

# Laboratory Activity # 4: Mongoose, MongoDB, CRUD

---

## Setting up

- Folder named "vigo\_api\_starter" was created.
- Express, nodemon, mongoose and MongoDB has been installed using npm

```
JS app.js > ...  
1  
2 const express = require('express');  
3 const mongoose = require('mongoose');  
4 const app = express();  
5 app.use(express.json());  
6  
  
// Connect to MongoDB  
mongoose  
  .connect('mongodb+srv://LazyVigo204:Lazy123204@vigoapi.osxrp6a.mongodb.net/?  
    retryWrites=true&w=majority')  
  .then(()=>{  
    console.log('Connected to MongoDB')  
  })  
  .catch((err)=>{  
    console.error('Connection Failed!',err);  
  })
```

Image above shows a connection to MongoDB database hosted on MongoDB Atlas. I've created my own database rather than copying on what is in the module. Just exploring. But I'm not getting far to the main point or target. Installation of mongoose and MongoDB and CRUD operation should be considered and demonstrate.

```
//define schema  
const UserSchema = mongoose.Schema({  
  lastname: String,    "lastname": Unknown word.  
  firstname: String,   "firstname": Unknown word.  
  gender : String,  
  birthday: Date,  
});
```

```
//creating a model using the created schema
const User = mongoose.model('User', UserSchema);
```

```
//route to display this text
app.get('/',(req,res)=>{
  res.send('User Profile API!');
})
```

```
//route to display user profiles
app.get('/users', async (req,res)=>{
  try{
    const user = await User.find();
    res.send(user);
  }catch (err) {
    res.status(500).json({message:err.message})
  }
})
```

```
43
44 //route to find and sort the user using lastname and its gender
45 app.get('/users/:lastname/:gender', async (req, res) => {
46   const { lastname, gender } = req.params;
47   try {
48     const users = await User.find(
49       {lastname:lastname,gender:gender},
50       {firstname:1, lastname:1},
51     ).sort({firstname:1,lastname:1}).limit(10);
52
53     if (users.length === 0) {
54       return res.status(404).send('User not found!')
55     }
56     res.status(200).json(users);
57   } catch (err) {
58     res.status(404).json({ message: err.message });
59   }
60 });
61
```

```
//route to create new user profile
app.post('/users', async (req, res) => {
  const user = new User({
    lastname: req.body.lastname, "lastname": Unknown word.
    firstname: req.body.firstname, "firstname": Unknown word.
    gender: req.body.gender,
    birthday: req.body.birthday
  });

  try {
    const newUser = await user.save();
    res.status(201).json(newUser);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
});
```

```
//port 3000
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```