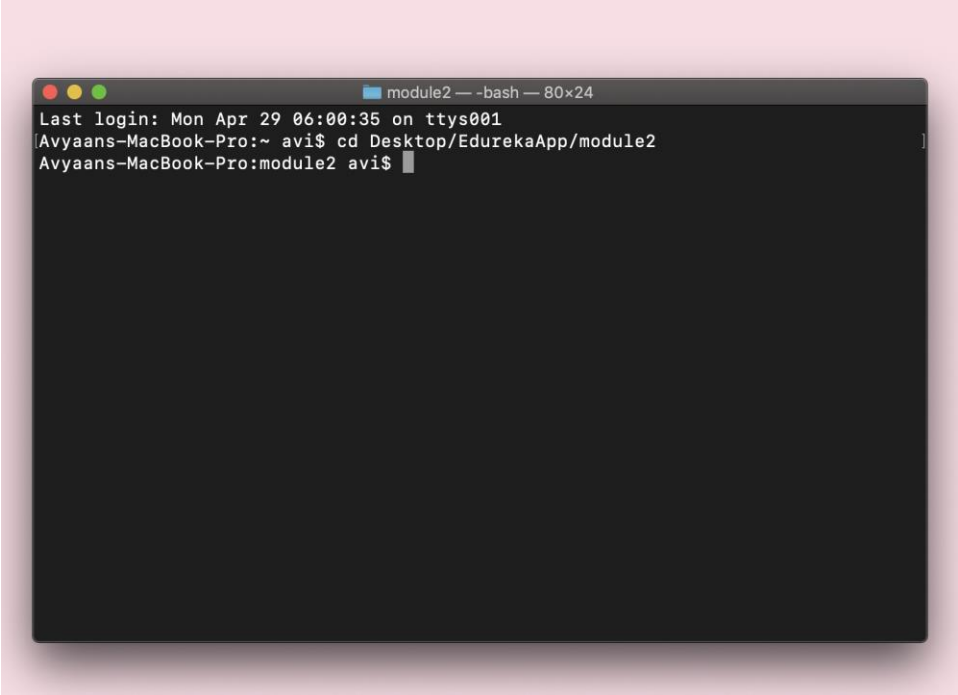
Module 2: File System Module And Express.js

Demo Document 1

edureka!

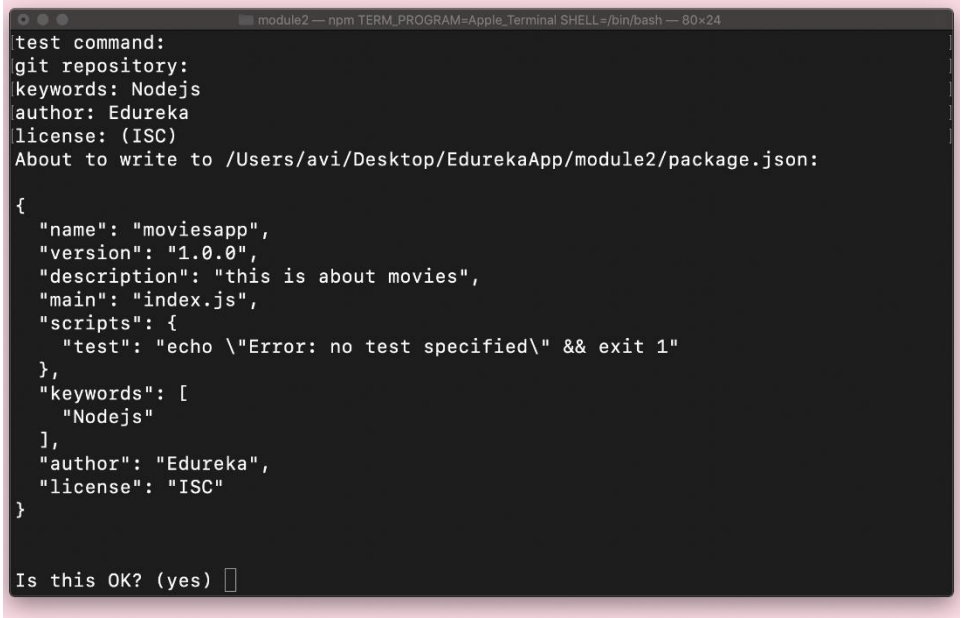
Build an API using express, read file with FS module, and deploy application using PM2 and Nginx

Step 1: Create new folder in system and navigate with terminal or command prompt.



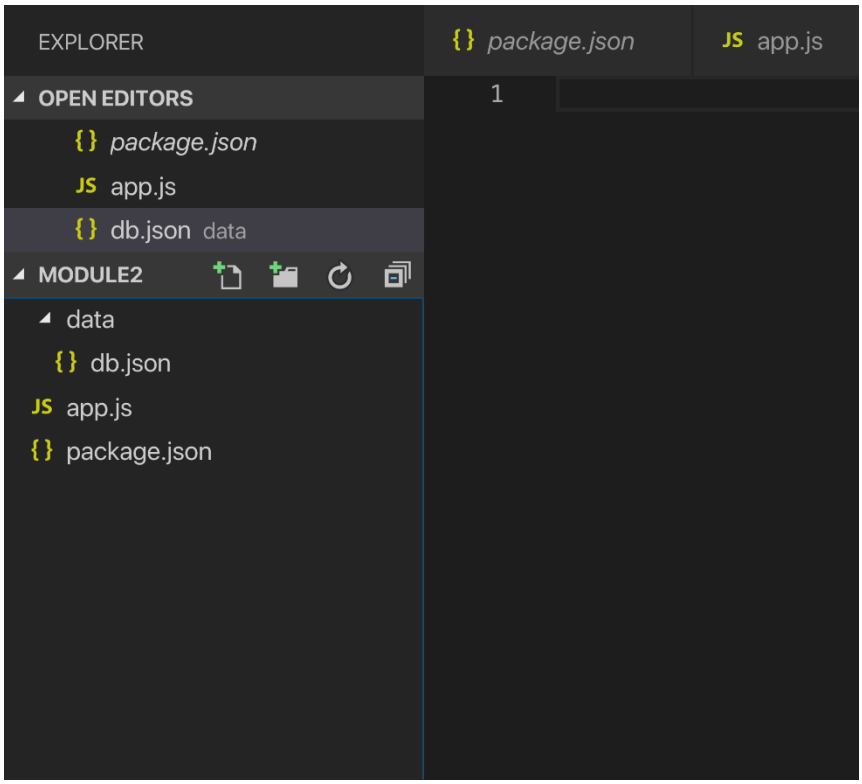
Step 2: Create 'package.json' inside the folder using command prompt with 'npm init'

Command and answer some question regarding app.

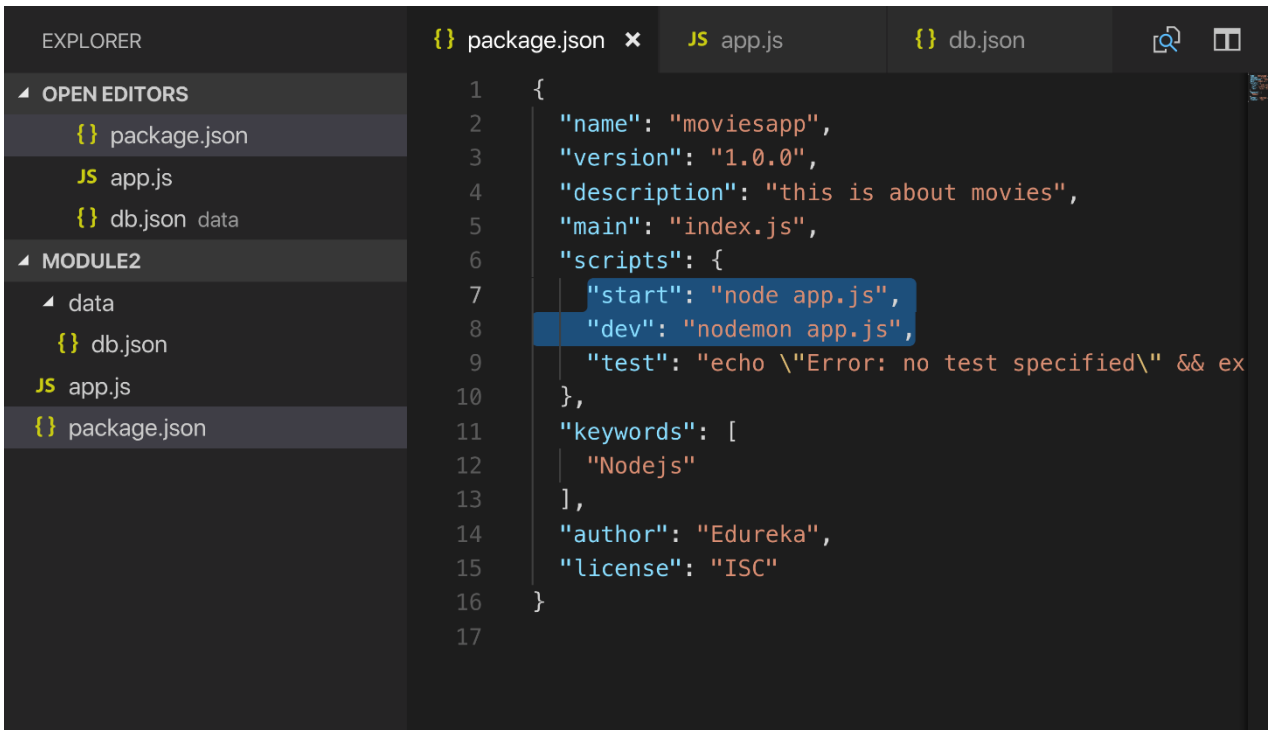


At the end type 'yes' and file with the name of package.json will be created.

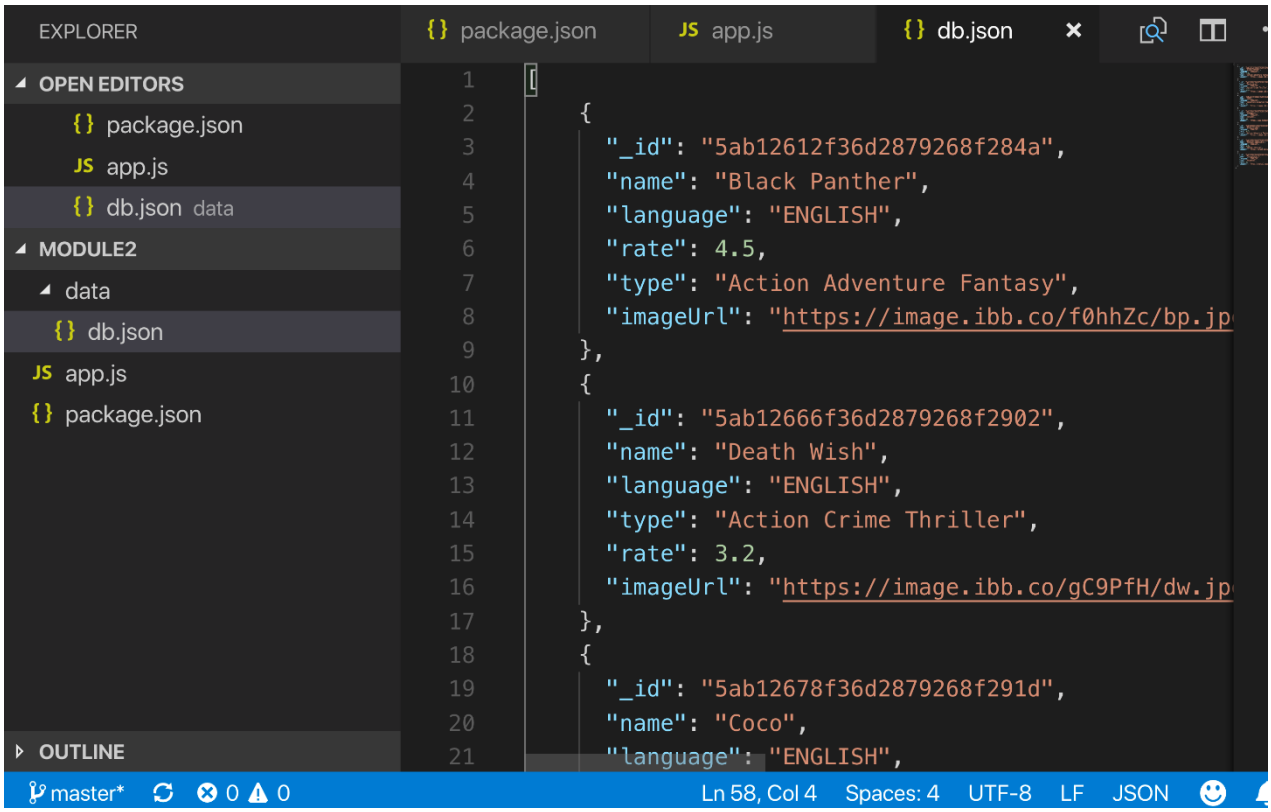
Step 3: Create a folder structure with one db folder to put ‘db.json ‘file with sample data and app.js for server code.



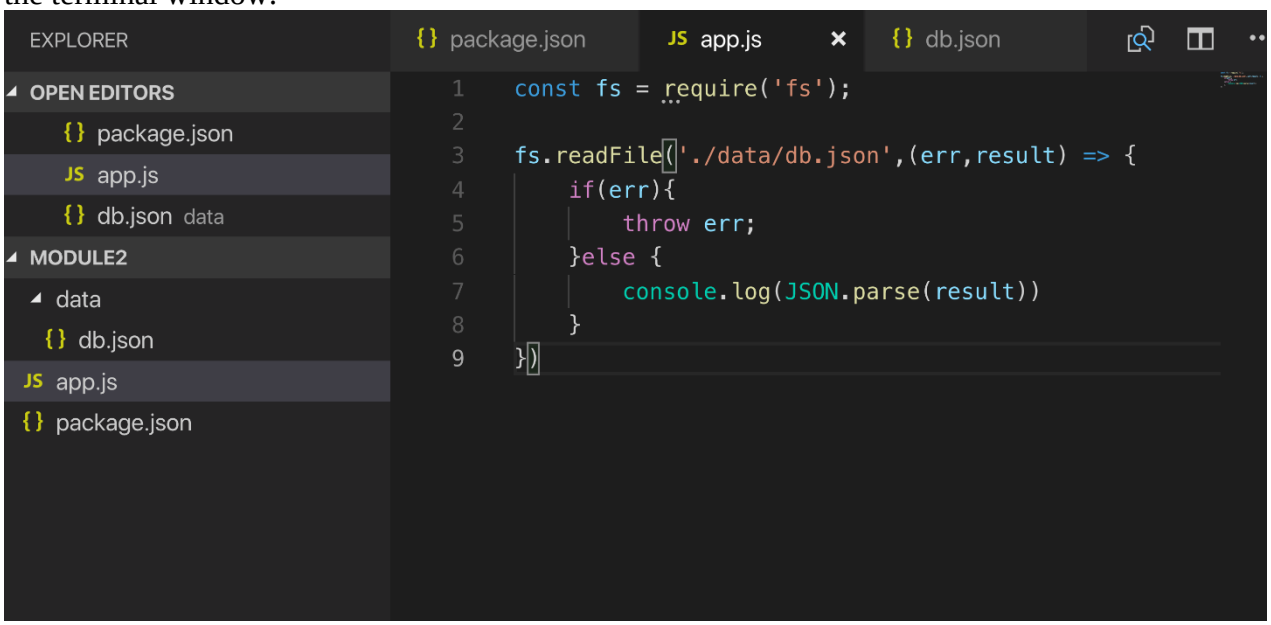
Step 4: Specify start and dev command in package.json to run application on production as well as Developement mode with node <filename>.



Step 5: Copy data into db.json using the db.json file present on LMS.



Step 6: Require Fs module in app.js and use ***fs.readFile*** to read data from db.json and console in the terminal window.



Step 7: Run "npm start" in command prompt to run the application and db.json will be send out in console window.

```
[nodemon] watching: *.*
[nodemon] starting `node app.js`
[ { _id: '5ab12612f36d2879268f284a',
    name: 'Black Panther',
    language: 'ENGLISH',
    rate: 4.5,
    type: 'Action Adventure Fantasy',
    imageUrl: 'https://image.ibb.co/f0hhZc/bp.jpg' },
  { _id: '5ab12666f36d2879268f2902',
    name: 'Death Wish',
    language: 'ENGLISH',
    type: 'Action Crime Thriller',
    rate: 3.2,
    imageUrl: 'https://image.ibb.co/gC9PfH/dw.jpg' },
  { _id: '5ab12678f36d2879268f291d',
    name: 'Coco',
    language: 'ENGLISH',
    type: 'Adventure Animation Family',
    rate: 5,
    imageUrl: 'https://image.ibb.co/dQwWSx/coco.jpg' },
  { _id: '5ab126b6f36d2879268f2943',
    name: 'Avengers',
    language: 'ENGLISH',
    type: 'Actione',
    rate: 2,
    imageUrl:
     'https://www.hindustantimes.com/rf/image_size_960x540/HT/p2/2018/04/01/Pictures/_46a0b2c0-3590-11e8-8c5f-3c6cc031651
  { _id: '5ab4e66b0c1d2b27846c6407',
    name: 'Black Friday',
    language: 'ENGLISH',
    rate: 4.5,
    type: 'Action Adventure Fantasy',
    imageUrl: 'https://image.ibb.co/f0hhZc/bp.jpg' },
  { _id: '5ab12686f36d2879268f2930',
    name: 'Mission Impossible',
    language: 'English',
    rate: 2.5,
    type: 'Horror Thriller',
    imageUrl:
     'https://pre00.deviantart.net/5d3b/th/pre/f/2017/313/2/b/mission_impossible__dark_directive_teaser_poster_by_themadb
  { _id: '5ab12698f36d2879268f293e',
    name: 'Incredibles 2',
    language: 'ENGLISH',
    type: 'Animated',
    rate: 4,
    imageUrl:
     'http://static1.squarespace.com/static/588a4776f5e23132a09d23b2/588a4e91be65945e50a36c0e/5b24084baa4a999c88a9f277/15
[nodemon] clean exit - waiting for changes before restart
```

Step 8: After read operation we will use ***fs.appendFile*** to write in text file in data folder .

fs.appendFile will append new data in same file in case of ***fs.writeFile*** it will override the data.

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

// Read data from file
fs.readFile('./data/db.json',(err,result) => {
    if(err){
        throw err;
    }else {
        console.log(JSON.parse(result))
    }
})

// Write data in text file
fs.appendFile('./data/mytext.txt','My text read file\n',(err) => {
    if(err) throw err;
    else{
        console.log('Data written successfully')
    }
})
```

Step 9: Again, use same command "npm start" or "npm run dev" we will see console data written successfully in node console.

```
Data written successfully
[nodemon] clean exit - waiting for changes before restart
```

Step 10: To start creating server with express first of all install express in same folder with command "npm install express".

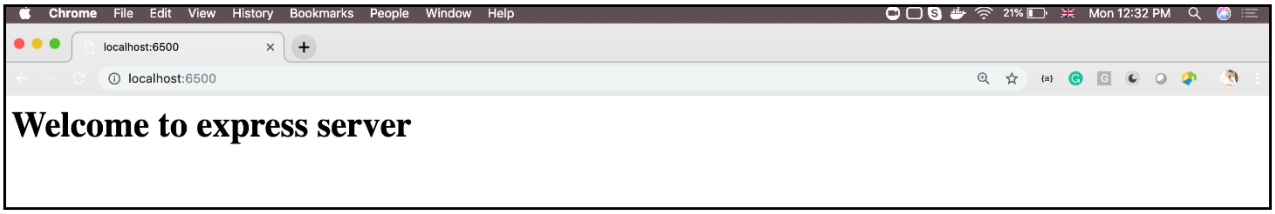
```
module2 — -bash — 83×19
Avyaans-MacBook-Pro:module2 avi$ npm install express
```

Step 11: Once install verify package must be added as dependency in package.json.

Step 12: Just like Fs require express also in app.js and create object. Define default route which will send default message and make server to listen on port provided.

```
const fs = require('fs');
// require express
const express = require('express');
// create object and add port
const app = express();
const port = 6500;

// Define default route with express
app.get('/',(req,res) => {
    res.send('<h1> Welcome to express server</h1>')
})

// Create server to listen on port
app.listen(port,(err) => {
    console.log('server is running on port '+port);
})
```

Step 13: Again run application. To avoid start stop use “dev” command to start application using “npm run dev” command. Open url localhost:6500 on the browser.

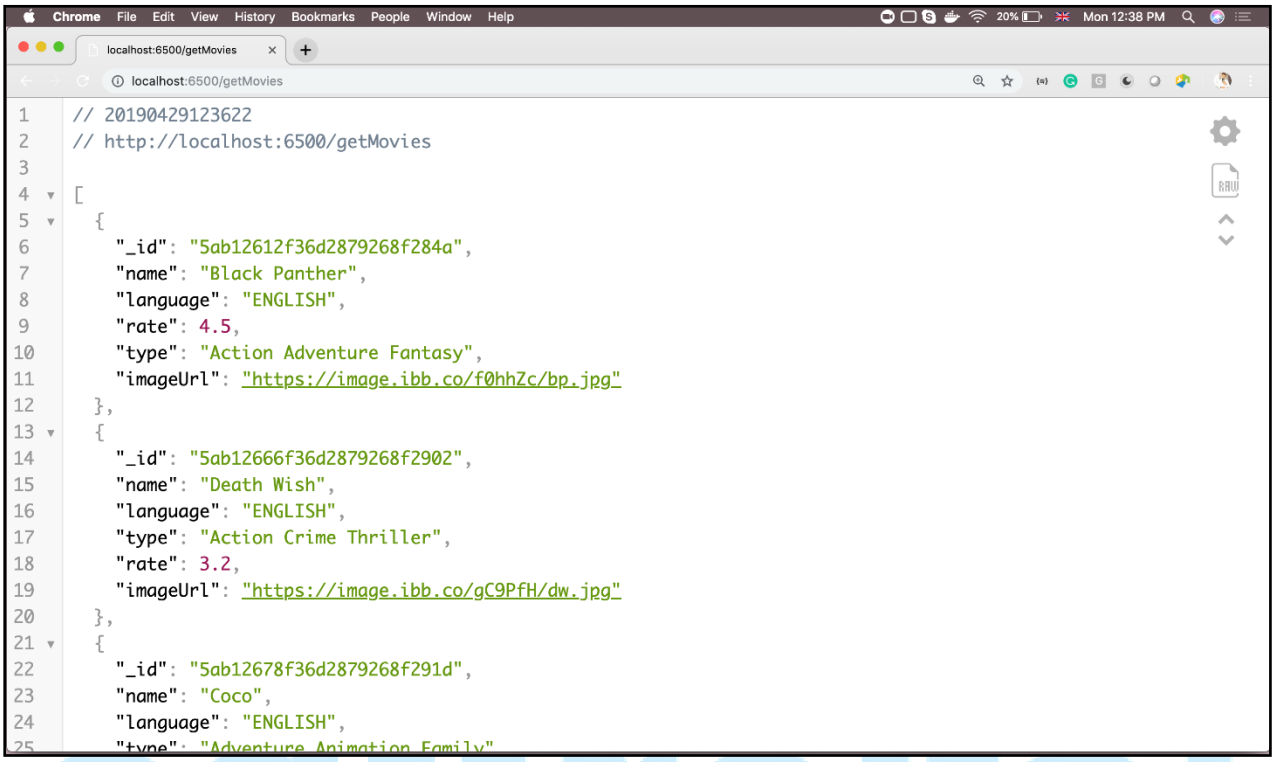


Step 14: Create one more route using express and add fs.readfile code in '/getMovies' route
This will help to read data from file and send as json to the browser.

```
// require express
const express = require('express');
// create object and add port
const app = express();
const port = 6500;

// Define default route with express
app.get('/',(req,res) => {
    res.send('<h1> Welcome to express server</h1>')
})

// Read file using express server
app.get('/getMovies', (req,res) => {
    fs.readFile('./data/db.json',(err,result) => {
        if(err){
            throw err;
        }else {
            res.send(JSON.parse(result))
        }
    })
})

// Create server to listen on port
app.listen(port,(err) => {
    console.log('server is running on port '+port);
```

In case of error it will throw error as well

Step 15: Finally run "http://localhost:6500/getMovies" to see response on the browser.
It will send JSON data saved in file to the browser.



Step 16: Lets run app over Nginx and achieve reverse proxy
Install "pm2" this help to run application to run in background and will keep it always running.

```
Avyaans-MacBook-Pro:code avi$ sudo npm install -g pm2
```

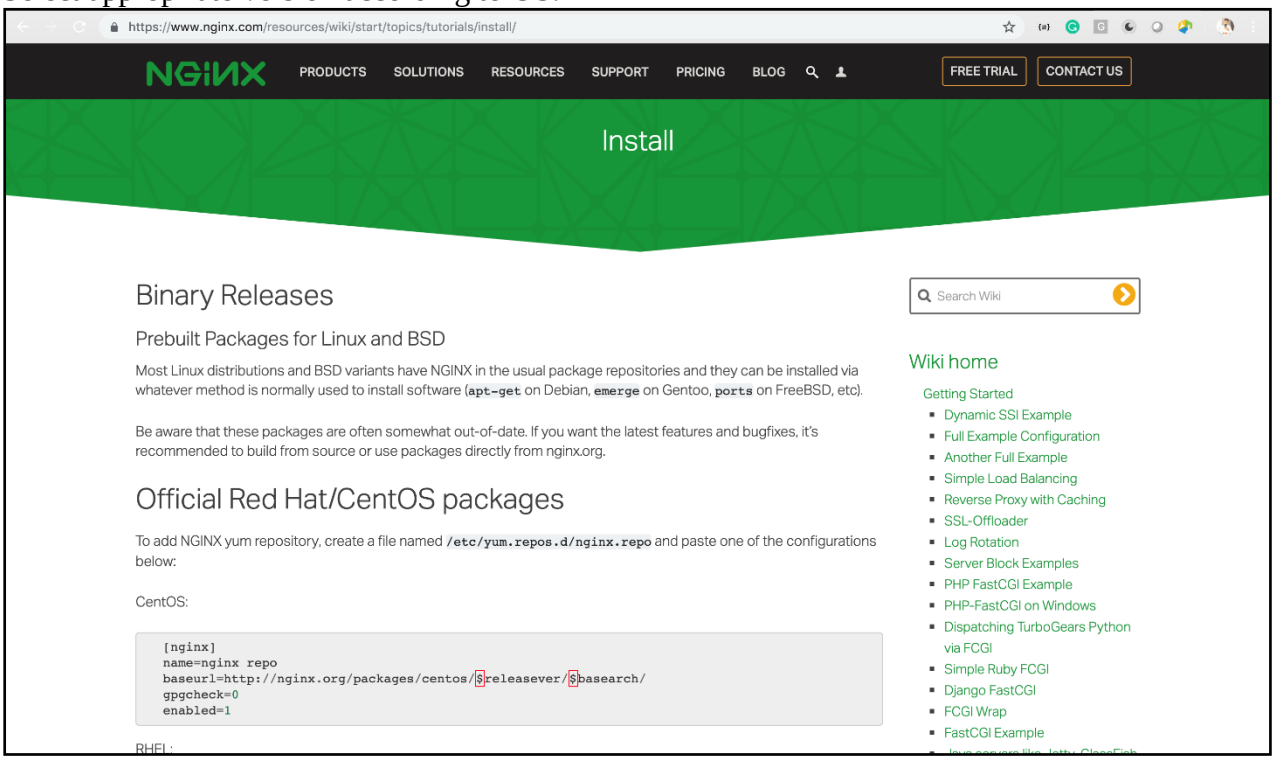
Step 17: Start Application using “pm2” and run in background.
To stop application use command “pm2 stop all”

```
Avyaans-MacBook-Pro:code avi$ pm2 start app.js
[PM2] Starting /Users/avi/Desktop/folder/EdurekaApp/module2/code/app.js in fork_mode (1 instance)
[PM2] Done.
```

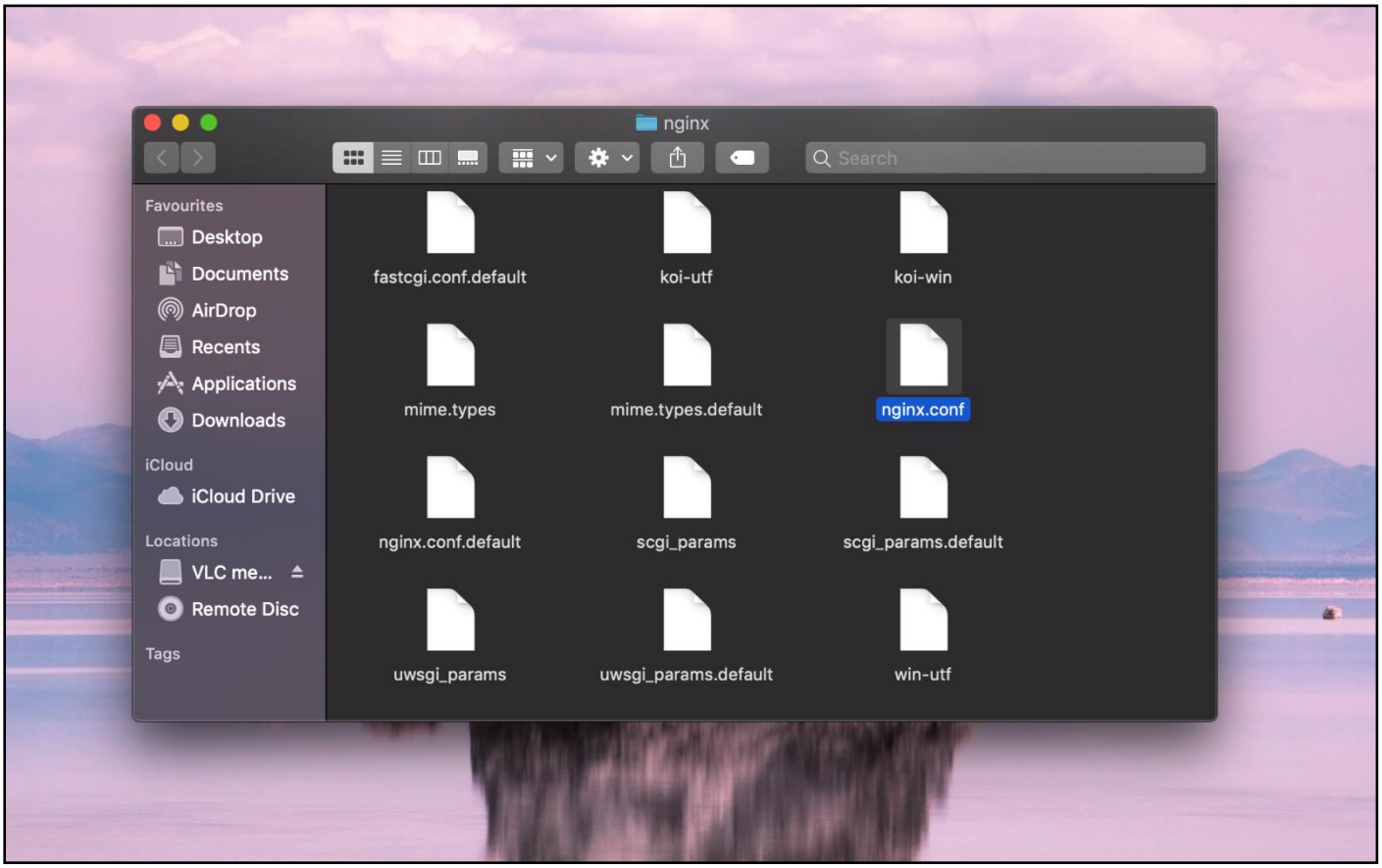
App name	id	version	mode	pid	status	restart	uptime	cpu	mem	user	watching
app	1	1.0.0	fork	3903	online	0	0s	0%	7.0 MB	avi	disabled
start	0	1.0.0	fork	0	stopped	0	0	0%	0 B	avi	disabled

```
 Use `pm2 show <id|name>` to get more details about an app
Avyaans-MacBook-Pro:code avi$
```

Step 18: Install Nginx for on the system
Select appropriate version according to OS.



Step 19: Navigate to Nginx folder in your system and edit nginx.conf file
Path for Mac: /usr/local/etc/nginx
Path for Windows: C:\nginx\conf



Step 20: Edit Nginx.conf file and server key add path of running application over Pm2
And provide path over which you want to run in reverse proxy.

```
        #location ~ \.php$ {
        #    root           html;
        #    fastcgi_pass   127.0.0.1:9000;
        #    fastcgi_index  index.php;
        #    fastcgi_param  SCRIPT_FILENAME  /scripts$fastcgi_script_name;
        #    include        fastcgi_params;
        #}

        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #    deny  all;
        #}
    }


    # another virtual host using mix of IP-, name-, and port-based configuration
    #
    server {
        listen       8081;
        server_name  localhost;

        location / {
           root   /var/www/react-app;
            index  index.html index.htm;
        }
    }

    server {
        listen       8082;
        server_name  localhost;

        location / {
           proxy_pass http://127.0.0.1:6500;
           proxy_http_version 1.1;
           proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection 'upgrade';
            proxy_set_header Host $host;
            proxy_cache_bypass $http_upgrade;
        }
    }



    # HTTPS server
    #
    #server {
    #    listen       443 ssl;
    #    server_name  localhost;

    #    ssl_certificate      cert.pem;
    #    ssl_certificate_key  cert.key;
```

Inside Conf file you can also set SSL configuration.

Step 21: New port for running application is localhost:8082 instead of localhost:6500 reverse proxy id implemented with Nginx