /users/me)

```
            message: 'Error: Email is invalid'
        },
    },
    tokens: [{
        token: {
            type: String,
            required: true
        }
    }]
}, { toJSON: { virtuals: true }, toObject:{virtuals: true} })

```

```
Aliev Omar *
router.get( path: '/users/:id', auth,async (req : Request<P, ResBody, ReqBody, ReqQuery, Lo
    try {
        const user : Query  = await User.findById(req.params.id)
        await user.populate('tasks');
        res.status( code: 200).send(user)
    }
    catch (error){
        res.status( code: 500).send(error)
    }
});

Aliev Omar
router.post( path: '/users', handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, L
    try {
        const user : HydratedDocument<InferSchemaType  = new User(req.body);
```



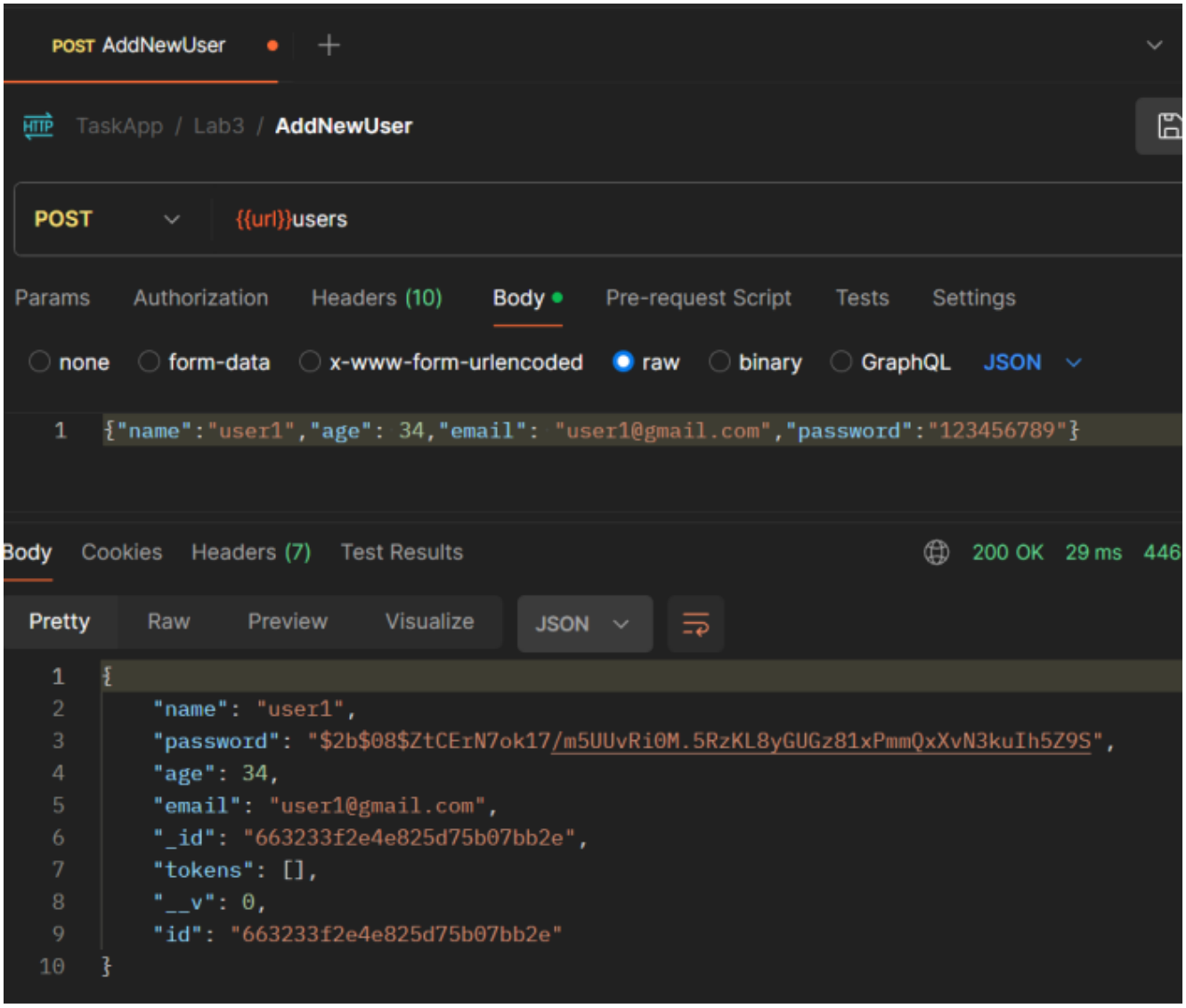
2. Реалізувати можливість роботи користувачам лише з власними завданнями

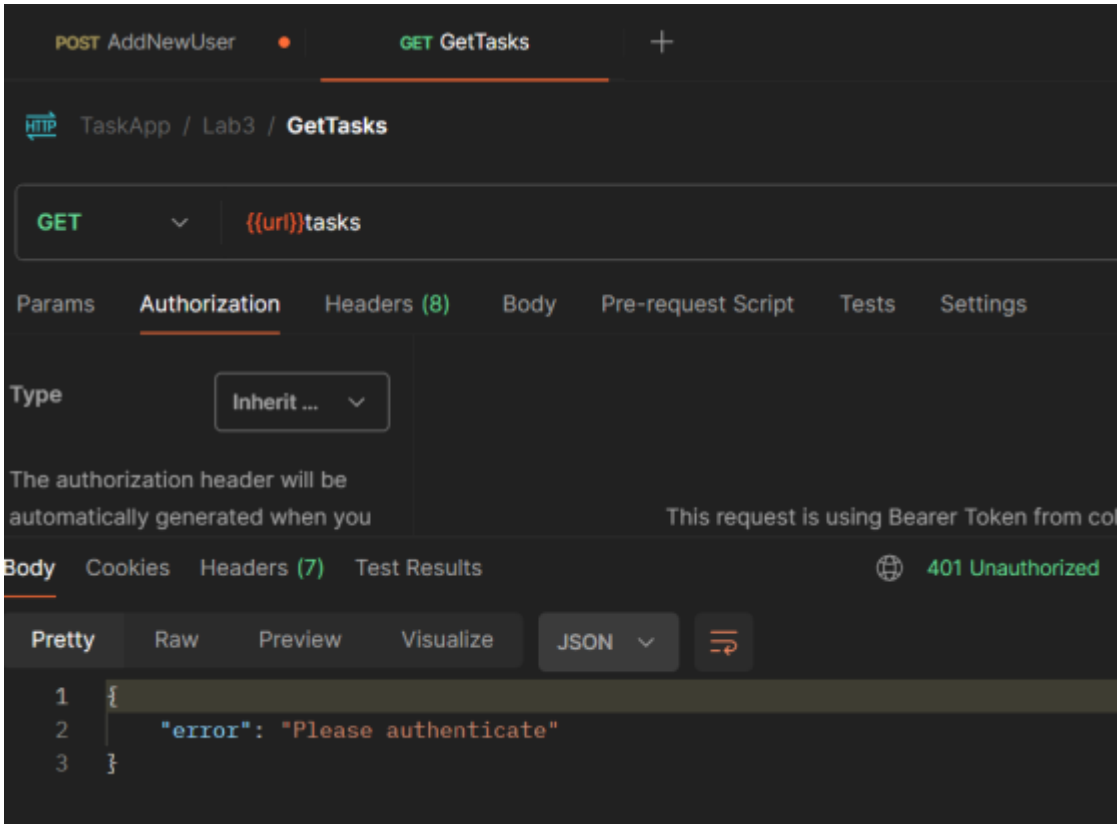
```
Aliev Omar *
router.get( path: '/tasks/:id', auth, async (req : Request<P, ResBody, ReqBody, ReqQuery, Loca
    try {
        const task : Query  = await Task.findById(req.params.id)
        if (!task) {
            throw new Error("Not existing")
        }
        const belongs : boolean  = task.owner._id.toString() === req.user._id.toString()
        if (!belongs) {
            return res.status( code: 404).send( body: {
                error: "IdError",
                message: "This task is not yours"
            })
        }
        await task.populate('owner')
        res.status( code: 200).send(task)
    }
    catch (error) {
        res.status( code: 500).send(error)
    }
});
```

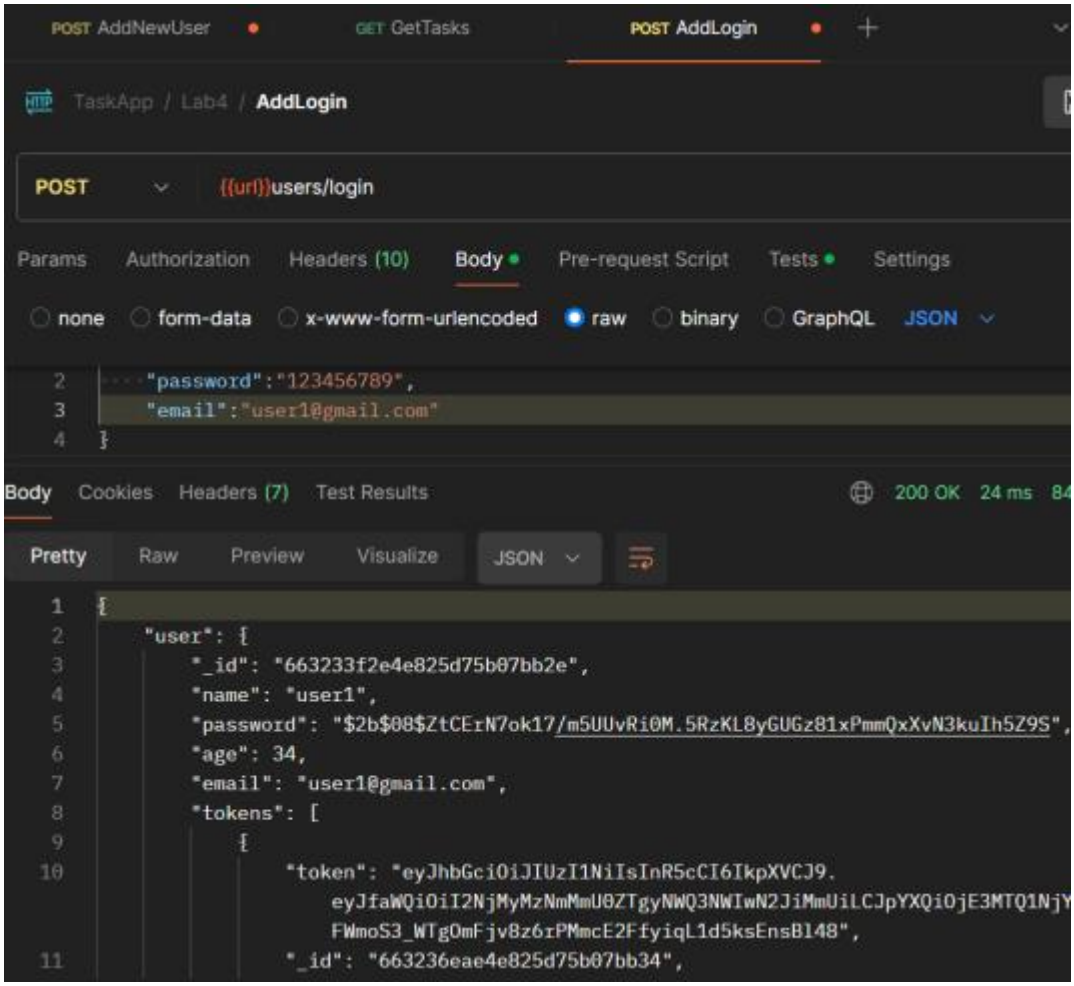
```
router.delete( path: '/tasks/:id', auth, async (req : Request<P, ResBody, ReqBody, ReqQuery,
    const taskId = req.params.id;
    try {
        const task : Query  = await Task.findById(taskId);
        if (task.owner._id.toString() !== req.user._id.toString()) {
            return res.status( code: 404).send( body: {
                error: "IdError",
                message: "This task is not yours"
            });
        }
        if (!task) {
            return res.status( code: 404).send( body: "Task not found");
        }
        await Task.deleteOne(task)
        res.status( code: 200).send(task);
    } catch (error) {
        console.error("Error deleting task:", error);
        res.status( code: 500).send( body: "Internal Server Error");
    }
});
```

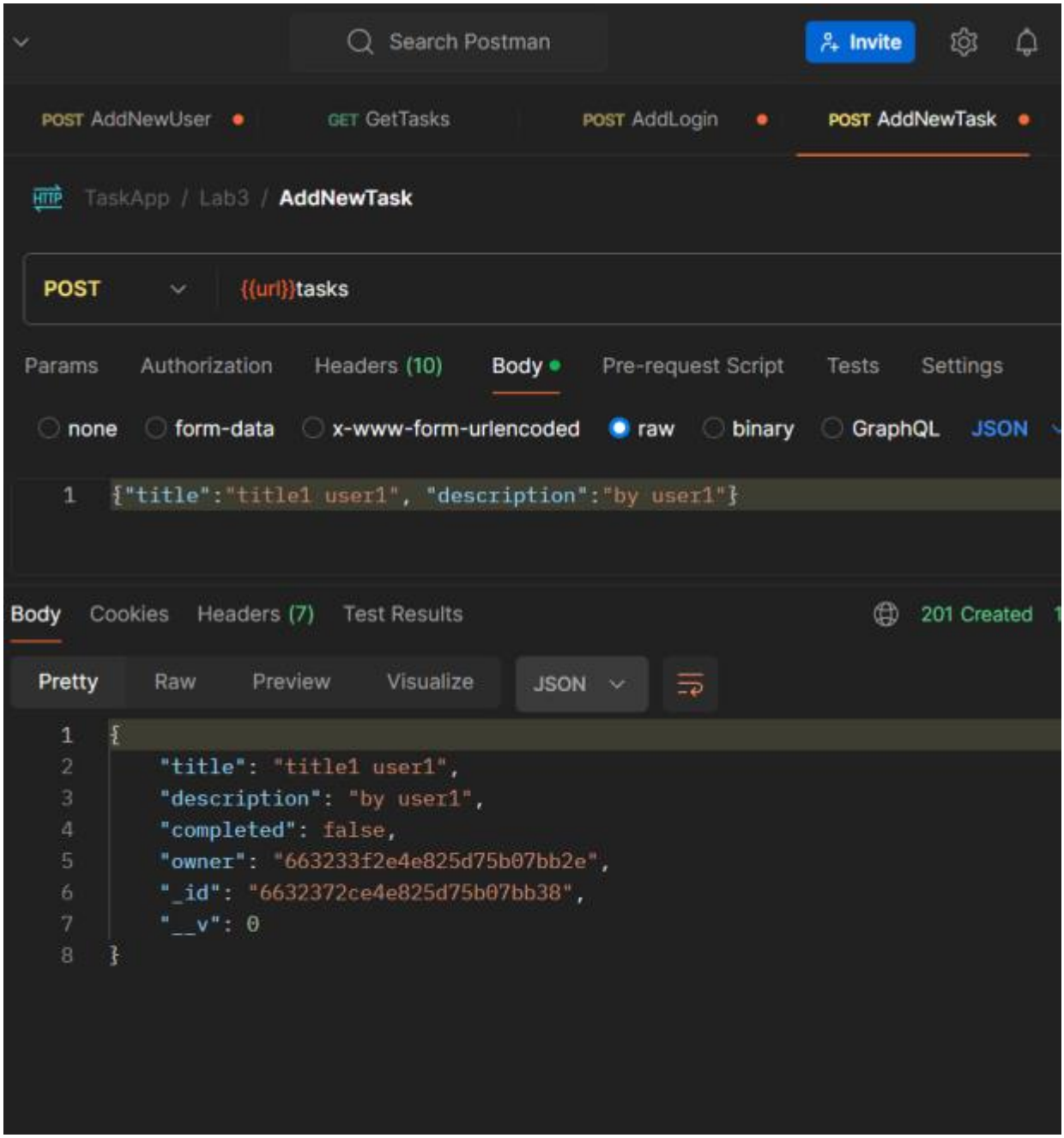
```
Aliev Omar *
router.patch( path: '/tasks/:id', auth, async (req : Request<P, ResBody, ReqBody, ReqQuery, Lo
    const taskId = req.params.id;
    try {
        const task : Query  = await Task.findById(taskId);
        if (task.owner._id.toString() !== req.user._id.toString()) {
            return res.status( code: 404).send( body: {
                error: "IdError",
                message: "This task is not yours"
            });
        }
        if (!task) {
            return res.status( code: 404).send( body: "Task not found");
        }
        const updateData : ReqBody  = req.body
        await Task.updateOne(task, updateData)
        res.status( code: 200).send(task);
    } catch (e) {
        next(e)
    }
})
```

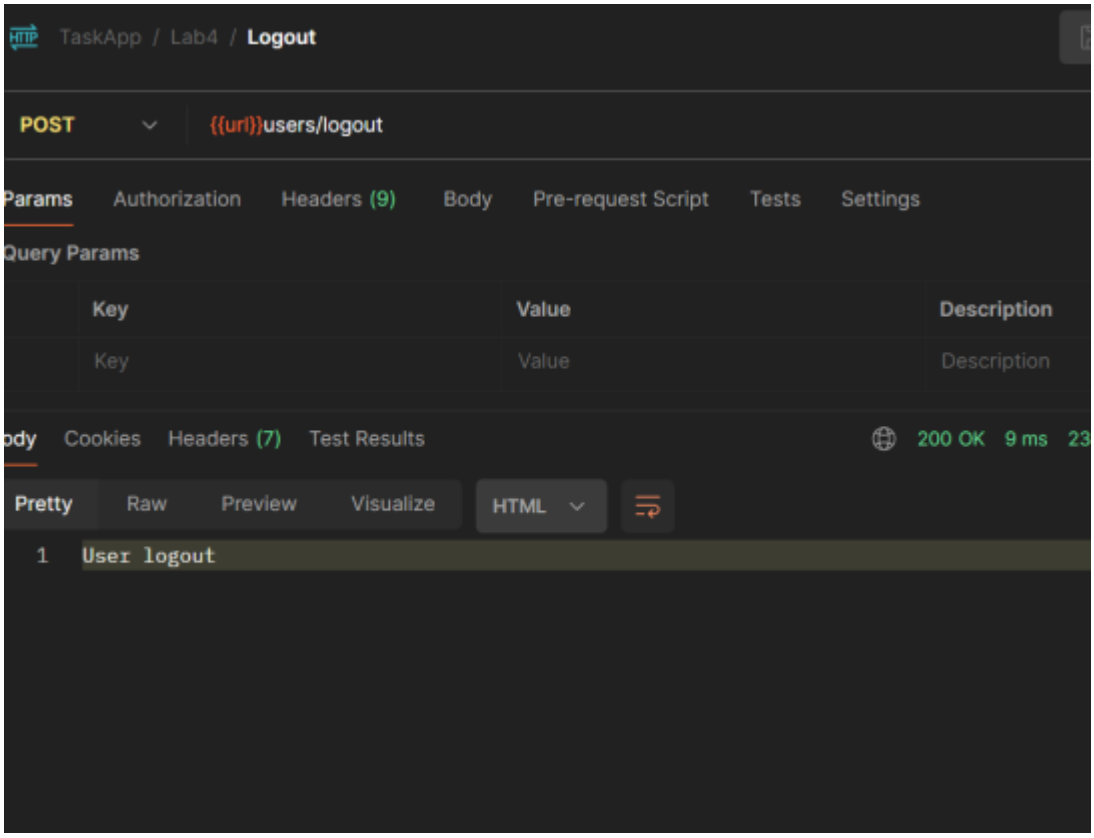
Послідовність тестів

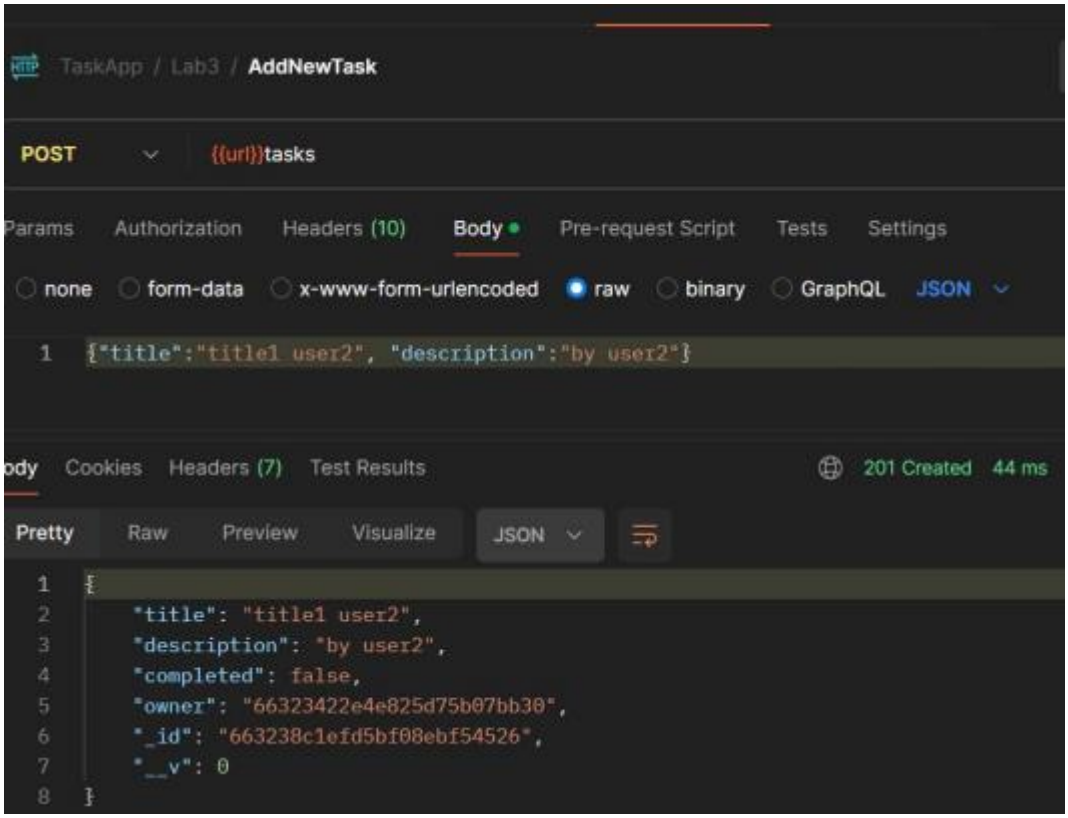

POST AddNewUser
TaskApp / Lab3 / AddNewUser
POST
{{url}}users
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
1 {"name":"user1","age": 34,"email": "user1@gmail.com","password":"123456789"}
Body
Cookies
Headers (7)
Test Results
200 OK
29 ms
446
Pretty
Raw
Preview
Visualize
JSON
1 {
2 "name": "user1",
3 "password": "$2b$08$ZtCErN7ok17/m5UUvRi0M.5RzKL8yGUGz81xPmmQxXvN3kuIh5Z9S",
4 "age": 34,
5 "email": "user1@gmail.com",
6 "_id": "663233f2e4e825d75b07bb2e",
7 "tokens": [],
8 "__v": 0,
9 "id": "663233f2e4e825d75b07bb2e"
10 }

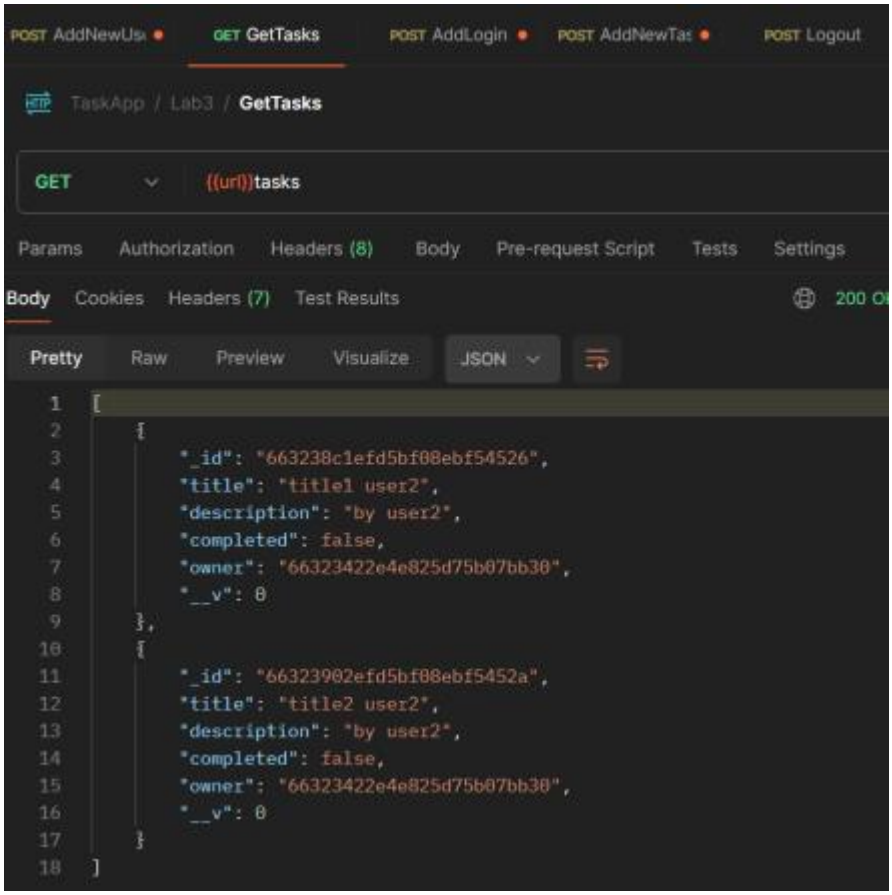

POST AddNewUser
GET GetTasks
TaskApp / Lab3 / GetTasks
GET
{{url}}tasks
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Type
Inherit ...
The authorization header will be
automatically generated when you
This request is using Bearer Token from co
Body
Cookies
Headers (7)
Test Results
401 Unauthorized
Pretty
Raw
Preview
Visualize
JSON
1 {
2 "error": "Please authenticate"
3 }

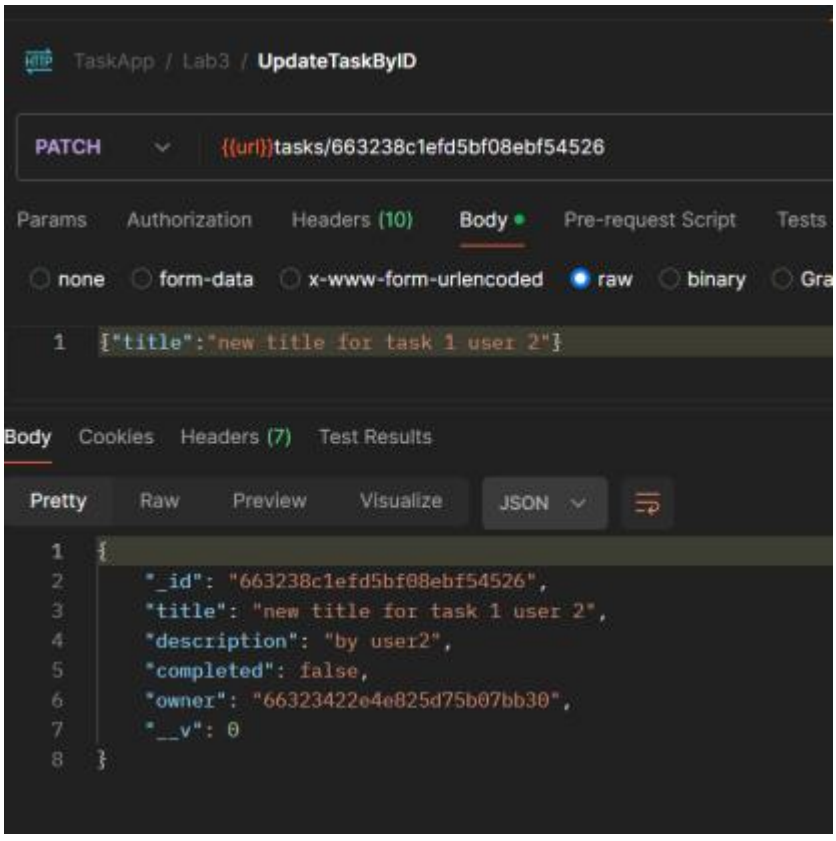

POST AddNewUser
GET GetTasks
POST AddLogin
TaskApp / Lab4 / AddLogin
POST
{{url}}users/login
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
"password":"123456789",
"email":"user1@gmail.com"
}
Body
Cookies
Headers (7)
Test Results
200 OK
24 ms
Pretty
Raw
Preview
Visualize
JSON
{
"user": {
"_id": "663233f2e4e825d75b07bb2e",
"name": "user1",
"password": "$2b$08$ZtCErN7ok17/m5UUvRi0M.5RzKL8yGUGz81xPmmQxXvN3kuIh5Z9S",
"age": 34,
"email": "user1@gmail.com",
"tokens": [
{
"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJfaWQiOiI2NjMyMzNmMmU0ZTgyNWQ3NWIwN2JiMmUiLCJpYXQiOjE3MTQ1NjY
FWmoS3_WTgOmFjv8z6rPMmcE2FfyiqL1d5ksEnsB148",
"_id": "663236eae4e825d75b07bb34",

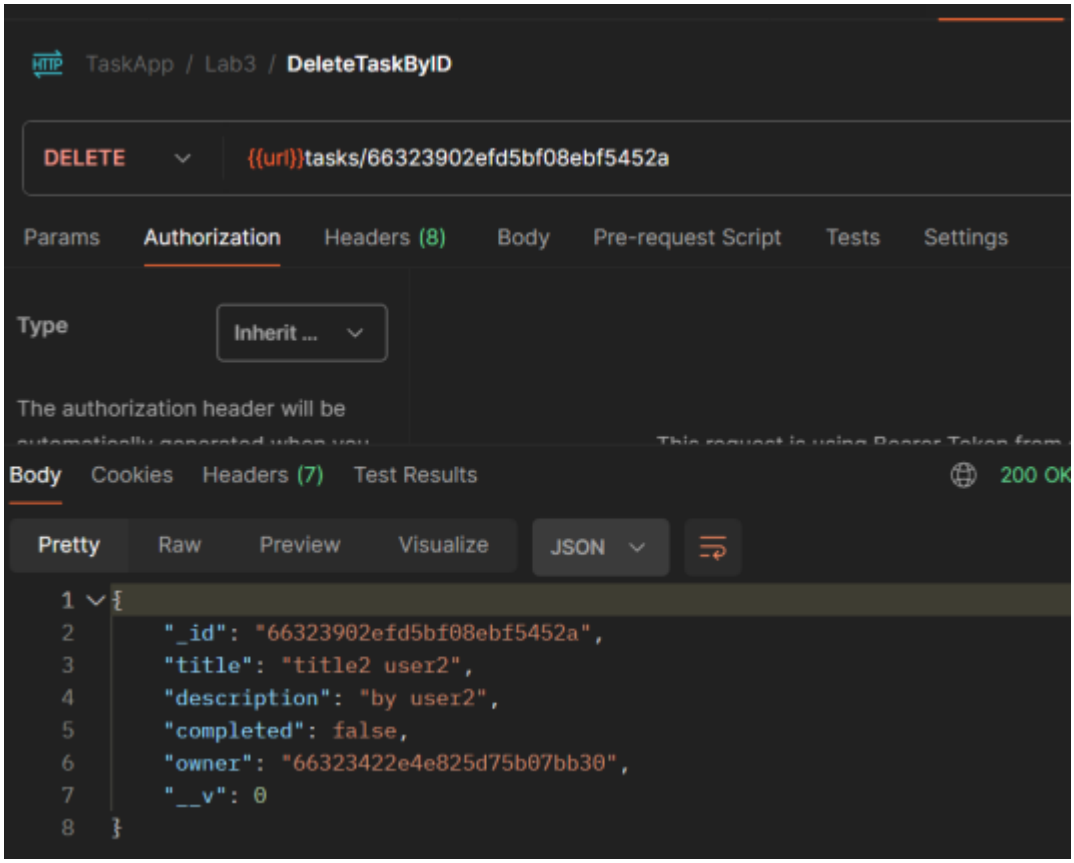

Search Postman
Invite
POST AddNewUser
GET GetTasks
POST AddLogin
POST AddNewTask
TaskApp / Lab3 / AddNewTask
POST
{{url}}tasks
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
1 {"title":"title1 user1", "description":"by user1"}
Body
Cookies
Headers (7)
Test Results
201 Created
Pretty
Raw
Preview
Visualize
JSON
{
"title": "title1 user1",
"description": "by user1",
"completed": false,
"owner": "663233f2e4e825d75b07bb2e",
"_id": "6632372ce4e825d75b07bb38",
"__v": 0
}

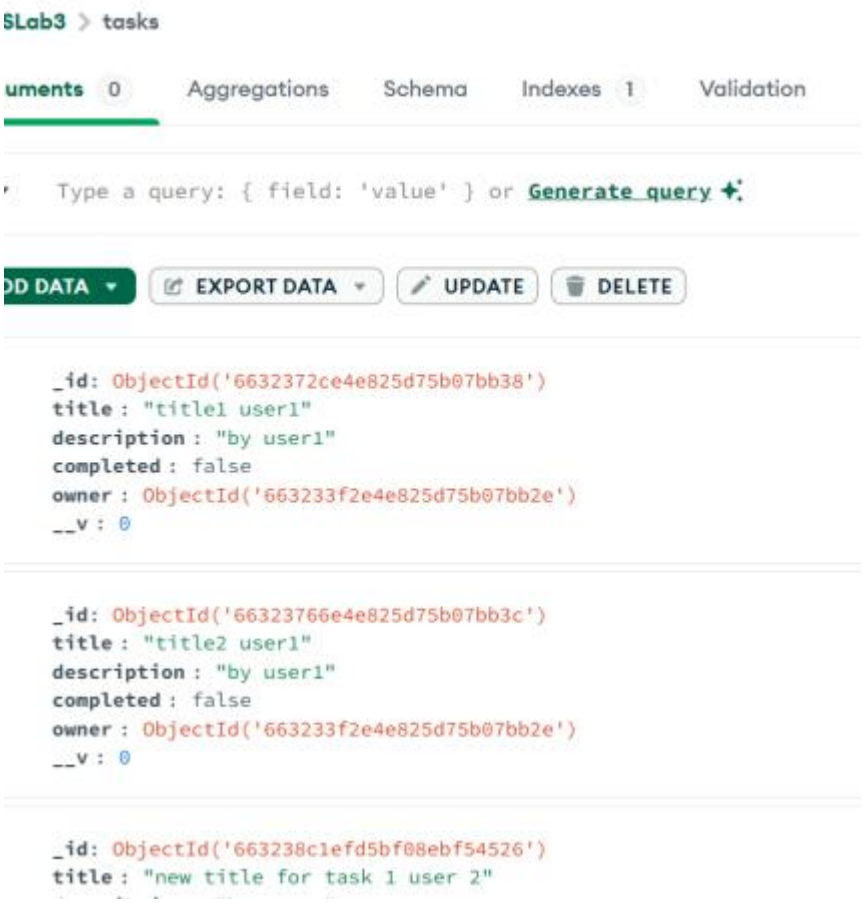

TaskApp / Lab4 / Logout
POST
{{url}}users/logout
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Query Params
Key
Value
Description
Key
Value
Description
Cookies
Headers (7)
Test Results
200 OK
9 ms
Pretty
Raw
Preview
Visualize
HTML
1 User logout

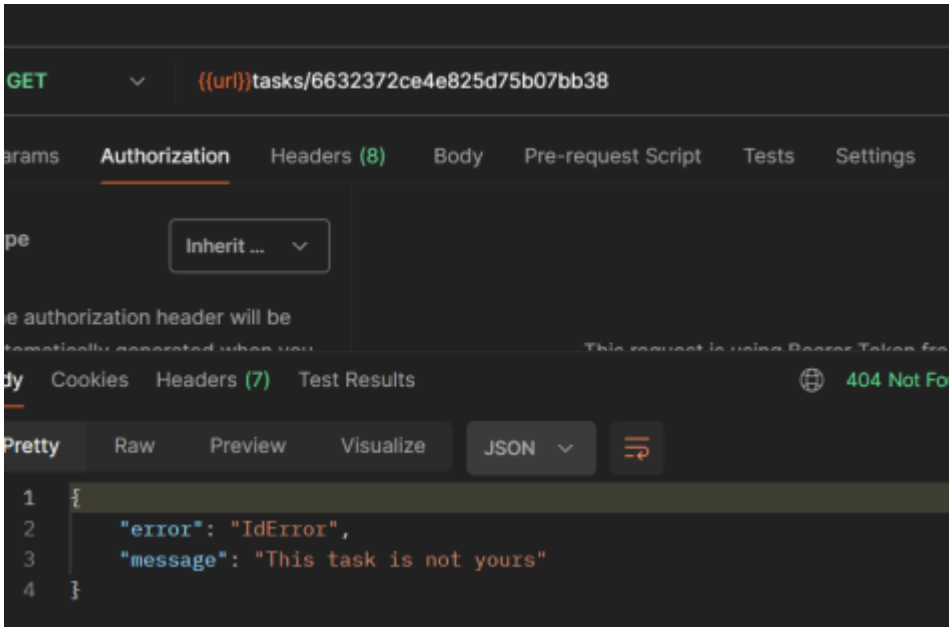

TaskApp / Lab3 / AddNewTask
POST
{{url}}tasks
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
1 {"title":"title1 user2", "description":"by user2"}
Cookies
Headers (7)
Test Results
201 Created
44 ms
Pretty
Raw
Preview
Visualize
JSON
1 {
2 "title": "title1 user2",
3 "description": "by user2",
4 "completed": false,
5 "owner": "66323422e4e825d75b07bb30",
6 "_id": "663238c1efd5bf08ebf54526",
7 "__v": 0
8 }

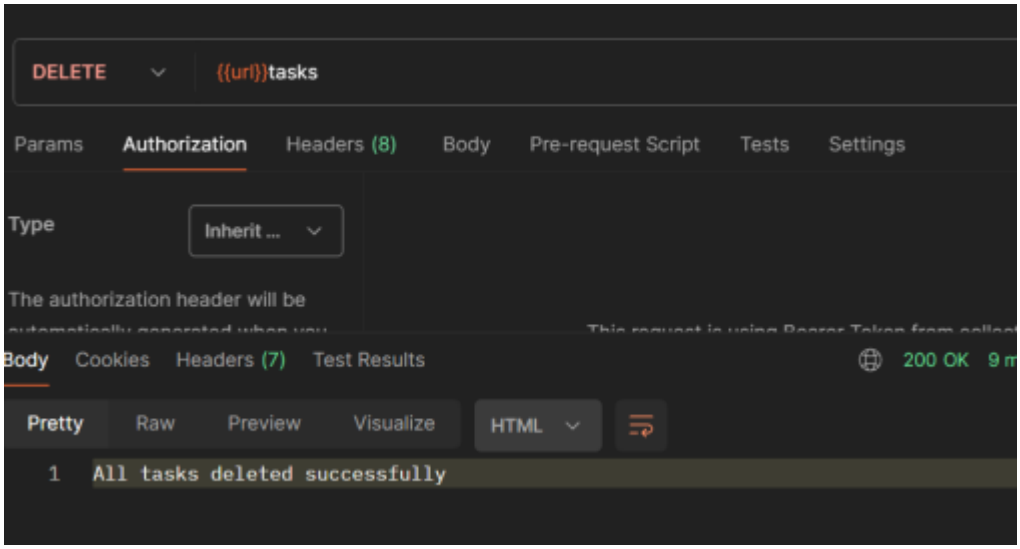

POST AddNewUs
GET GetTasks
POST AddLogin
POST AddNewTas
POST Logout
TaskApp / Lab3 / GetTasks
GET
{{url}}tasks
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Body
Cookies
Headers (7)
Test Results
200 O
Pretty
Raw
Preview
Visualize
JSON
[
{
"_id": "663238c1efd5bf08ebf54526",
"title": "title1 user2",
"description": "by user2",
"completed": false,
"owner": "66323422e4e825d75b07bb30",
"__v": 0
},
{
"_id": "66323902efd5bf08ebf5452a",
"title": "title2 user2",
"description": "by user2",
"completed": false,
"owner": "66323422e4e825d75b07bb30",
"__v": 0
}
]

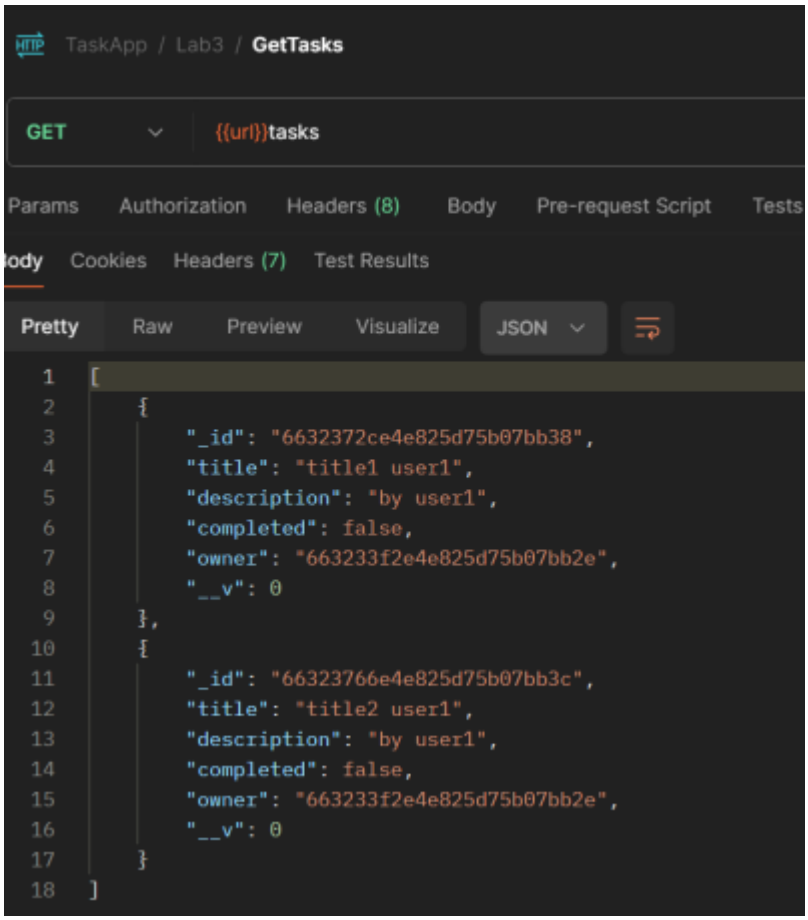

TaskApp / Lab3 / UpdateTaskByID
PATCH
{{url}}tasks/663238c1efd5bf08ebf54526
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
none
form-data
x-www-form-urlencoded
raw
binary
Gra
{"title":"new title for task 1 user 2"}
Body
Cookies
Headers (7)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"_id": "663238c1efd5bf08ebf54526",
"title": "new title for task 1 user 2",
"description": "by user2",
"completed": false,
"owner": "66323422e4e825d75b07bb30",
"__v": 0
}

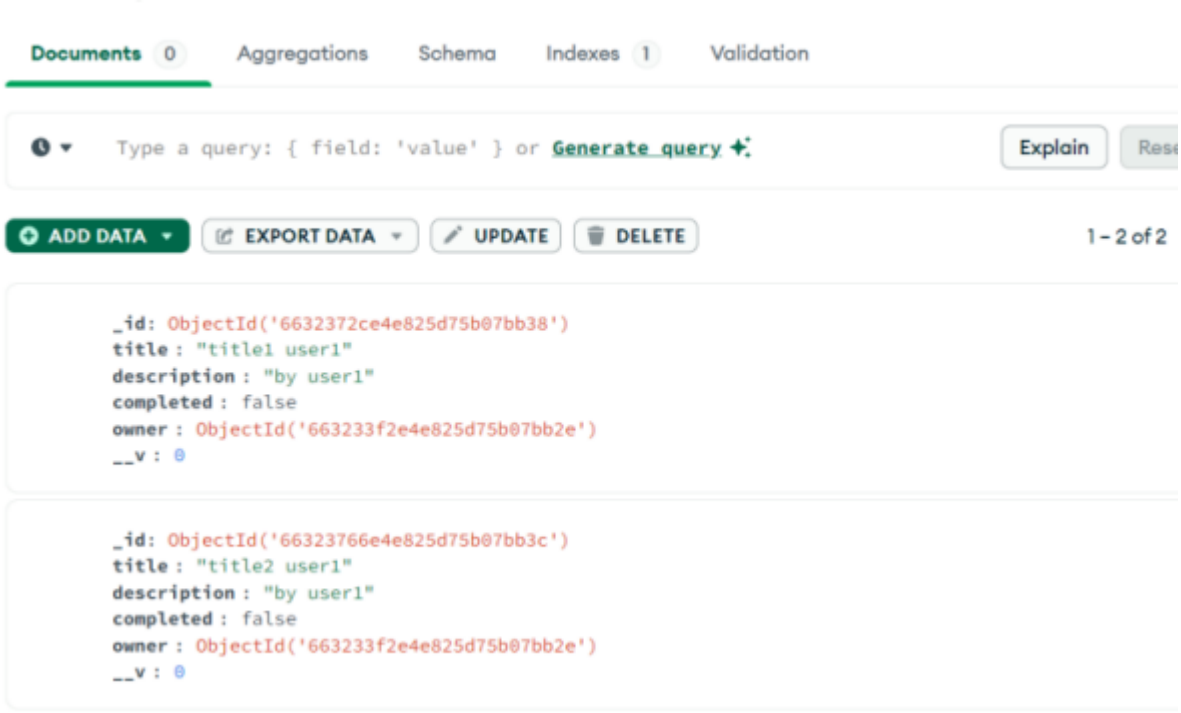

TaskApp / Lab3 / DeleteTaskByID
DELETE
{{url}}tasks/66323902efd5bf08ebf5452a
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Type
Inherit ...
The authorization header will be
Body
Cookies
Headers (7)
Test Results
200 OK
Pretty
Raw
Preview
Visualize
JSON
{
"_id": "66323902efd5bf08ebf5452a",
"title": "title2 user2",
"description": "by user2",
"completed": false,
"owner": "66323422e4e825d75b07bb30",
"__v": 0
}


SLab3 > tasks
uments 0
Aggregations
Schema
Indexes 1
Validation
Type a query: { field: 'value' } or Generate query
DD DATA
EXPORT DATA
UPDATE
DELETE
_id: ObjectId('6632372ce4e825d75b07bb38')
title : "title1 user1"
description : "by user1"
completed : false
owner : ObjectId('663233f2e4e825d75b07bb2e')
__v : 0
_id: ObjectId('66323766e4e825d75b07bb3c')
title : "title2 user1"
description : "by user1"
completed : false
owner : ObjectId('663233f2e4e825d75b07bb2e')
__v : 0
_id: ObjectId('663238c1efd5bf08ebf54526')
title : "new title for task 1 user 2"


GET
{{url}}tasks/6632372ce4e825d75b07bb38
arams
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
pe
Inherit ...
e authorization header will be
dy
Cookies
Headers (7)
Test Results
404 Not Fo
Pretty
Raw
Preview
Visualize
JSON
{
"error": "IdError",
"message": "This task is not yours"
}


DELETE
{{url}}tasks
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Type
Inherit ...
The authorization header will be
Body
Cookies
Headers (7)
Test Results
200 OK
Pretty
Raw
Preview
Visualize
HTML
1 All tasks deleted successfully


TaskApp / Lab3 / GetTasks
GET
{{url}}tasks
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Cookies
Headers (7)
Test Results
Pretty
Raw
Preview
Visualize
JSON
[
{
"_id": "6632372ce4e825d75b07bb38",
"title": "title1 user1",
"description": "by user1",
"completed": false,
"owner": "663233f2e4e825d75b07bb2e",
"__v": 0
},
{
"_id": "66323766e4e825d75b07bb3c",
"title": "title2 user1",
"description": "by user1",
"completed": false,
"owner": "663233f2e4e825d75b07bb2e",
"__v": 0
}
]



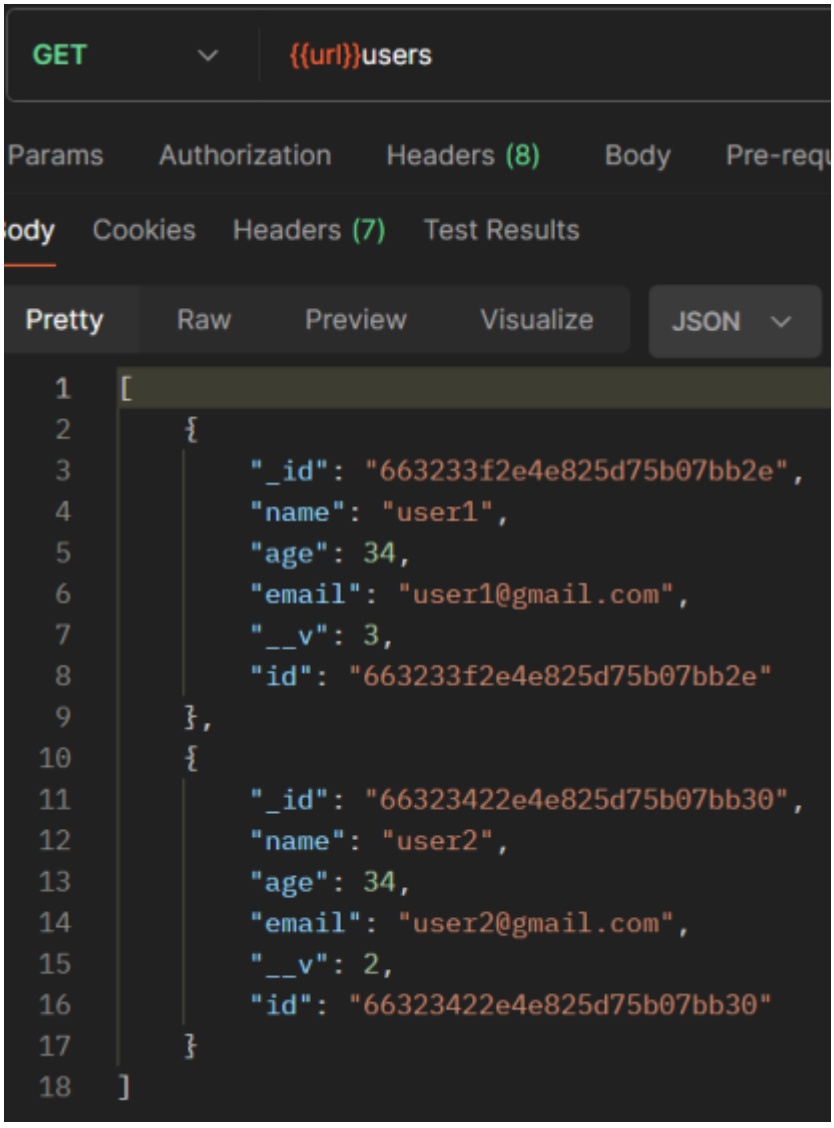
3. Забороніть відправку захищених даних на клієнт

```
    password: {type: String...},
    age: {type: Number...},
    email: {type: String...},
    tokens: [{...}]
}, { toJSON: { virtuals: true }, toObject:{virtuals: true} })

userSchema.virtual( name: 'tasks', {ref: 'Task'...})

Aliev Omar *
```

```
userSchema.methods.toJSON = function() {
    const user = this
    const userObject = user.toObject()
    delete userObject.password
    delete userObject.tokens
    return userObject
}
```



Висновок: на лабораторному занятті ми навчились зв’язувати дані.