Kristianstad University
SE-291 88 Kristianstad
+46 44-250 30 00
www.hkr.se

Bachelor of Computer Science in Education
Semester Year. Spring Semester 2022
Faculty of Natural Sciences

Lab 1
NodeJS

Kiasar "Kia" Mian

Content

Database

For this lab I decided on MongoDB. The document structure that was decided on is as follows:

_id String
name String
age Int32

Tasks

For this lab I decided to use *dotenv* as well as *ejs* to allow for easier and better handled dynamic loading of page routing and content.

All Users

Fetching from */api/users/* each account object is then looped through and the div *user_list* gets populated with content. It also assigns the user row an ID that matches the users id.

User Details

This function makes use of the *this* property. Since the button has a data-id matching the users id this is used to fetch the correct user from the database using a jQuery built in method *.attr*. The fields are then targeted and the name and age is rendered for that specific user.

Create User

This function targets the two input fields dedicated for creating a user. The values *name & age* are collected and then passed on to the backend that handles the generation of an ID.

Update User

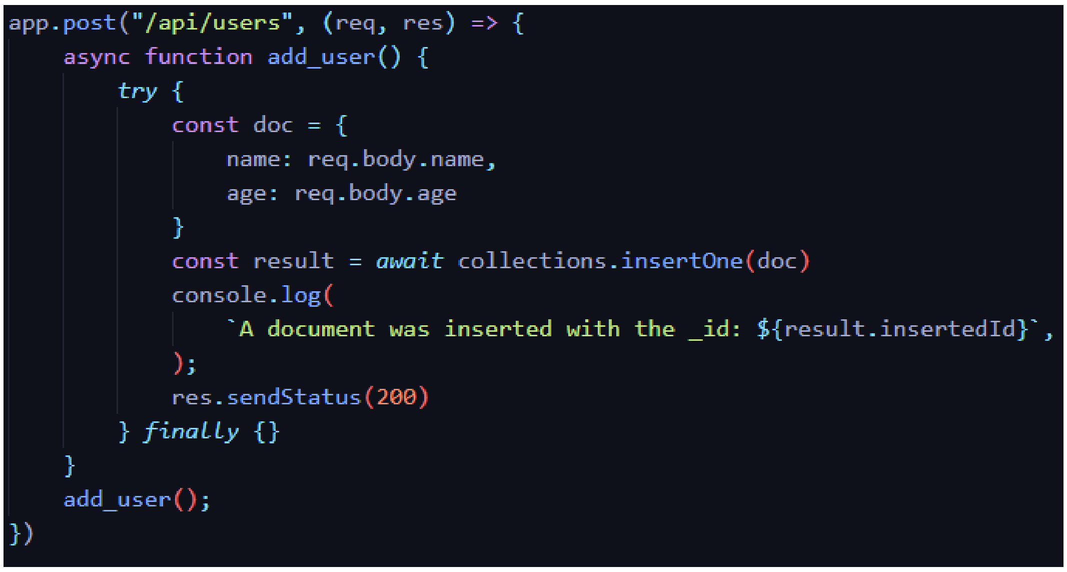
The update user section is in two parts. First the user pressed the *pencil* icon of the user they wish to edit. That users information is the populated into the user update section. The ID section is auto-filled but also disabled for a reason, this is to ensure that a incorrect ID isn't passed.

Delete User

The delete user like the update and view details function gets the *data-id* through the *this* object, and then sends a fetch request to the backend with the correct ID to delete.

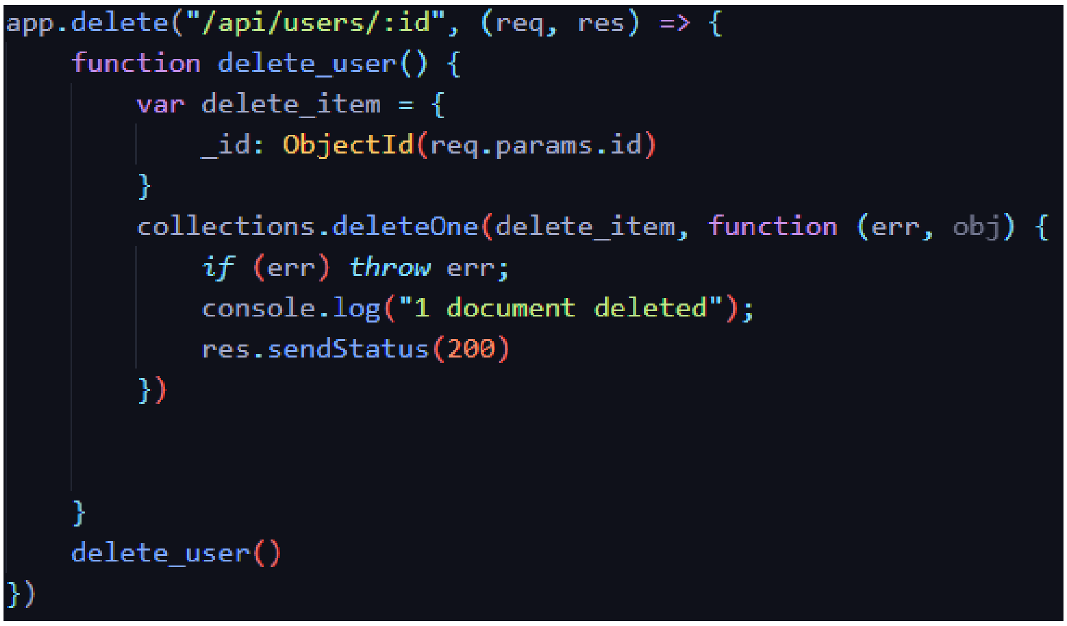
API

Create user – POST - /api/users



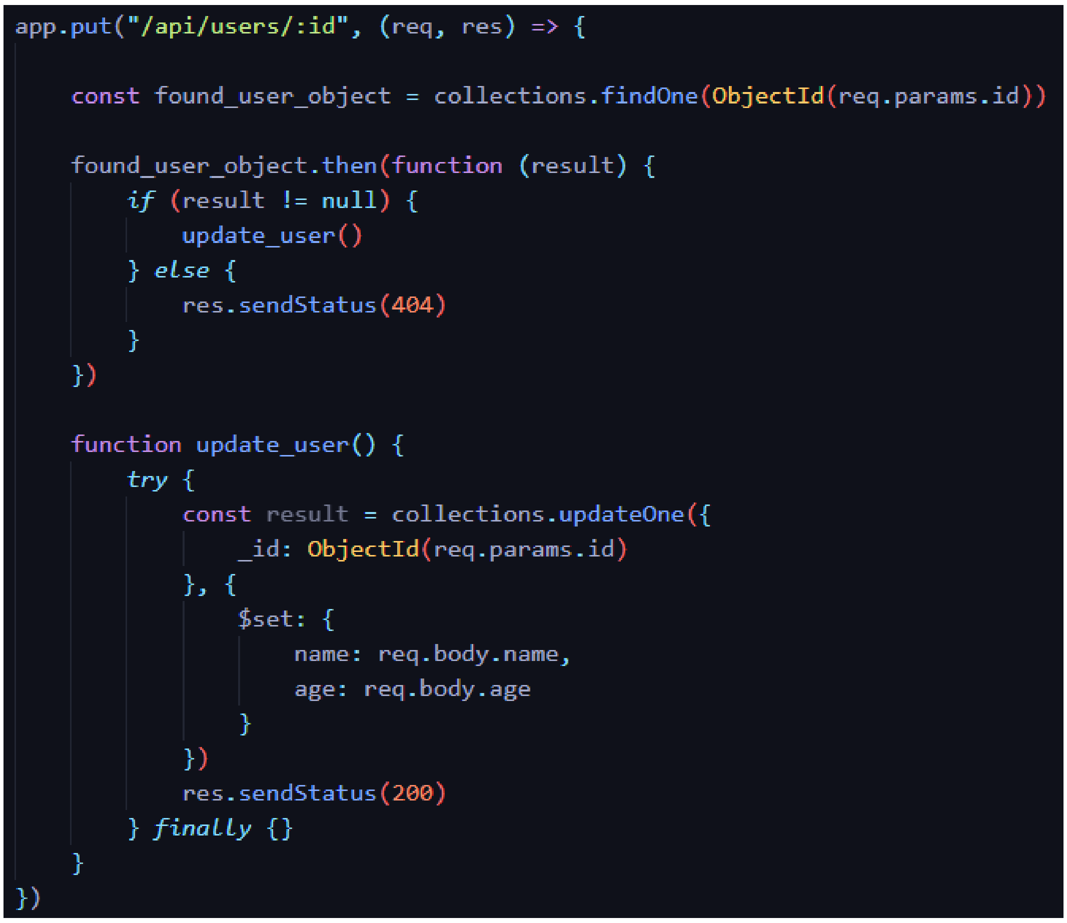
```
app.post("/api/users", (req, res) => {
    async function add_user() {
        try {
            const doc = {
                name: req.body.name,
                age: req.body.age
            }
            const result = await collections.insertOne(doc)
            console.log(
                `A document was inserted with the _id: ${result.insertedId}`,
            );
            res.sendStatus(200)
        } finally {}
    }
    add_user();
})
```

Delete user – DELETE - /api/users/:id



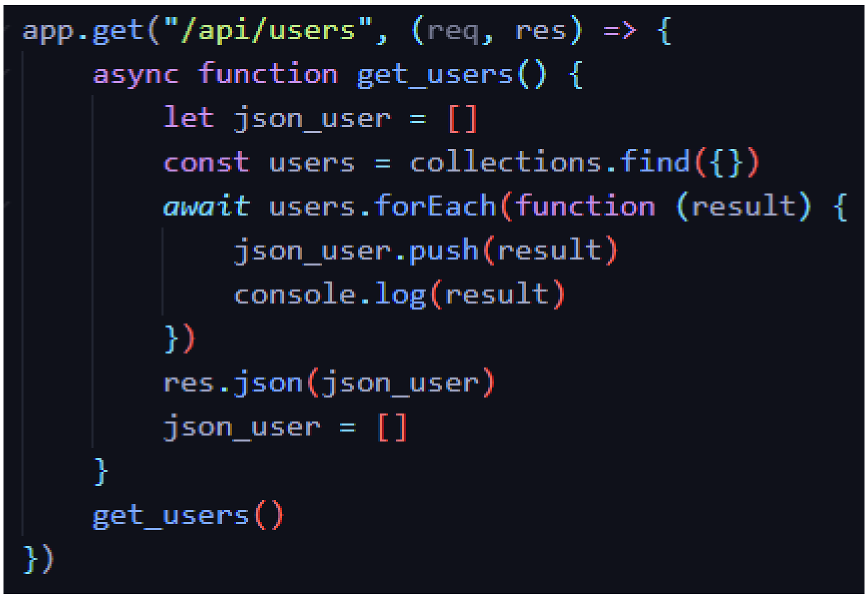
```
app.delete("/api/users/:id", (req, res) => {
    function delete_user() {
        var delete_item = {
            _id: ObjectId(req.params.id)
        }
        collections.deleteOne(delete_item, function (err, obj) {
            if (err) throw err;
            console.log("1 document deleted");
            res.sendStatus(200)
        })


    }
    delete_user()
})
```

Update user – PUT - /api/users/:id



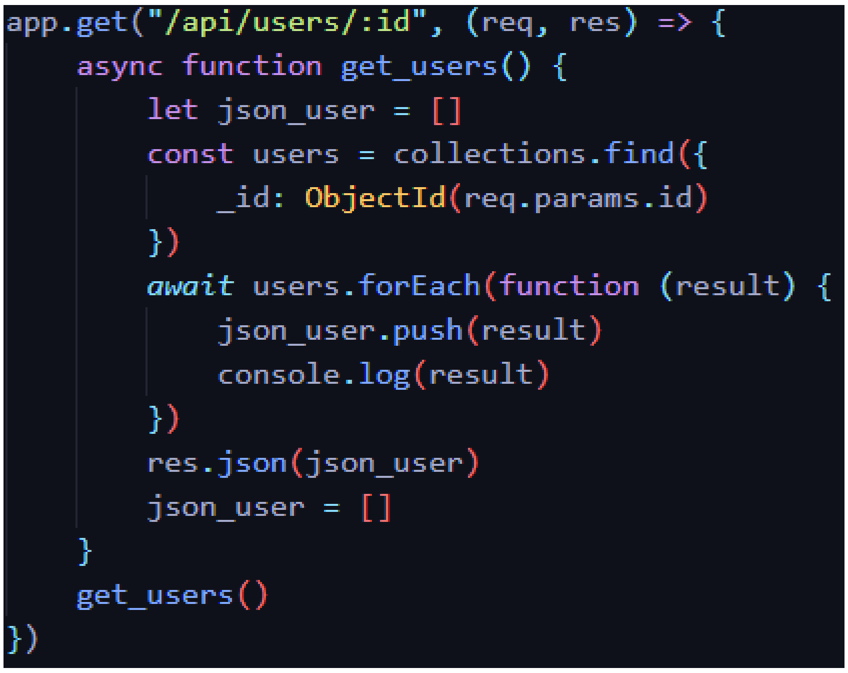
```
app.put("/api/users/:id", (req, res) => {

    const found_user_object = collections.findOne(ObjectId(req.params.id))

    found_user_object.then(function (result) {
        if (result != null) {
            update_user()
        } else {
            res.sendStatus(404)
        }
    })

    function update_user() {
        try {
            const result = collections.updateOne({
                _id: ObjectId(req.params.id)
            }, {
                $set: {
                    name: req.body.name,
                    age: req.body.age
                }
            })
            res.sendStatus(200)
        } finally {}
    }
})
```

View All – GET - /api/users



```
app.get("/api/users", (req, res) => {
    async function get_users() {
        let json_user = []
        const users = collections.find({})
        await users.forEach(function (result) {
            json_user.push(result)
            console.log(result)
        })
        res.json(json_user)
        json_user = []
    }
    get_users()
})
```

View info – GET - /api/users/:id



```
app.get("/api/users/:id", (req, res) => {
    async function get_users() {
        let json_user = []
        const users = collections.find({
            _id: ObjectId(req.params.id)
        })
        await users.forEach(function (result) {
            json_user.push(result)
            console.log(result)
        })
        res.json(json_user)
        json_user = []
    }
    get_users()
})
```