

Employee REST API

Environment set up

Open a terminal and navigate to project folder to run the following

1. Install all the npm modules from package.json
 - a. npm i
2. Set up environment variable
 - a. Set port to 3000
 - i. Set PORT=3000
 - b. Set separate debuggers for levels debug and error in the app
 - i. Set DEBUG=app:*
 - c. Set NODE_ENV to development to look into the Morgan enabled cross cutting logging. Set it to production if you don't wish to see the logging.
 - i. Set NODE_ENV=development

Project info: <http://localhost:3000/readme.pdf>

1. Git hub project location for
 - a. nodejs backend
 - i. <https://github.com/smukherjee16/nodejs>
 - b. Standalone frontend
 - i. <https://github.com/smukherjee16/nodejs/tree/master/employee>
2. Project file
 - a. Index.js – main file with all the employees endpoint implementations
 - b. validations/validations.js – validations rules for request params
 - c. tests/index.test.js – unit tests for mostly validations
 - d. routes/emp.js – Tried to set up a route for the employee's endpoints but running into issues. So left the implementations in index.js for now.
 - e. Routes/home.js – this is a basic root endpoint route set up that works
 - f. Employee/ - All the front end files for the stand alone ui.
 - i. References:
 1. <https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=c-rud-data-table-for-database-with-modal-form>
 2. <https://github.com/uxsolutions/bootstrap-datepicker>

Technologies/Tools

Tools: Visual Studio Code, Postman

Technologies: Node.js, HTML, JavaScript, CSS, RESTful Webservices

Testing Stats and Code Coverage

Run tests: npm test

```
C:\NodeJs>npm test

> npmdemo@1.0.0 test C:\NodeJs
> jest --watchAll --verbose --coverage
  app:debug Morgan enabled for cross cutting logging in dev environment... +0ms
  app:debug Listening on port 3000... +57ms
PASS tests/index.test.js
  ✓ absolute
  testValidRole
    ✓ Valid roles test: TRUE (4ms)
    ✓ Invalid role test: INVALID FALSE
  testCanAddRole
    ✓ Only one ceo test: Add CEO: FALSE (1ms)
    ✓ Only one ceo test: Add MANAGER: TRUE
  testHireDateFormat
    ✓ Valid Hire Date format: YYYY-MM-DD: TRUE (2ms)
    ✓ Invalid Hire Date format: YYYY-Mon-DD: FALSE
  testHireDateInPast
    ✓ Valid Hire Date in past: 2016-03-20: TRUE (1ms)
    ✓ Invalid Hire Date in pfutureast: 2026-03-20: FALSE
  findEmployeeById
    ✓ Valid id: TRUE (1ms)
    ✓ Valid id: TRUE (1ms)

File                                % Stmts   % Branch   % Funcs   % Lines  Uncovered Line #s
-----
All files                          39.67     29.17     27.59     39.66
NodeJs                             27.66      8.82      10        27.54
  index.js                         27.66      8.82      10        27.54    ... 39,340,341,342
NodeJs/middleware                  28.57     100         0        28.57
  auth.js                          25         100         0         25          2,3,4
  logger.js                        33.33     100         0        33.33          2,3
NodeJs/routes                       80         100         0         80
  home.js                          80         100         0         80          7
NodeJs/validations                 90.32     78.57     100        93.1
  validations.js                   90.32     78.57     100        93.1        51,52
-----

Test Suites: 1 passed, 1 total
Tests:       11 passed, 11 total
Snapshots:  0 total
Time:        3.267s
Ran all test suites.

Watch Usage
  > Press f to run only failed tests.
  > Press o to only run tests related to changed files.
  > Press p to filter by a filename regex pattern.
  > Press t to filter by a test name regex pattern.
  > Press q to quit watch mode.
  > Press Enter to trigger a test run.
```

Code Coverage

All files

38.59% Statements 75/194 29.17% Branches 34/48 27.59% Functions 10/36 38.55% Lines 89/231

Press n or / to go to the next uncovered block, b, p or k for the previous block.

File	Statements	Branches	Functions	Lines
NPMDemo	<div><div></div></div> 26.24%	<div><div></div></div> 37/141	<div><div></div></div> 8.82%	<div><div></div></div> 3/34
NPMDemo/middleware	<div><div></div></div> 28.57%	<div><div></div></div> 2/7	<div><div></div></div> 100%	<div><div></div></div> 0/0
NPMDemo/routes	<div><div></div></div> 80%	<div><div></div></div> 4/5	<div><div></div></div> 100%	<div><div></div></div> 0/1
NPMDemo/validations	<div><div></div></div> 90.32%	<div><div></div></div> 28/31	<div><div></div></div> 78.57%	<div><div></div></div> 11/14

Employee Endpoints

Navigate to project folder in Terminal and run nodemon index.js

Task: Get all employee

Endpoint: `http://localhost:3000/api/employees`

Http Method: GET

Responses:

Success: 200

GET `http://localhost:3000/api/employees` Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (11) Test Results Status: 200 OK Time: 98 ms

Pretty Raw Preview JSON

```
1 [
2   {
3     "id": 1,
4     "firstName": "Mark",
5     "lastName": "Anthony",
6     "role": "VP",
7     "hireDate": "2017-12-10",
8     "favoriteMessage1": "message1",
9     "favoriteMessage2": "message2"
10  },
11  {
12    "id": 2,
13    "firstName": "Dave",
14    "lastName": "Barton",
15    "role": "MANAGER",
16    "hireDate": "2017-10-10",
17    "favoriteMessage1": "message11",
18    "favoriteMessage2": "message22"
19  }
20 ]
```

Manage Employees							+ Add New Employee
Id	First Name	Last Name	Role	Hire Date	Message 1	Message 2	
1	Mark	Anthony	VP	2017-12-10	message1	message2	
2	Dave	Barton	MANAGER	2017-10-10	message11	message22	

Task: Get employee by id

Endpoint: `http://localhost:3000/api/employees/:id`

Http Method: GET

Responses:

Id not found: 404

Success: 200

GET `http://localhost:3000/api/employees/1` Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (11) Test Results Status: 200 OK Time: 56 ms

Pretty Raw Preview JSON

```
1 {
2   "id": 1,
3   "firstName": "Mark",
4   "lastName": "Anthony",
5   "role": "VP",
6   "hireDate": "2017-12-10",
7   "favoriteMessage1": "message1",
8   "favoriteMessage2": "message2"
9 }
```

Task: Add new employee

Endpoint: <http://localhost:3000/api/employees>

Http Method: POST

Responses:

Validation failure: 400

Success: 201

Concerns: I have set up Promises chained to return the 2 favorite messages in the POST API call. But these 2 fields are not showing results. However, if I make a GET call to this new id or I do a GET all request, then the messages show. Need to investigate.

The screenshot displays a REST client interface with two panels. The top panel shows a POST request to `http://localhost:3000/api/employees` with a JSON body. The bottom panel shows the response, which is a 201 status code and a JSON object containing the created employee's details and two favorite messages.

POST Request:

```
POST http://localhost:3000/api/employees
{
  "firstName": "Sayeri",
  "lastName": "Nukherjee",
  "role": "CEO",
  "hireDate": "2017-10-10"
}
```

Response (201 Created):

```
{
  "id": "1566914058422",
  "firstName": "Sayeri",
  "lastName": "Nukherjee",
  "role": "CEO",
  "hireDate": "2017-10-10",
  "favoriteMessage1": "",
  "favoriteMessage2": ""
}
```

The bottom panel also shows a GET request to `http://localhost:3000/api/employees/1566914058422` with a 200 OK status and a response that includes the same employee details and two favorite messages.

```
GET http://localhost:3000/api/employees/1566914058422
{
  "id": "1566914058422",
  "firstName": "Sayeri",
  "lastName": "Nukherjee",
  "role": "CEO",
  "hireDate": "2017-10-10",
  "favoriteMessage1": "[I'm going to get 12 eggs and part of a dead animal. Dealer's choice. Please and thank you.\\"]",
  "favoriteMessage2": "Stress is like a pulse, if you have it you're alive."
}
```

Manage Employees

[+ Add New Employee](#)

Id	First Name	Last Name	Role
1	Mark	Anthony	VP
2	Dave	Barton	MANAGER
1566914058422	Sayari	Mukherjee	CEO

Add New Employee

First Name:

Last Name:

Role:

Hire Date:

Message 2

message2

message22

Stress is like a pulse, if you have it you're alive.

Manage Employees							+ Add New Employee
Id	First Name	Last Name	Role	Hire Date	Message 1	Message 2	
1	Mark	Anthony	VP	2017-12-10	message1	message2	
2	Dave	Barton	MANAGER	2017-10-10	message11	message22	
1566914058422	Sayari	Mukherjee	CEO	2017-10-10	["I'm going to get 12 eggs and part of a dead animal. Dealer's choice. Please and thank you."]	Stress is like a pulse, if you have it you're alive.	
1566914354897	Brian	Adams	MANAGER	2019-08-01	["I love riddles!"]	Stress is like a pulse, if you have it you're alive.	

BAD REQUEST and ERROR HANDLING

Create a new record using a randomly generated value as the unique identifier (i.e. `_id` field). Validate that the following fields are included in the POST body and have the right type/format as posted below:

- `firstName` (String)
- `lastName` (String)
- `hireDate` (YYYY-MM-DD format must be in the past)
- `role` (String) - must be one of the following (case-insensitive):
 - CEO (can only be one of these)
 - VP
 - MANAGER
 - LACKEY

Following validations are done for both POST and PUT

Invalid Role:

POST http://localhost:3000/api/employees

Authorization Headers (2) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "firstName": "Sayari",
3   "lastName": "Mukherjee",
4   "role": "INVALID",
5   "hireDate": "2017-10-10"
6 }
```

Body Cookies Headers (11) Test Results

Status: 400 Bad Request Time: 82 ms

Pretty Raw Preview HTML

```
1 BAD REQUEST: Invalid role
```

First Name/Last Name empty:

POST http://localhost:3000/api/employees

Authorization Headers (2) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "firstName": "",
3   "lastName": "Mukherjee",
4   "role": "VP",
5   "hireDate": "2017-10-10"
6 }
```

Body Cookies Headers (11) Test Results

Status: 400 Bad Request Time: 60 ms

Pretty Raw Preview HTML

```
1 "firstName" is not allowed to be empty
```

Invalid date format:

POST http://localhost:3000/api/employees

Authorization Headers (2) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "firstName": "Test",
3   "lastName": "USER",
4   "role": "VP",
5   "hireDate": "2017-JUL-10"
6 }
```

Body Cookies Headers (11) Test Results

Status: 400 Bad Request Time: 59 ms

Pretty Raw Preview HTML

```
1 BAD REQUEST: Date format is incorrect. Expected format is YYYY-MM-DD
```

Only one CEO:

POST ⌵ http://localhost:3000/api/employees Params Send Save ⌵

Authorization Headers (2) **Body** Pre-request Script Tests Code

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ⌵

```
1 {  
2   "firstName": "Test",  
3   "lastName": "USERA",  
4   "role": "CEO",  
5   "hireDate": "2017-08-10"  
6 }
```

Body Cookies Headers (11) Test Results Status: 400 Bad Request Time: 60 ms

Pretty Raw Preview HTML ⌵ 🔍

1 BAD REQUEST: Can't have more than one employee as the CEO

Task: Update employee by Path Param: id

Endpoint: http://localhost:3000/api/employees/:id

Http Method: PUT

Responses:

Id not found: 404

Validation failure: 400

Success: 200

PUT ⌵ http://localhost:3000/api/employees/1 Params Send Save ⌵

Authorization Headers (2) **Body** Pre-request Script Tests Code

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ⌵

```
1 {  
2   "firstName": "Mark - UPDATED",  
3   "lastName": "Anthony - UPDATED",  
4   "role": "vp",  
5   "hireDate": "2017-12-10"  
6 }
```

Body Cookies Headers (11) Test Results Status: 200 OK Time: 58 ms

Pretty Raw Preview JSON ⌵ 🔍

```
1 {  
2   "id": 1,  
3   "firstName": "Mark - UPDATED",  
4   "lastName": "Anthony - UPDATED",  
5   "role": "vp",  
6   "hireDate": "2017-12-10",  
7   "favoriteMessage1": "message1",  
8   "favoriteMessage2": "message2"  
9 }
```

Invalid Id:

PUT ⌵ http://localhost:3000/api/employees/1000 Params Send Save ⌵

Authorization Headers (2) **Body** Pre-request Script Tests Code

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ⌵

```
1 {  
2   "firstName": "Mark - UPDATED",  
3   "lastName": "Anthony - UPDATED",  
4   "role": "vp",  
5   "hireDate": "2017-12-10"  
6 }
```

Body Cookies Headers (11) Test Results Status: 404 Not Found Time: 64 ms

Pretty Raw Preview HTML ⌵ 🔍

1 Employee by id 1000 not found!

Only one CEO:

PUT ▼ http://localhost:3000/api/employees/1 Params Send Save ▼

Authorization Headers (2) **Body** Pre-request Script Tests Code

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "firstName": "Mark - UPDATED",
3   "lastName": "Anthony - UPDATED",
4   "role": "CEO",
5   "hireDate": "2017-12-10"
6 }
```

Body Cookies Headers (11) Test Results Status: 400 Bad Request Time: 31 ms

Pretty Raw Preview HTML ▼ 🔍

1 BAD REQUEST: Can't have more than one employee as the CEO

Task: Delete employee by Path Param: id

Endpoint: http://localhost:3000/api/employees/:id

Http Method: DELETE

Responses:

Id not found: 404

Success: 204

DELETE ▼ http://localhost:3000/api/employees/1566914354897 Params Send Save ▼

Authorization Headers (2) **Body** Pre-request Script Tests Code

Type No Auth ▼

Body Cookies Headers (9) Test Results Status: 204 No Content Time: 68 ms

Pretty Raw Preview Text ▼ 🔍

1

GET ▼ http://localhost:3000/api/employees/1566914354897 Params Send Save ▼

Authorization Headers (2) **Body** Pre-request Script Tests Code

Type No Auth ▼

Body Cookies Headers (11) Test Results Status: 404 Not Found Time: 62 ms

Pretty Raw Preview HTML ▼ 🔍

1 Employee by id 1566914354897 not found!

Invalid Id:

DELETE ▼ http://localhost:3000/api/employees/1000 Params Send Save ▼

Authorization Headers (2) **Body** Pre-request Script Tests Code

Type No Auth ▼

Body Cookies Headers (11) Test Results Status: 404 Not Found Time: 59 ms

Pretty Raw Preview HTML ▼ 🔍

1 Employee by id 1000 not found!