NodeJS and ReactJS Assignments

1. Node JS:

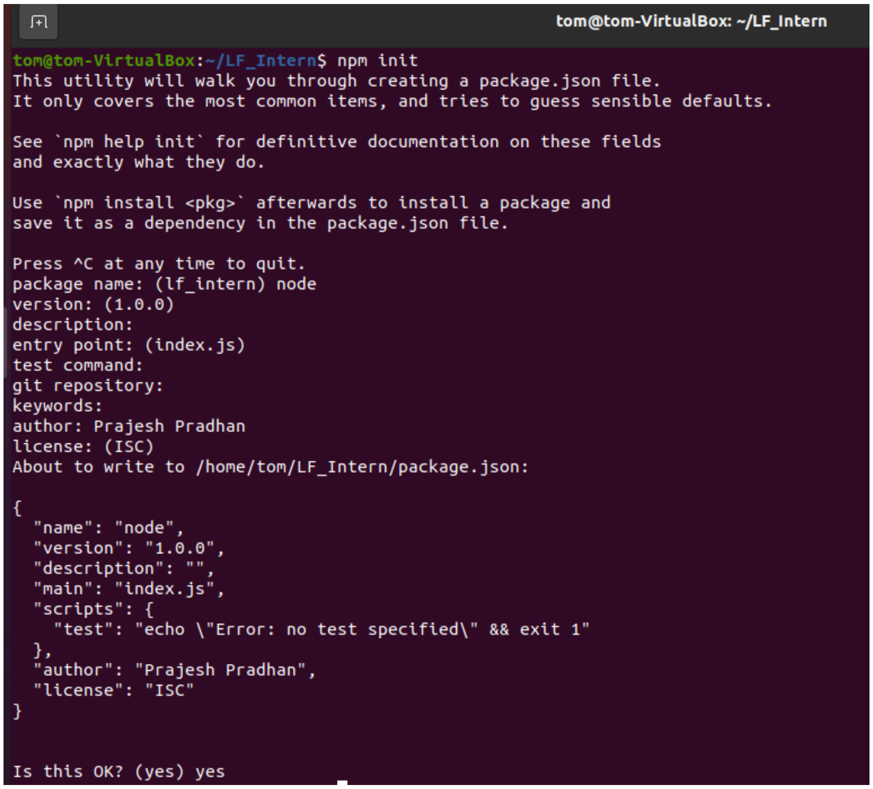
Install Node js on local VM.

```
~ curl -fsSL https://deb.nodesource.com/setup_17.x | sudo -E bash -

~ sudo apt-get install -y nodejs
```

Create 2 API's running on ports 6080 and 7080 with messages "Hello Node JS" and "Node JS installed successfully" respectively.

First Create directory for the apis and initialize the node application

```
~ mkdir lf_intern
~ cd lf_intern
~ npm init
```



```
tom@tom-VirtualBox:~/LF_Intern$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (lf_intern) node
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author: Prajesh Pradhan
license: (ISC)
About to write to /home/tom/LF_Intern/package.json:

{
  "name": "node",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Prajesh Pradhan",
  "license": "ISC"
}


Is this OK? (yes) yes
```

Now to create 2 apis, 2 js files are created, ap1.js and api2.js

api1.js

```
var http = require('http');
http.createServer(function(req,res){
      res.writeHead(200,{'Content-Type':'text/plain'});
      res.end('Hello Node JS');
}).listen(6080);
console.log('Server started on localhost:6080; press Ctrl-C to terminate...!');
```

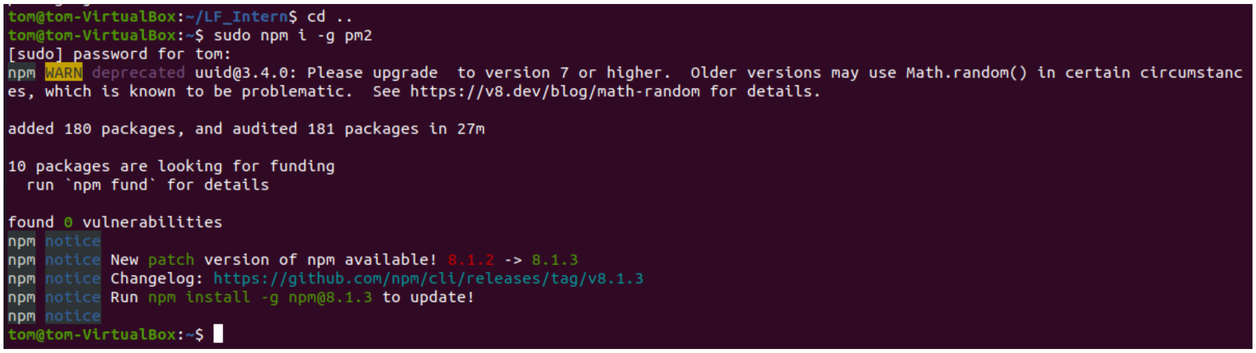
api2.js

```
var http = require('http');
http.createServer(function(req,res){
      res.writeHead(200,{'Content-Type':'text/plain'});
      res.end('Node JS installed seccessfully');
}).listen(7080);
console.log('Server started on localhost:7080; press Ctrl-C to terminate...!');
```

Install pm2 tool and create 4 clusters of both Node's.

To install pm2:

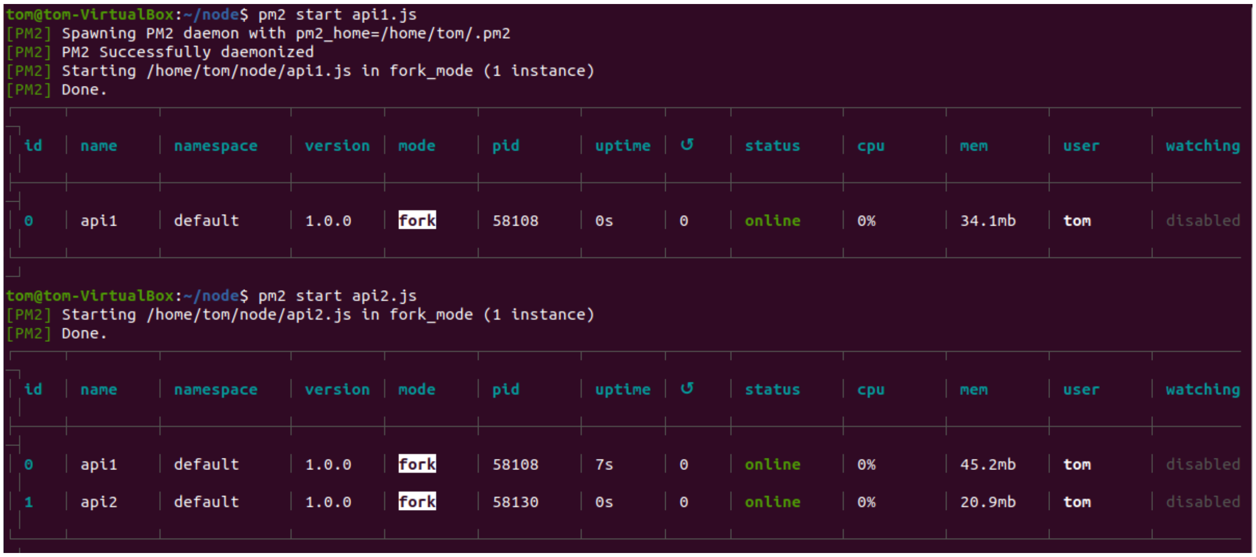
```
~ npm install -g pm2
```



```
tom@tom-VirtualBox:~/LF_Intern$ cd ..
tom@tom-VirtualBox:~$ sudo npm i -g pm2
[sudo] password for tom:
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain circumstanc
es, which is known to be problematic.  See https://v8.dev/blog/math-random for details.

added 180 packages, and audited 181 packages in 27m

10 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New patch version of npm available! 8.1.2 -> 8.1.3
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.1.3
npm notice Run npm install -g npm@8.1.3 to update!
npm notice
tom@tom-VirtualBox:~$ 
```

To start a node application, we can use node or npm, pm2 can also be used as such

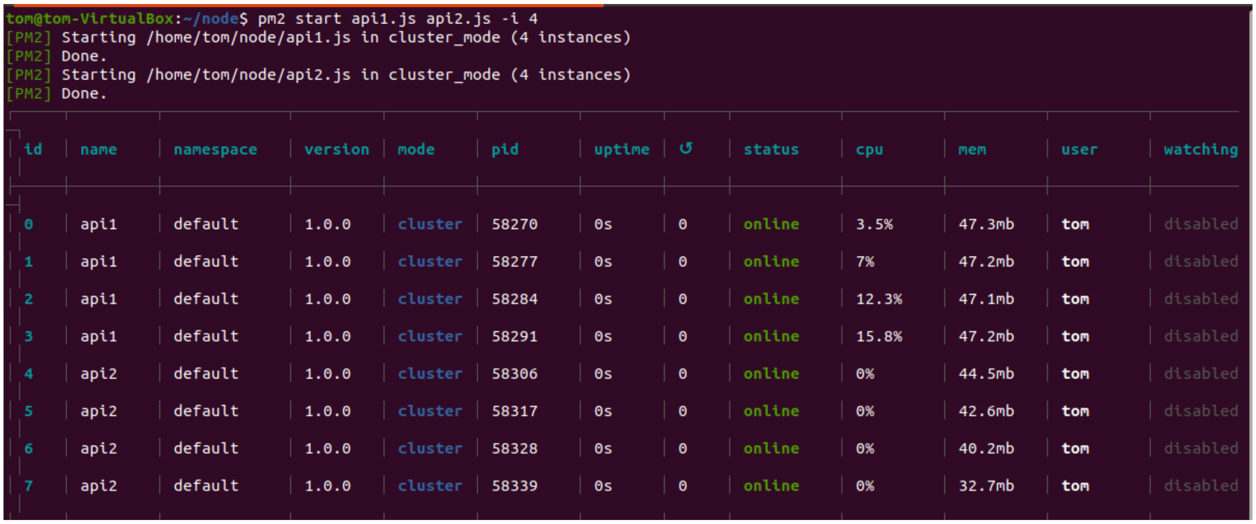
```
~ pm2 start api1.js
```



```
tom@tom-VirtualBox:~/node$ pm2 start api1.js
[PM2] Spawning PM2 daemon with pm2_home=/home/tom/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /home/tom/node/api1.js in fork_mode (1 instance)
[PM2] Done.
│ id │ name │ namespace │ version │ mode │ pid   │ uptime │ ↺ │ status │ cpu │ mem    │ user │ watching │
│ 0  │ api1 │ default   │ 1.0.0   │ fork │ 58108 │ 0s     │ 0 │ online │ 0%  │ 34.1mb │ tom  │ disabled │
tom@tom-VirtualBox:~/node$ pm2 start api2.js
[PM2] Starting /home/tom/node/api2.js in fork_mode (1 instance)
[PM2] Done.
│ id │ name │ namespace │ version │ mode │ pid   │ uptime │ ↺ │ status │ cpu │ mem    │ user │ watching │
│ 0  │ api1 │ default   │ 1.0.0   │ fork │ 58108 │ 7s     │ 0 │ online │ 0%  │ 45.2mb │ tom  │ disabled │
│ 1  │ api2 │ default   │ 1.0.0   │ fork │ 58130 │ 0s     │ 0 │ online │ 0%  │ 20.9mb │ tom  │ disabled │
```

To create cluster of 4 of each of the node applications we can use the command

```
~ pm2 start api1.js api2.js -i 4
```

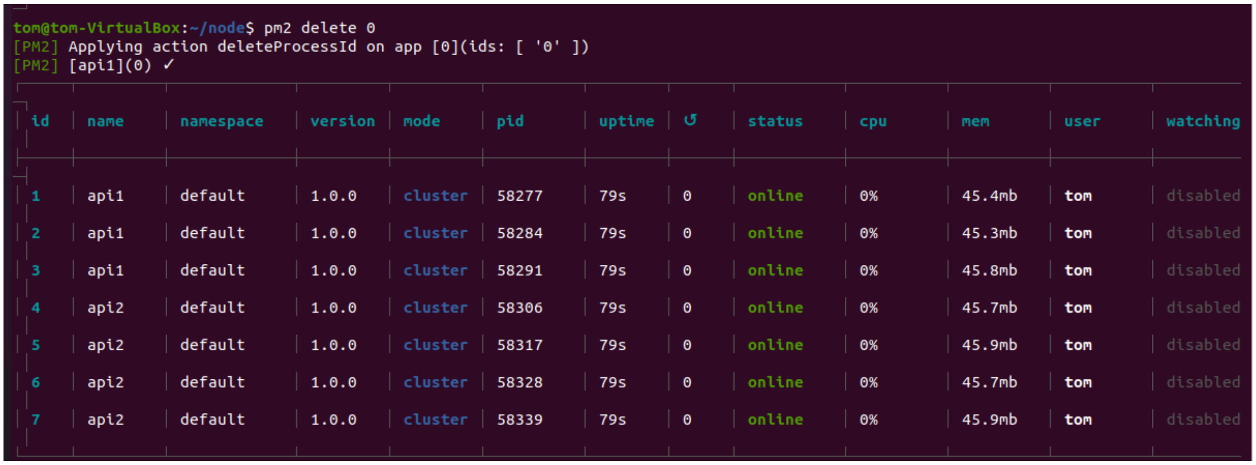


```
tom@tom-VirtualBox:~/node$ pm2 start api1.js api2.js -i 4
[PM2] Starting /home/tom/node/api1.js in cluster_mode (4 instances)
[PM2] Done.
[PM2] Starting /home/tom/node/api2.js in cluster_mode (4 instances)
[PM2] Done.
│ id │ name │ namespace │ version │ mode    │ pid   │ uptime │ ↺ │ status │ cpu   │ mem    │ user │ watching │
│ 0  │ api1 │ default   │ 1.0.0   │ cluster │ 58270 │ 0s     │ 0 │ online │ 3.5%  │ 47.3mb │ tom  │ disabled │
│ 1  │ api1 │ default   │ 1.0.0   │ cluster │ 58277 │ 0s     │ 0 │ online │ 7%    │ 47.2mb │ tom  │ disabled │
│ 2  │ api1 │ default   │ 1.0.0   │ cluster │ 58284 │ 0s     │ 0 │ online │ 12.3% │ 47.1mb │ tom  │ disabled │
│ 3  │ api1 │ default   │ 1.0.0   │ cluster │ 58291 │ 0s     │ 0 │ online │ 15.8% │ 47.2mb │ tom  │ disabled │
│ 4  │ api2 │ default   │ 1.0.0   │ cluster │ 58306 │ 0s     │ 0 │ online │ 0%    │ 44.5mb │ tom  │ disabled │
│ 5  │ api2 │ default   │ 1.0.0   │ cluster │ 58317 │ 0s     │ 0 │ online │ 0%    │ 42.6mb │ tom  │ disabled │
│ 6  │ api2 │ default   │ 1.0.0   │ cluster │ 58328 │ 0s     │ 0 │ online │ 0%    │ 40.2mb │ tom  │ disabled │
│ 7  │ api2 │ default   │ 1.0.0   │ cluster │ 58339 │ 0s     │ 0 │ online │ 0%    │ 32.7mb │ tom  │ disabled │
```

Delete all 4 clusters one-by-one

To delete pm2 instances we can use **delete** command from **pm2** as such:

```
~ pm2 delete 0
```



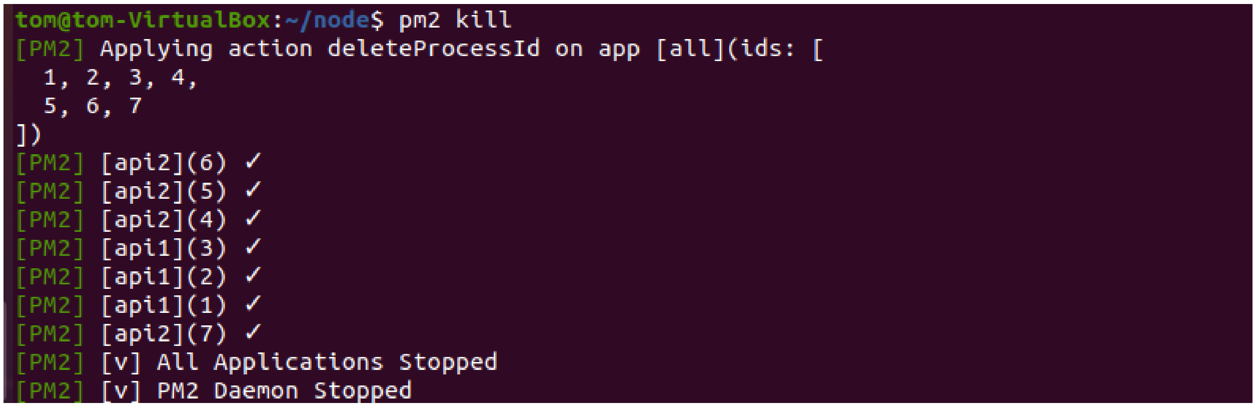
```
tom@tom-VirtualBox:~/node$ pm2 delete 0
[PM2] Applying action deleteProcessId on app [0](ids: [ '0' ])
[PM2] [api1](0) ✓
│ id │ name │ namespace │ version │ mode    │ pid   │ uptime │ ↺ │ status │ cpu │ mem    │ user │ watching │
│ 1  │ api1 │ default   │ 1.0.0   │ cluster │ 58277 │ 79s    │ 0 │ online │ 0%  │ 45.4mb │ tom  │ disabled │
│ 2  │ api1 │ default   │ 1.0.0   │ cluster │ 58284 │ 79s    │ 0 │ online │ 0%  │ 45.3mb │ tom  │ disabled │
│ 3  │ api1 │ default   │ 1.0.0   │ cluster │ 58291 │ 79s    │ 0 │ online │ 0%  │ 45.8mb │ tom  │ disabled │
│ 4  │ api2 │ default   │ 1.0.0   │ cluster │ 58306 │ 79s    │ 0 │ online │ 0%  │ 45.7mb │ tom  │ disabled │
│ 5  │ api2 │ default   │ 1.0.0   │ cluster │ 58317 │ 79s    │ 0 │ online │ 0%  │ 45.9mb │ tom  │ disabled │
│ 6  │ api2 │ default   │ 1.0.0   │ cluster │ 58328 │ 79s    │ 0 │ online │ 0%  │ 45.7mb │ tom  │ disabled │
│ 7  │ api2 │ default   │ 1.0.0   │ cluster │ 58339 │ 79s    │ 0 │ online │ 0%  │ 45.9mb │ tom  │ disabled │
```

In Similar way we can do it for all the processes

```
~ pm2 delete 1
~ pm2 delete 2
~ pm2 delete 3
~ pm2 delete 4
~ pm2 delete 5
~ pm2 delete 6
~ pm2 delete 7
```

We can also **kill** all the process as once as such:

```
~ pm2 kill
```



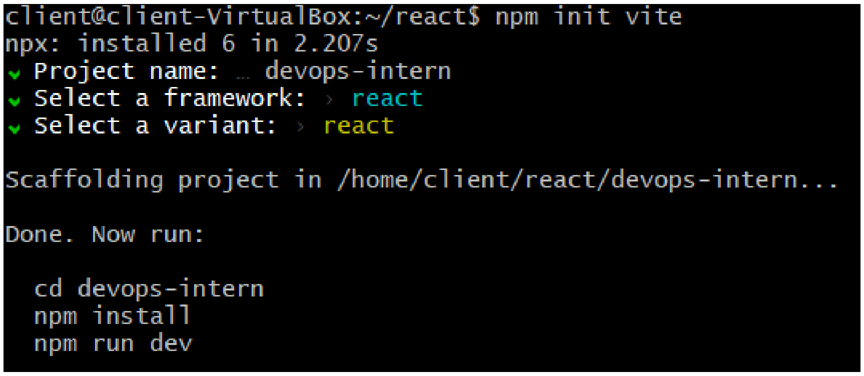
```
tom@tom-VirtualBox:~/node$ pm2 kill
[PM2] Applying action deleteProcessId on app [all](ids: [
  1, 2, 3, 4,
  5, 6, 7
])
[PM2] [api2](6) ✓
[PM2] [api2](5) ✓
[PM2] [api2](4) ✓
[PM2] [api1](3) ✓
[PM2] [api1](2) ✓
[PM2] [api1](1) ✓
[PM2] [api2](7) ✓
[PM2] [v] All Applications Stopped
[PM2] [v] PM2 Daemon Stopped
```

2. React JS:

Install React.js.

To install reactJS we will be using Vite, so first to initialize vite

```
~ npm init vite
```



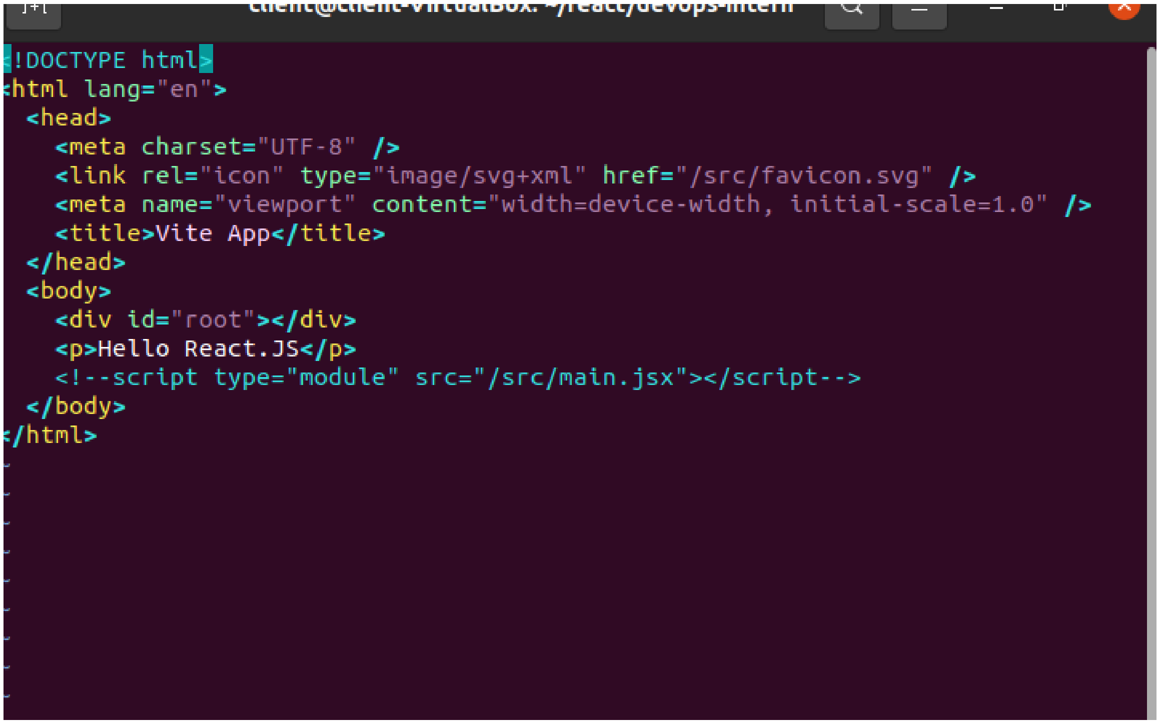
```
client@client-VirtualBox:~/react$ npm init vite
npx: installed 6 in 2.207s
✔ Project name: … devops-intern
✔ Select a framework: › react
✔ Select a variant: › react

Scaffolding project in /home/client/react/devops-intern...

Done. Now run:

  cd devops-intern
  npm install
  npm run dev
```

Creating a React Application and print message "Hello React.js"

Now we need to install the dependencies of the react application which can be done with the command

```
~ npm install
```

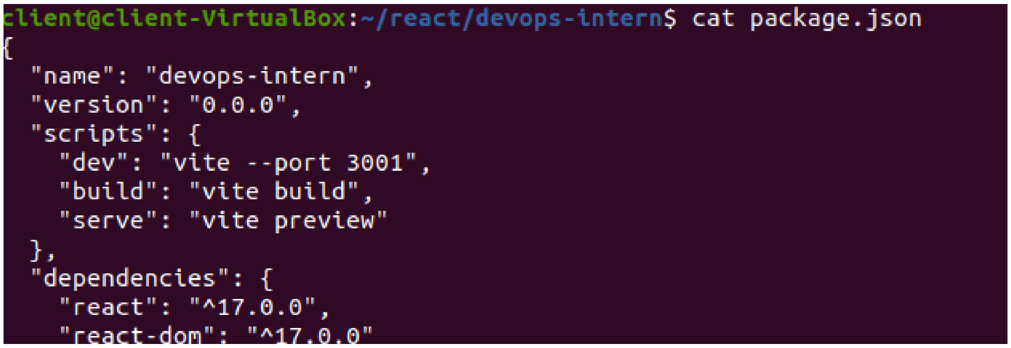
Now to print the message we change **index.html**



```
client@client-VirtualBox: ~/react/devops-intern
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/src/favicon.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Vite App</title>
  </head>
  <body>
    <div id="root"></div>
    <p>Hello React.JS</p>
    <!--script type="module" src="/src/main.jsx"></script-->
  </body>
</html>
~
~
~
~
~
~
~
~
~
```

Change the default port 3000 to 3001

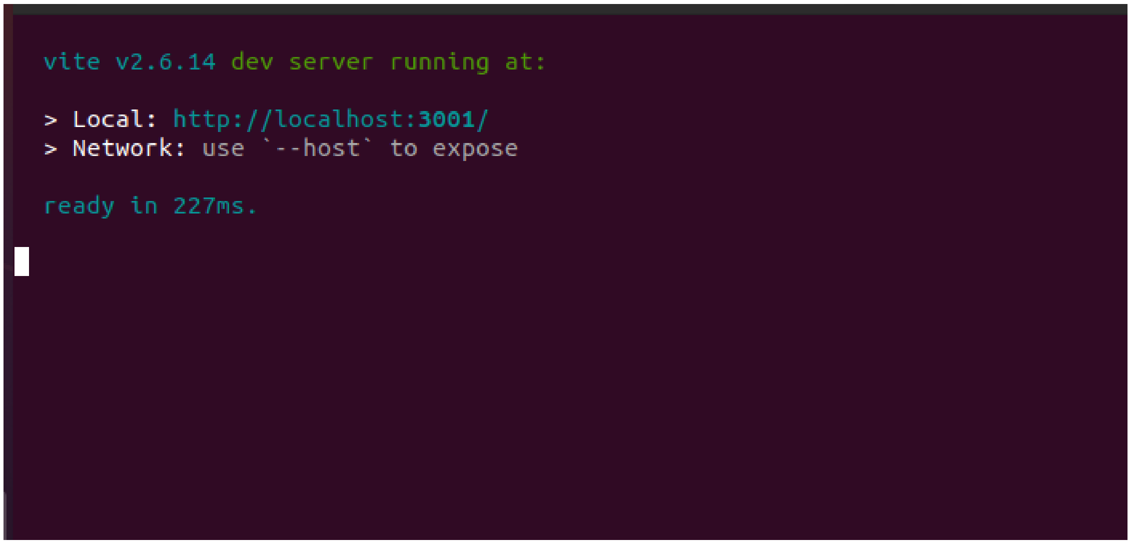
To change the port that react app runs on we will change the script code inside the **package.json** file as such



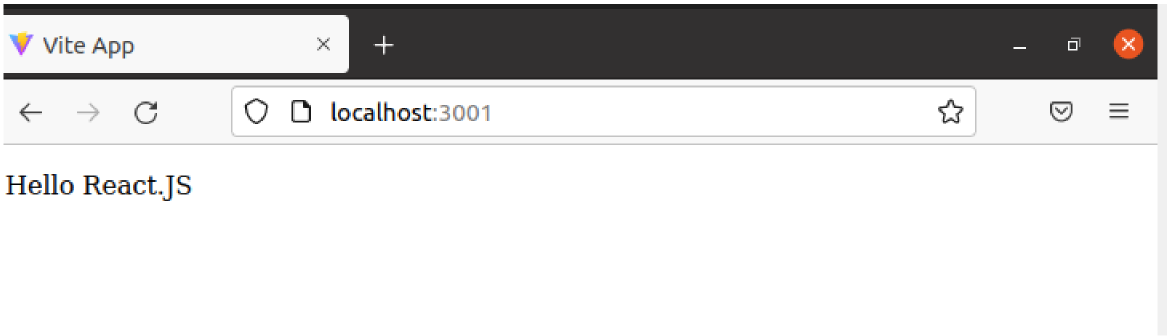
```
client@client-VirtualBox:~/react/devops-intern$ cat package.json
{
  "name": "devops-intern",
  "version": "0.0.0",
  "scripts": {
    "dev": "vite --port 3001",
    "build": "vite build",
    "serve": "vite preview"
  },
  "dependencies": {
    "react": "^17.0.0",
    "react-dom": "^17.0.0"
```

And lastly to simply run it we use the command:

~ npm run dev



```
  vite v2.6.14 dev server running at:

  > Local: http://localhost:3001/
  > Network: use `--host` to expose

  ready in 227ms.

```



Hello React.JS