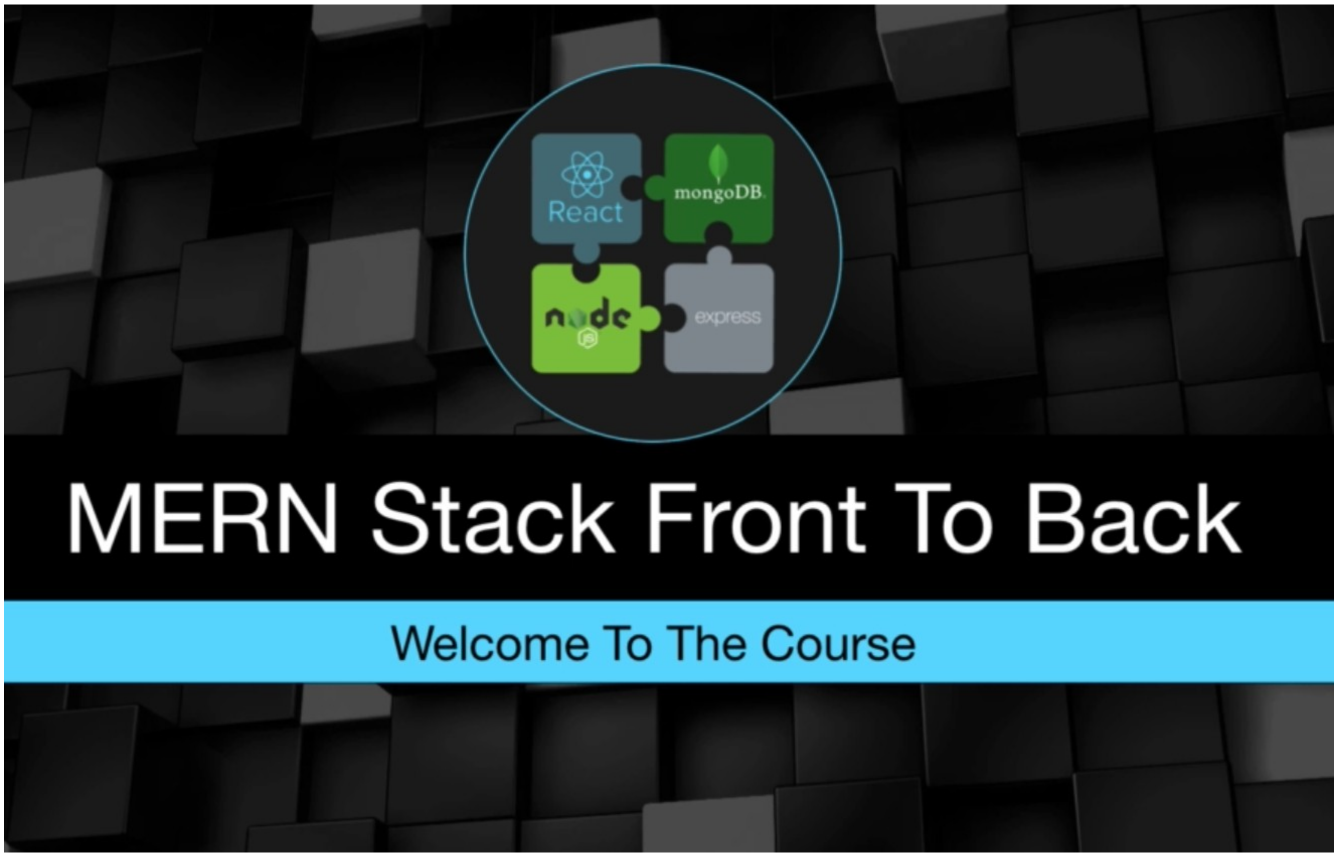
MERN Stack Front To Back: Full Stack React, Redux & Node.js (2020)

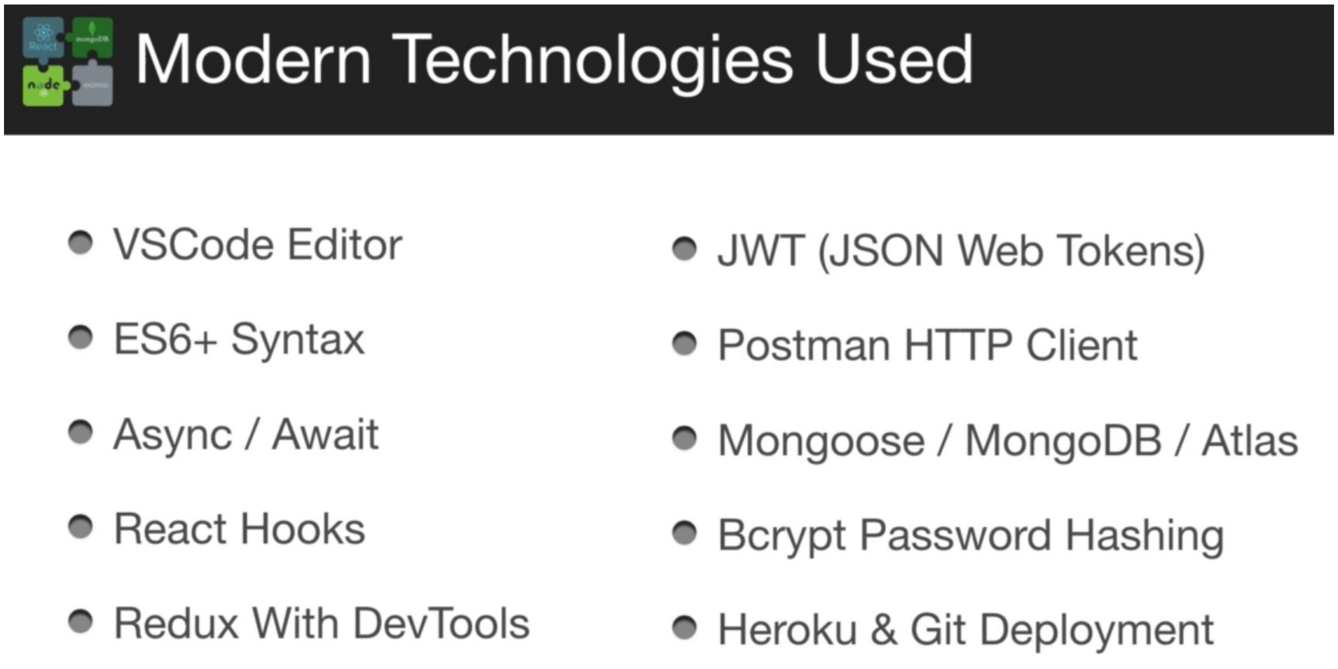


https://www.udemy.com/course/mern-stack-front-to-back/

by Brad Traversy

Table of Contents

0. Introduction


Modern Technologies Used

- VSCode Editor
- ES6+ Syntax
- Async / Await
- React Hooks
- Redux With DevTools
- JWT (JSON Web Tokens)
- Postman HTTP Client
- Mongoose / MongoDB / Atlas
- Bcrypt Password Hashing
- Heroku & Git Deployment

0.1 Environment & Setup

0.1.1 Node.js

Windows: https://nodejs.org/en/ - just download and install with GUI.

Linux:
CLI=> *sudo apt-get update*
CLI=> *sudo apt install curl build-essential*
CLI=> *curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -*
CLI=> *sudo apt install -y nodejs*
CLI=> *node -v [=> v14.16.1]*

0.1.2 Visual Studio Code (VSC)

Windows: https://code.visualstudio.com/ - just download and install with GUI.

Linux:
CLI=> *sudo apt-get update*
CLI=> *sudo apt install code*

0.1.3 GIT

Windows: https://git-scm.com/ just download and install with GUI.

Linux: **GIT is a basic component**

0.1.4 Postman

Windows: https://www.postman.com/ - just download and install with GUI.

Linux: *https://www.postman.com/downloads/ - download archive tar.gz*
https://dl.pstmn.io/download/latest/linux64

go to download folder and unpack that file with command *tar xvzf [**PACKAGENAME**].tar.gz*

CLI=> *tar xvzf Postman-linux-x64-8.6.1.tar.gz*

go to folder and use shortcut

0.1.5 React Developer Tools chrome extension

0.1.6 Redux DevTools chrome extension

0.1.7 VSC extensions

- Bracket Pair Colorizer - https://marketplace.visualstudio.com/items?itemName=CoenraadS.bracket-pair-colorizer

- ES7 React/Redux/GraphQL/React-Native snippets - https://marketplace.visualstudio.com/items?itemName=dsznajder.es7-react-js-snippets

- Prettier - Code formatter - https://marketplace.visualstudio.com/items?itemName=esbenp.prettier-vscode

VSC=>Manage=>Settings=> “format on save” should be enable

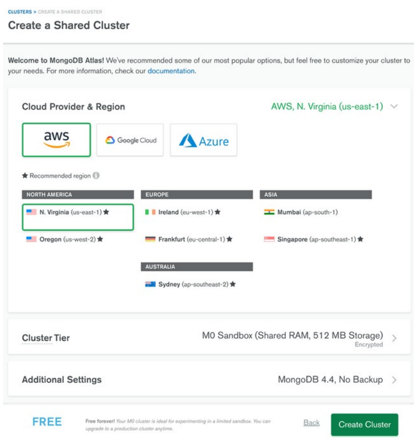
1. Express & MongoDB Setup

1.1 MongoDB (create a new cluster for project)

Create an account at https://www.mongodb.com/. Sign in to an account.

Create a new project – **MERN-devconnector**

Create a new cluster - **Cluster0**



CLUSTERS > CREATE A SHARED CLUSTER

Create a Shared Cluster

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our documentation.

Cloud Provider & Region: AWS, N. Virginia (us-east-1)

★ Recommended region

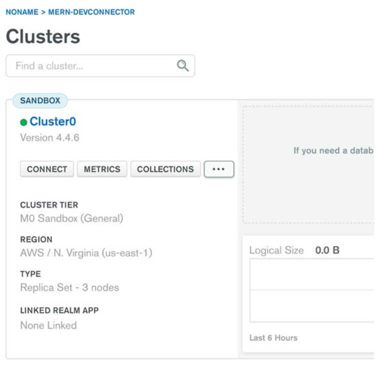
NORTH AMERICA	EUROPE	ASIA
N. Virginia (us-east-1) ★	Ireland (eu-west-1) ★	Mumbai (ap-south-1)
Oregon (us-west-2) ★	Frankfurt (eu-central-1) ★	Singapore (ap-southeast-1) ★
	AUSTRALIA	
	Sydney (ap-southeast-2) ★	

Cluster Tier: M0 Sandbox (Shared RAM, 512 MB Storage) Encrypted

Additional Settings: MongoDB 4.4, No Backup

FREE — **Free forever!** *Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.*

Push button “CONNECT” and add your IP adress and create a new database user to connect to a DB.



NONAME > MERN-DEVCONNECTOR

Clusters

SANDBOX

Cluster0

Version 4.4.6

CLUSTER TIER: M0 Sandbox (General)

REGION: AWS / N. Virginia (us-east-1)

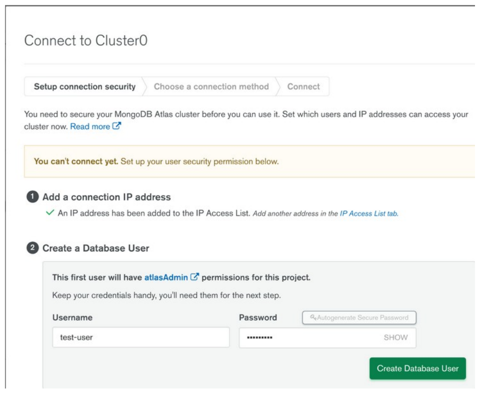
TYPE: Replica Set - 3 nodes

LINKED REALM APP: None Linked

If you need a datab

Logical Size 0.0 B

Last 6 Hours



Connect to Cluster0

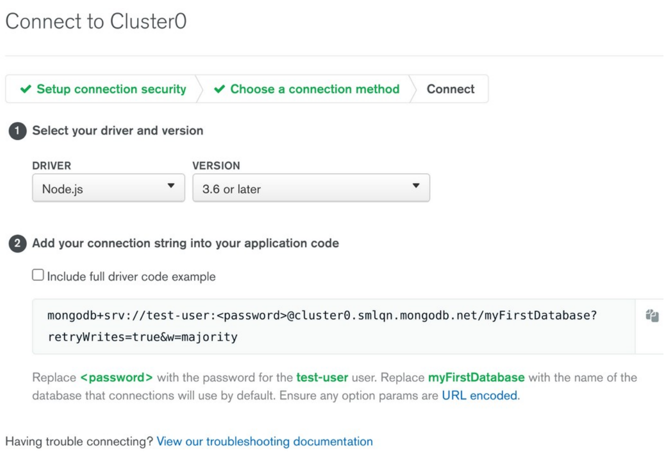
Setup connection security › Choose a connection method › Connect

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. Read more

You can't connect yet. Set up your user security permission below.

1. Add a connection IP address
   ✓ An IP address has been added to the IP Access List. *Add another address in the IP Access List tab.*
2. Create a Database User
   This first user will have atlasAdmin permissions for this project.
   Keep your credentials handy, you'll need them for the next step.
   Username: test-user
   Password: ••••••••••

Push button “Choose a connection method” and “connect your application” and save that link



Connect to Cluster0

✓ Setup connection security › ✓ Choose a connection method › Connect

1. Select your driver and version
   DRIVER: Node.js
   VERSION: 3.6 or later
2. Add your connection string into your application code
   ☐ Include full driver code example

```
mongodb+srv://test-user:<password>@cluster0.smlqn.mongodb.net/myFirstDatabase?
retryWrites=true&w=majority
```

Replace **<password>** with the password for the **test-user** user. Replace **myFirstDatabase** with the name of the database that connections will use by default. Ensure any option params are URL encoded.

Having trouble connecting? View our troubleshooting documentation

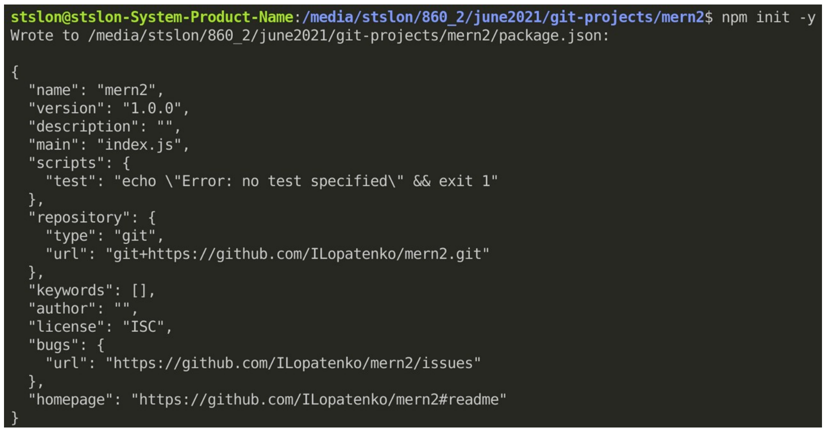
1.1 Express (install a package and setup a server)

1.1.1 Create a .gitignore file

It is a file with list of files/folders witch will be ignored by GIT. Add **node_modules** folder

1.1.2 Initialize an NMP project and setup entry point

CLI=> *npm init -y*



```
stslon@stslon-System-Product-Name:/media/stslon/860_2/june2021/git-projects/mern2$ npm init -y
Wrote to /media/stslon/860_2/june2021/git-projects/mern2/package.json:

{
  "name": "mern2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/ILopatenko/mern2.git"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/ILopatenko/mern2/issues"
  },
  "homepage": "https://github.com/ILopatenko/mern2#readme"
}
```

This command creates a new file package.json with basic information about project and all the dependencies. I’m going to change entry point (“main”:“server.js”) for this project to **server.js**

1.1.3 Install all the packages as regular dependencies

I’am going to use in this project next packages:

- **express** – backend server;

- **express-validator** – for validation data;

- **bcryptjs** – for making hash for passwords;

- **config** – for making global variables;

- **gravatar** – for working with user’s avatars;

- **jsonwebtoken** – for working with JWT;

- **mongoose** – for working with mongoDB database;

- **request** – for working with another API based services;

CLI=> *npm install* ***express express bcryptjs config gravatar jsonwebtoken mongoose request***

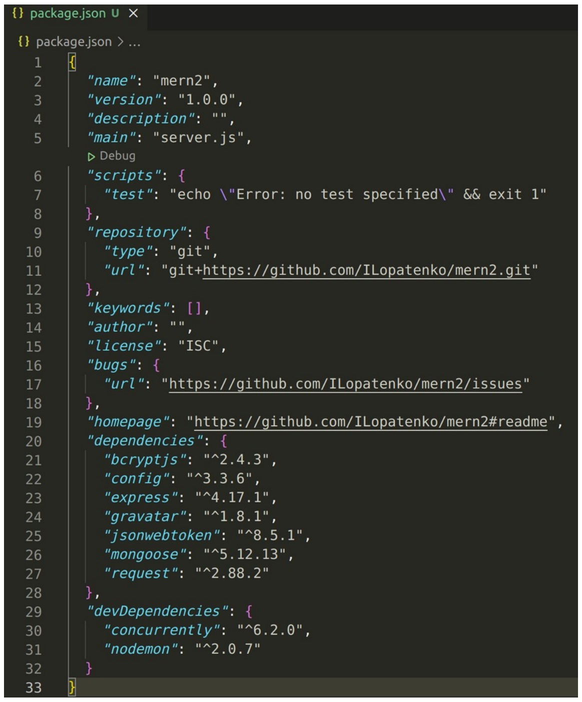
1.1.4 Install all the packages as DEV dependencies

DEV dependencies will be installed without direct reference to a project (npm will add them to package.json file as a devDependencies)

- **nodemon –** will track all the changes and make a restart server automatically;

- **concurrently –** allows me to run few scripts at the same time with a single command;

CLI=> *npm install* ***-D nodemon concurrently***

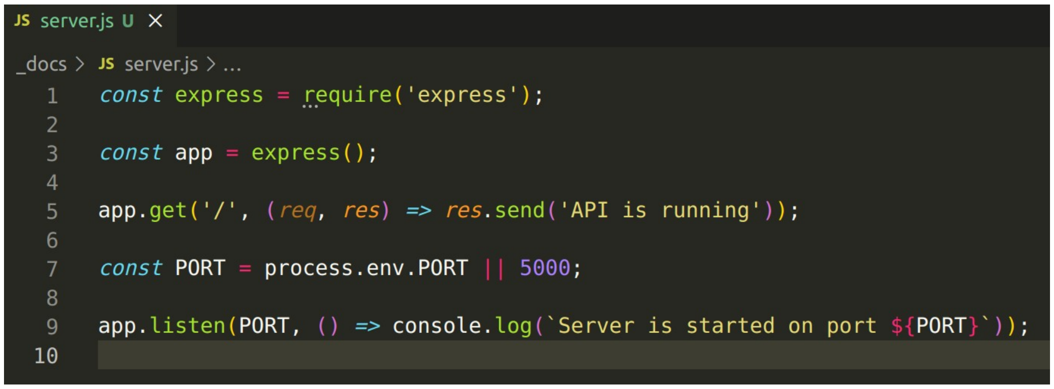


package.json U

package.json > ...

```json
{
  "name": "mern2",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/ILopatenko/mern2.git"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/ILopatenko/mern2/issues"
  },
  "homepage": "https://github.com/ILopatenko/mern2#readme",
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "config": "^3.3.6",
    "express": "^4.17.1",
    "gravatar": "^1.8.1",
    "jsonwebtoken": "^8.5.1",
    "mongoose": "^5.12.13",
    "request": "^2.88.2"
  },
  "devDependencies": {
    "concurrently": "^6.2.0",
    "nodemon": "^2.0.7"
  }
}
```

1.1.5 Create an express server

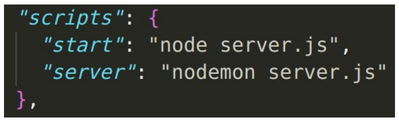


server.js U

_docs > server.js > ...

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

const app = express();

app.get('/', (req, res) => res.send('API is running'));

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => console.log(`Server is started on port ${PORT}`));
```

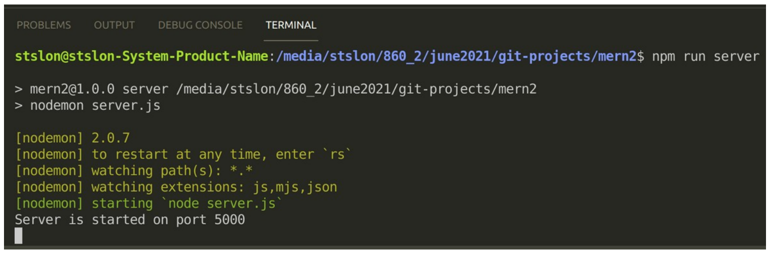
1.1.6 Change run script at package.json



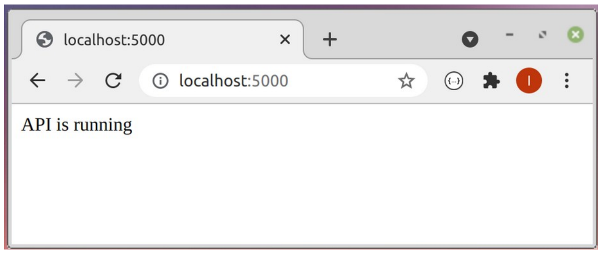
```
"scripts": {
  "start": "node server.js",
  "server": "nodemon server.js"
},
```

Now command “npm run start” will run server.js with node.js and command “npm run server” will run server.js with nodemon

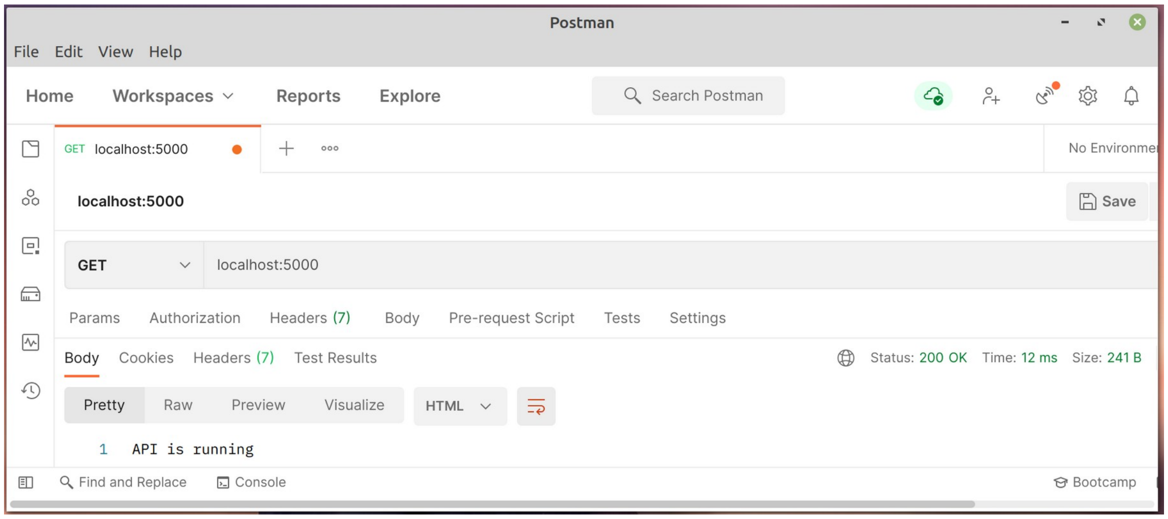
1.1.7 The first run



```
stslon@stslon-System-Product-Name:/media/stslon/860_2/june2021/git-projects/mern2$ npm run server

> mern2@1.0.0 server /media/stslon/860_2/june2021/git-projects/mern2
> nodemon server.js

[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
Server is started on port 5000
```



API is running



GET localhost:5000

Status: 200 OK Time: 12 ms Size: 241 B

```
API is running
```

What was used in project?

Operating Systems: Linux Mint 20.1 and Windows 10

Node.js – javascript runtime enviroment.

NPM

GIT -

MongoDB

Mongoose

nodemon

postman