

Node.js - Hands On



```
$ node -v  
V12.4.0
```

```
$ npm -v  
6.9.2
```

hello-world.js

```
console.log('Hi there!');
```

```
$ node hello-world.js  
Hi there!
```

bmi.service.js

```
class BMIService {  
  static getIndex(weight, height) {  
    return weight / (height * height);  
  }  
}
```

```
module.exports = BMIService;
```

```
const BMIService = require('./BMIService');
```

```
const weight = 75;
```

```
const height = 1.71;
```

```
const decimalDigits = 2;
```

```
const bmi = BMIService  
  .getIndex(weight, height)  
  .toFixed(decimalDigits);
```

```
console.log('BMI is: ' + bmi);
```

```
$ node index.js
```

```
BMI is: 25.65
```

bmi.service.js

```
export class BMIService {  
  static getIndex(weight, height) {  
    return weight / (height * height);  
  }  
}
```



```
import { BMIService } from './bmi.service';
```

```
const weight = 75;
```

```
const height = 1.71;
```

```
const decimalDigits = 2;
```

```
const bmi = BMIService  
  .getIndex(weight, height)  
  .toFixed(decimalDigits);
```

```
console.log('BMI is: ' + bmi);
```

```
$ node --experimental-modules --es-module-specifier-resolution=node index.js  
(node:17840) ExperimentalWarning: The ESM module loader is experimental.  
BMI is: 25.65
```

package.json

```
{  
  "private": true,  
  "type": "module",  
  "scripts": {  
    "start": "node --experimental-modules --es-module-specifier-resolution=node ./index.js"  
  }  
}
```

```
$ npm start
```

```
> @ start c:\dm124\03-modules\modern-js
```

```
> node --experimental-modules --es-module-specifier-resolution=node ./index.js
```

```
(node:12984) ExperimentalWarning: The ESM module loader is experimental.
```

```
BMI is: 25.65
```



index.js

```
console.log( '\x1b[33m%s\x1b[0m', 'Hi!');
```

```
$ node index.js
```

```
Hi!
```



Terminal string styling done right

build **passing** coverage **100%** dependents **44265** downloads **1.48B** unicorn **approved** code style **XO** types **Flow | TypeScript**

bold dim italic underline **inverse** strikethrough black
red green yellow blue magenta cyan white gray bgBlack
bgRed **bgGreen** **bgYellow** **bgBlue** **bgMagenta** **bgCyan** **bgWhite**


```
$ npm init -y
```

```
Wrote to C:\dm124\04-npm\package.json:
```

```
{  
  "name": "04-npm",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC"  
}
```

package.json

```
{  
  "name": "04-npm",  
  "private": true  
}
```

```
$ npm i chalk
npm notice created a lockfile as package-lock.json. You should commit this file.
+ chalk@2.4.2
added 7 packages from 3 contributors and audited 7 packages in 0.781s
found 0 vulnerabilities
```

index.js

```
const chalk = require('chalk');  
  
console.log(chalk.yellow('Hi!'));
```

```
$ node index.js
```

```
Hi!
```

```
$ npm i useful-string  
+ useful-string@0.0.2  
added 1 package from 1 contributor and audited 8 packages in 2.217s  
found 0 vulnerabilities
```

```
const chalk = require('chalk');
const string = require('useful-string');

const names = [
  "Felipe Godinho de Almeida",
  "Francine de Fátima Braga",
  "Gerson Lourenço de Carvalho",
  "Pereira da Conceição"
];

const print = name => console.log(chalk.blue(name));

names
  .map(string.hyphenate)
  .forEach(print);
```

```
$ node index.js
```

```
felipe-godinho-de-almeida
```

```
francine-de-fátima-braga-amorim
```

```
gerson-lourenço-de-carvalho
```

```
pereira-da-conceição
```


How to alternate the colors?

```
$ node index.js
```

felipe-godinho-de-almeida

francine-de-fátima-braga-amorim

gerson-lourenço-de-carvalho

pereira-da-conceição

server.js

```
const http = require('http');

const host = 'localhost';
const port = 3000;

function requestHandler(request, response) {
  console.log(request.url);
  response.end('Hi there again!');
}

http.createServer(requestHandler)
  .listen(port, () => {
    console.log(`Server up on http://${host}:${port}`);
  });
```

```
$ node server.js
```

```
Server up on http://localhost:3000
```

GET

http://localhost:3000/products

Send

Save

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Cookies

Code

Comments (0)

Query Params

KEY

VALUE

DESCRIPTION

...

Bulk Edit

Key

Value

Description

Body

Cookies

Headers (3)

Test Results

Status: 200 OK

Time: 8ms

Size: 115 B

Save Response

Pretty

Raw

Preview

Text



1 Hi there again!

How to response an HTML file?

```
$ node server.js
```

Server up on <http://localhost:3000>



Express 4.17.1

Fast, unopinionated,
minimalist web framework for
Node.js

Front-end

TODO List

What needs to be done?

- ☐ Express
- ☐ Node.js
- ☐ REST API
- ☒ ~~JavaScript~~

Client Application

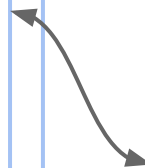
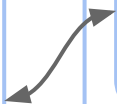
Back-end

REST APIs

Business
Logic



Database



```
$ npm init -y
```

```
Wrote to \t160-nodejs-restful\package.json:
```

```
{  
  "name": "t160-nodejs-restful",  
  "private": true,  
  "scripts": {  
    "start": "",  
    "start:dev": ""  
  }  
}
```



```
$ npm install express
```

```
$ npm install morgan
```

```
$ npm install @types/express --save-dev
```

```
$ npm install @types/node --save-dev
```

app.js

```
const express = require('express');
const app = express();

app.use((request, response, next) => {
  response.status(200).json({
    message: 'It works baby!'
  });
});

module.exports = app;
```

server.js

```
const http = require('http');
const app = require('./app');

const host = process.env.HOST || 'localhost';
const port = process.env.PORT || 8080;

http.createServer(app)
  .listen(port, () => {
    console.log(`Server up on http://${host}:${port}`);
  });
```

```
$ node server.js
```

```
Server up on http://localhost:8080
```

GET

http://localhost:8080/api/tasks

Send

Save

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests ●

Cookies

Code

Comments (0)

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (6)

Test Results (1/2)

Status: 200 OK

Time: 17ms

Size: 240 B

Save Response ▼

Pretty

Raw

Preview

JSON ▼



```
1 {  
2   "message": "It works baby!"  
3 }
```


tasks.js

```
const express = require('express');
const router = express.Router();

router.post('/', (request, response) => {
  response.status(200).json({
    message: 'Handling POST requests to /tasks'
  });
});

module.exports = router;
```

app.js

```
const express = require('express');  
const app = express();
```

```
const taskRoutes = require('./api/routes/tasks');  
app.use('/api/tasks', taskRoutes);
```

```
module.exports = app;
```

```
$ npm install nodemon --save-dev
```

package.json

```
{  
  "name": "t160-nodejs-restful",  
  "private": true,  
  "scripts": {  
    "start": "node server",  
    "start:dev": "nodemon server"  
  },  
  "dependencies": { ... },  
  "devDependencies": {  
    "@types/express": "^4.17.0",  
    "@types/node": "^12.7.2",  
    "nodemon": "^1.19.1"  
  }  
}
```

```
$ npm run start:dev
```

POST

http://localhost:3000/products/

Send

Save

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Cookies

Code

Comments (0)

Query Params

KEY

VALUE

DESCRIPTION

...

Bulk Edit

Key

Value

Description

Body

Cookies

Headers (6)

Test Results

Status: 200 OK

Time: 10ms

Size: 261 B

Save Response

Pretty

Raw

Preview

JSON



1

2

3

```
"message": "Handling POST requests to /products"
```

...

```
router.get('/', (request, response) => {  
  response.status(200).json({  
    message: 'Handling GET requests to /tasks'  
  });  
});
```

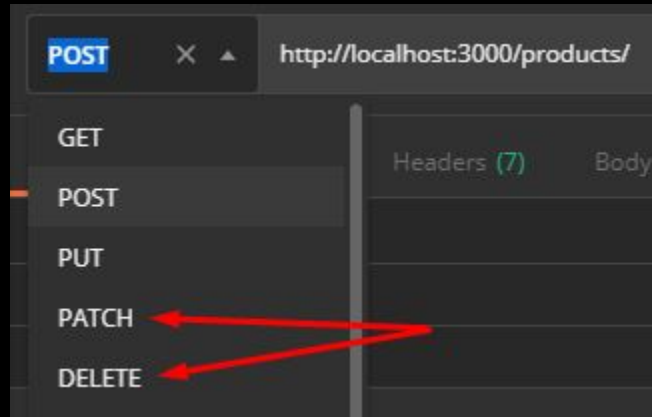
...

...

```
router.get('/:taskId', (request, response) => {  
  const id = request.params.taskId;  
  response.status(200).json({  
    message: 'Task with ID = ${id} was fetched'  
  });  
});
```

...

How to add other HTTP Verbs?



GET

http://localhost:8080/api/hobbit

Send

Save

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Cookies

Code

Comments (0)

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (7)

Test Results

Status: 404 Not Found

Time: 9ms

Size: 398 B

Save Response

Pretty

Raw

Preview

HTML



```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="utf-8">
6    <title>Error</title>
7  </head>
8
9  <body>
10   <pre>Cannot GET /api/hobbit</pre>
11 </body>
12
13 </html>
```

not-found.js

```
const notFound = (request, response, next) => {  
  response.status(404).json({  
    message: `Route to ${request.method} ${request.url} not found`  
  });  
}  
  
module.exports = notFound;
```

app.js

```
const express = require('express');
const morgan = require('morgan');

const app = express();
app.use(morgan('dev'));
app.use(express.json());

app.use('/api/tasks', require('./api/routes/tasks'));
app.use(require('./api/utils/not-found'));

module.exports = app;
```

GET

http://localhost:8080/api/hobbit

Send

Save

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Cookies

Code

Comments (0)

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (6)

Test Results

Status: 404 Not Found

Time: 31ms

Size: 267 B

Save Response

Pretty

Raw

Preview

JSON



```
1 {  
2   "message": "Route to GET /api/hobbit not found"  
3 }
```

check-auth.js

```
const checkAuth = (request, response, next) => {  
  const token = request.headers.authorization;  
  if (/aW5hdGVsOmFsdW5vc2luYXR1bA==/.test(token)) {  
    next();  
  } else {  
    response.status(401).json({  
      message: 'Not authorized'  
    });  
  }  
};  
  
module.exports = checkAuth;
```

...

```
router.post('/', checkAuth, (request, response) => {  
  const task = {  
    done: request.body.done,  
    description: request.body.description  
  }  
  response.status(200).json({  
    message: 'Task was created',  
    task  
  });  
});
```

...

► Add Item

Examples (0) ▼

POST ▼

http://localhost:8080/api/tasks

Send ▼

Save ▼

Params

Authorization

Headers (8)

Body ●

Pre-request Script

Tests ●

Cookies Code Comments (0)

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

● GraphQL ^{BETA}

JSON (application/json) ▼

Beautify

```
1 {  
2   "done": false,  
3   "description": "Learning RxJS"  
4 }
```

Body

Cookies

Headers (6)

Test Results (0/2)

Status: 401 Unauthorized

Time: 36ms

Size: 250 B

Save Response ▼

Pretty

Raw

Preview

JSON ▼



```
1 {  
2   "message": "Not authorized"  
3 }
```


POST

http://localhost:8080/api/tasks

Send

Save

Params

Authorization ●

Headers (8)

Body ●

Pre-request Script

Tests ●

Cookies Code Comments (0)

TYPE

Basic Auth

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Preview Request



Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables](#)



Username

inatel

Password

alunosinatel



Show Password

...

tasks.js

```
let db = {};
```

```
let sequence = 0;
```

```
router.post('/', checkAuth, (request, response) => {
```

```
  const newTask = {
```

```
    id: ++sequence,
```

```
    done: request.body.done || false,
```

```
    description: request.body.description
```

```
  }
```

```
  db[newTask.id] = newTask;
```

```
  response.status(201).json(newTask);
```

```
});
```

...

...

```
router.get('/', (request, response) => {  
  const toArray = key => db[key];  
  const tasks = Object.keys(db).map(toArray);  
  
  tasks.length  
    ? response.json(tasks)  
    : response.status(204);  
});
```

...

...

```
router.get('/:taskId', (request, response) => {  
  const task = db[request.params.taskId];  
  task  
    ? response.json(task)  
    : notFound(request, response)  
});
```

...

tasks.js

...

```
router.patch('/:taskId', checkAuth, (request, response) => {  
  const task = db[request.params.taskId];  
  if (task) {  
    task.done = request.body.done || task.done;  
    task.description = request.body.description || task.description;  
    response.json(task);  
  } else {  
    notFound(request, response);  
  }  
});
```

...

...

```
router.delete('/:taskId', checkAuth, (request, response) => {  
  const task = db[request.params.taskId];  
  if(task) {  
    delete db[task.id];  
    response.status(200).json({  
      message: 'Deleted'  
    });  
  } else {  
    notFound(request, response);  
  }  
});
```

...

nest

A progressive Node.js framework for building efficient, reliable and scalable server-side applications.

