



[Course](#) > [Workin...](#) > [Modul...](#) > Modul...

Module 4 Assignment Lab: REST API with Mongoose

The assignment lab for this module is to build a REST API using Mongoose. Unlike the tutorial labs, there will be no step by step instructions but all of the information you need to know should be in the modules. Please attempt to solve the assignment lab on your own before viewing the solution.

REST API with Mongoose

In the previous module, you've built a CRUD RESTful with Express and the native MongoDB Node.js driver (mongodb on npm). However, as you saw in Module 4 Mongoose offers a lot of useful features and benefits compared to the native MongoDB driver. Let's use Mongoose as a database library and ODM for a server.

In this lab, you'll implement a persistent Express REST API server with MongoDB and the Mongoose native driver.

Create a new file `server.js` which will have the code for the CRUD API which will have four endpoints:

1. GET `/accounts`
2. POST `/accounts`
3. PUT `/accounts/:id`
4. DELETE `/accounts/:id`

Use `save()`, `remove()`, `find()` and `findById()` Mongoose methods to save, remove and find documents. Define the account schema as having two fields:

```
name: String,  
balance: Number
```

For the Express server, use `morgan` for request logging, `errorhandler` for error handling and `body-parser` for parsing of payloads.

When you are done, be sure to test the endpoints!

You can test your endpoints with the following curl requests:

//posts account data

```
curl -H "Content-Type: application/json" -X POST -d '{"balance": "1000", "name": "savings"}'
"http://localhost:3000/accounts"
```

//gets account data

```
curl "http://localhost:3000/accounts"
```

//updates account data at specific id, NOTE: replace 'id' in
"http://localhost:3000/accounts/id" with the id generated by the previous POST command

```
curl -H "Content-Type: application/json" -X PUT -d '{"balance": "1500"}'
"http://localhost:3000/accounts/id"
```

//deletes account data at specific id, NOTE: replace 'id' in
"http://localhost:3000/accounts/id" with the id generated by the previous POST command

```
curl -X DELETE "http://localhost:3000/accounts/id"
```

Submission Instructions

To submit the assignment for feedback, put all the program files into GitHub and post a link to your code repository in the Assignment 4 Submissions section of the forums.

In addition to providing the GitHub link, please also answer the following questions about your project:

1. Walk us through the design of your project. Why did you design your project the way you did? What difficulties did you overcome?





2. How did you test your project to verify that it works? If you used any specific curl requests, let us know.
3. Let us know if anything doesn't work as intended so your reviewers can know ahead of time
- A link to the forums submissions section is provided below:

Assignment 4 Submissions

Hide Discussion

Topic: Assignment 4 Submissions / REST API with Mongoose

Add a Post

Show all posts ▼	by recent activity ▼
 NodeJs-Mongoose-APIRest https://github.com/RodrigoGironi/mongoose It´s need the test, I'm work for best in the project, oK? ...	2 ▼
 Assignment 4 Submissions https://github.com/kalamanman/assignment4Mongoose https://github.com/kalamanman/assignment4Mongoose I f you can review my solution to assignmen...	1 ▼
 Assignment 4 https://github.com/muhammad-asad-26/Introduction-to-NodeJS-Module4-Lab/blob/master/app.js	1 ▼
 Module 4 submission Here's a link to my github: https://github.com/browns112358/IntroToNodejs/tree/master/crud_mong...	1 ▼

Feedback Instructions

Please provide feedback to your fellow students!

Please comment on the following:

1. Does the code work correctly? Download the code from the GitHub link and then run the server. Next, test the server by sending curl requests to it to see if the MongoDB database is updating correctly. There are several curl request examples listed at the bottom of the assignment instructions. Make sure to run 'npm install' to download all the required dependencies. Also make sure you have a MongoDB instance running to hold data.

2. How does the code quality look? Does it make sense? Would you do anything different? What did you like about the code design?

Sample Feedback:

1. The code works as intended. I entered the following curl requests (curl ...) and verified that I can create, read, update and delete data.
2. The code had bad styling and the submitter kept reusing sections of code over and over. I would have separated certain parts of the code into a separate function.

Feedback Honor Agreement

1 point possible (graded)

Please review at least 2 assignment submissions in the forums. Try to review ones that haven't been reviewed yet. Once you have done so you can indicate that you have completed the assignment feedback by answer the question below.

I have reviewed at least 2 assignment submissions.

☐ I do not intend to review any assignments

☐ I have reviewed at least 2 assignment submissions

Submit

[Learn About Verified Certificates](#)

© All Rights Reserved