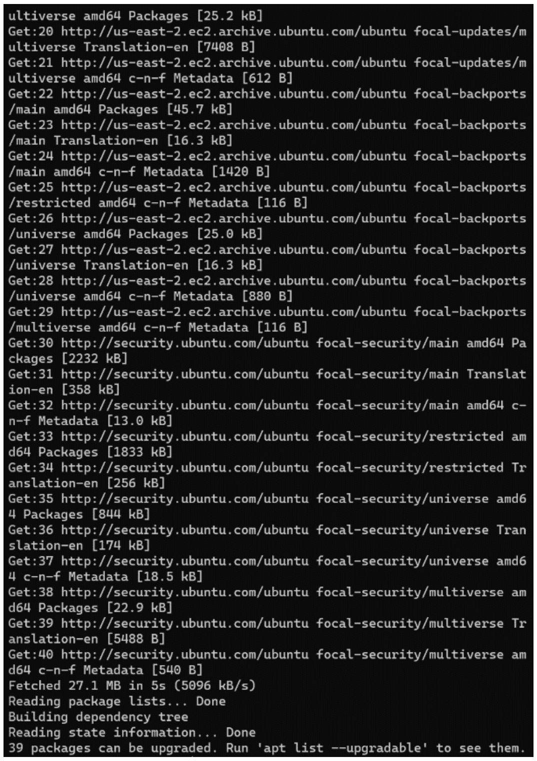
STEP 1 – BACKEND CONFIGURATION

Update ubuntu

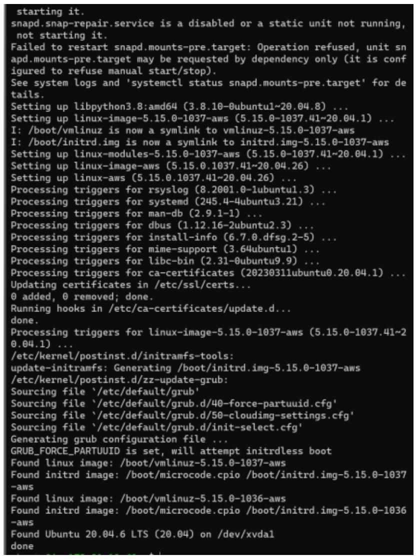
```
sudo apt update
```



```
ultiverse amd64 Packages [25.2 kB]
Get:20 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/m
ultiverse Translation-en [7408 B]
Get:21 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/m
ultiverse amd64 c-n-f Metadata [612 B]
Get:22 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports
/main amd64 Packages [45.7 kB]
Get:23 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports
/main Translation-en [16.3 kB]
Get:24 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports
/main amd64 c-n-f Metadata [1420 B]
Get:25 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports
/restricted amd64 c-n-f Metadata [116 B]
Get:26 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports
/universe amd64 Packages [25.0 kB]
Get:27 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports
/universe Translation-en [16.3 kB]
Get:28 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports
/universe amd64 c-n-f Metadata [880 B]
Get:29 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports
/multiverse amd64 c-n-f Metadata [116 B]
Get:30 http://security.ubuntu.com/ubuntu focal-security/main amd64 Pa
ckages [2232 kB]
Get:31 http://security.ubuntu.com/ubuntu focal-security/main Translat
ion-en [358 kB]
Get:32 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-
n-f Metadata [13.0 kB]
Get:33 http://security.ubuntu.com/ubuntu focal-security/restricted am
d64 Packages [1833 kB]
Get:34 http://security.ubuntu.com/ubuntu focal-security/restricted Tr
anslation-en [256 kB]
Get:35 http://security.ubuntu.com/ubuntu focal-security/universe amd6
4 Packages [844 kB]
Get:36 http://security.ubuntu.com/ubuntu focal-security/universe Tran
slation-en [174 kB]
Get:37 http://security.ubuntu.com/ubuntu focal-security/universe amd6
4 c-n-f Metadata [18.5 kB]
Get:38 http://security.ubuntu.com/ubuntu focal-security/multiverse am
d64 Packages [22.9 kB]
Get:39 http://security.ubuntu.com/ubuntu focal-security/multiverse Tr
anslation-en [5488 B]
Get:40 http://security.ubuntu.com/ubuntu focal-security/multiverse am
d64 c-n-f Metadata [540 B]
Fetched 27.1 MB in 5s (5096 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
39 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Upgrade ubuntu

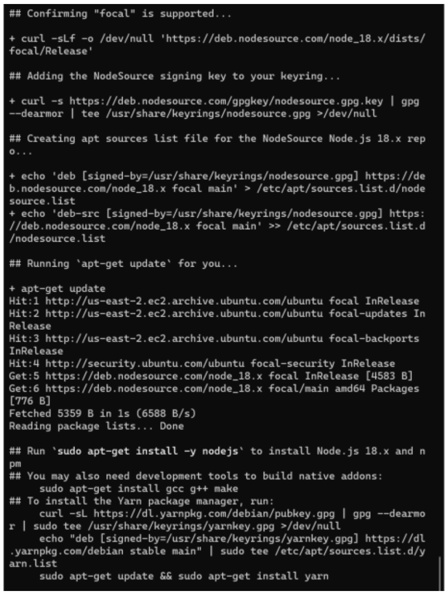
```
sudo apt upgrade
```



```
 starting it.
snapd.snap-repair.service is a disabled or a static unit not running,
 not starting it.
Failed to restart snapd.mounts-pre.target: Operation refused, unit sn
apd.mounts-pre.target may be requested by dependency only (it is conf
igured to refuse manual start/stop).
See system logs and 'systemctl status snapd.mounts-pre.target' for de
tails.
Setting up libpython3.8:amd64 (3.8.10-0ubuntu1~20.04.8) ...
Setting up linux-image-5.15.0-1037-aws (5.15.0-1037.41~20.04.1) ...
I: /boot/vmlinuz is now a symlink to vmlinuz-5.15.0-1037-aws
I: /boot/initrd.img is now a symlink to initrd.img-5.15.0-1037-aws
Setting up linux-modules-5.15.0-1037-aws (5.15.0-1037.41~20.04.1) ...
Setting up linux-image-aws (5.15.0.1037.41~20.04.26) ...
Setting up linux-aws (5.15.0.1037.41~20.04.26) ...
Processing triggers for rsyslog (8.2001.0-1ubuntu1.3) ...
Processing triggers for systemd (245.4-4ubuntu3.21) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.3) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
Processing triggers for ca-certificates (20230311ubuntu0.20.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for linux-image-5.15.0-1037-aws (5.15.0-1037.41~2
0.04.1) ...
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-5.15.0-1037-aws
/etc/kernel/postinst.d/zz-update-grub:
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/40-force-partuuid.cfg'
Sourcing file `/etc/default/grub.d/50-cloudimg-settings.cfg'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
GRUB_FORCE_PARTUUID is set, will attempt initrdless boot
Found linux image: /boot/vmlinuz-5.15.0-1037-aws
Found initrd image: /boot/microcode.cpio /boot/initrd.img-5.15.0-1037
-aws
Found linux image: /boot/vmlinuz-5.15.0-1036-aws
Found initrd image: /boot/microcode.cpio /boot/initrd.img-5.15.0-1036
-aws
Found Ubuntu 20.04.6 LTS (20.04) on /dev/xvda1
done
```

Lets get the location of Node.js software from Ubuntu repositories.

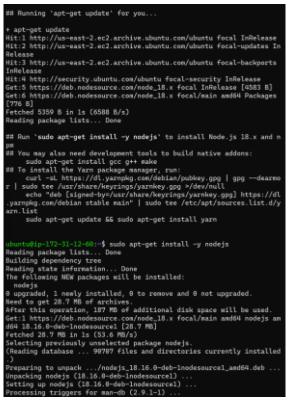
```
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
```



```
## Confirming "focal" is supported...

+ curl -sLf -o /dev/null 'https://deb.nodesource.com/node_18.x/dists/
focal/Release'

## Adding the NodeSource signing key to your keyring...

+ curl -s https://deb.nodesource.com/gpgkey/nodesource.gpg.key | gpg
--dearmor | tee /usr/share/keyrings/nodesource.gpg >/dev/null

## Creating apt sources list file for the NodeSource Node.js 18.x rep
o...

+ echo 'deb [signed-by=/usr/share/keyrings/nodesource.gpg] https://de
b.nodesource.com/node_18.x focal main' > /etc/apt/sources.list.d/node
source.list
+ echo 'deb-src [signed-by=/usr/share/keyrings/nodesource.gpg] https:
//deb.nodesource.com/node_18.x focal main' >> /etc/apt/sources.list.d
/nodesource.list

## Running `apt-get update` for you...

+ apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates In
Release
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports
InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:5 https://deb.nodesource.com/node_18.x focal InRelease [4583 B]
Get:6 https://deb.nodesource.com/node_18.x focal/main amd64 Packages
[776 B]
Fetched 5359 B in 1s (6588 B/s)
Reading package lists... Done

## Run `sudo apt-get install -y nodejs` to install Node.js 18.x and n
pm
## You may also need development tools to build native addons:
     sudo apt-get install gcc g++ make
## To install the Yarn package manager, run:
     curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | gpg --dearmo
r | sudo tee /usr/share/keyrings/yarnkey.gpg >/dev/null
     echo "deb [signed-by=/usr/share/keyrings/yarnkey.gpg] https://dl
.yarnpkg.com/debian stable main" | sudo tee /etc/apt/sources.list.d/y
arn.list
     sudo apt-get update && sudo apt-get install yarn
```

Install Node.js with the command below

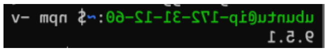
```
sudo apt-get install -y nodejs
```



```
## Running `apt-get update` for you...

+ apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates In
Release
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports
InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:5 https://deb.nodesource.com/node_18.x focal InRelease [4583 B]
Get:6 https://deb.nodesource.com/node_18.x focal/main amd64 Packages
[776 B]
Fetched 5359 B in 1s (6588 B/s)
Reading package lists... Done

## Run `sudo apt-get install -y nodejs` to install Node.js 18.x and n
pm
## You may also need development tools to build native addons:
     sudo apt-get install gcc g++ make
## To install the Yarn package manager, run:
     curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | gpg --dearmo
r | sudo tee /usr/share/keyrings/yarnkey.gpg >/dev/null
     echo "deb [signed-by=/usr/share/keyrings/yarnkey.gpg] https://dl
.yarnpkg.com/debian stable main" | sudo tee /etc/apt/sources.list.d/y
arn.list
     sudo apt-get update && sudo apt-get install yarn


ubuntu@ip-172-31-12-60:~$ sudo apt-get install -y nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 28.7 MB of archives.
After this operation, 187 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_18.x focal/main amd64 nodejs am
d64 18.16.0-deb-1nodesource1 [28.7 MB]
Fetched 28.7 MB in 1s (53.6 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 98787 files and directories currently installed
.)
Preparing to unpack .../nodejs_18.16.0-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (18.16.0-deb-1nodesource1) ...
Setting up nodejs (18.16.0-deb-1nodesource1) ...
Processing triggers for man-db (2.9.1-1) ...
```

Verify the node installation with the command below

```
npm -v
```

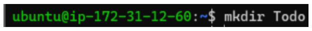


```
ubuntu@ip-172-31-12-60:~$ npm -v
9.5.1
```

Application Code Setup

Create a new directory for your To-Do project:

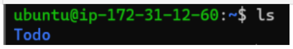
```
mkdir Todo
```



```
ubuntu@ip-172-31-12-60:~$ mkdir Todo
```

Run the command below to verify that the `Todo` directory is created with `ls` command

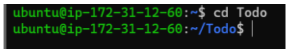
```
Ls
```



```
ubuntu@ip-172-31-12-60:~$ ls
Todo
```

Now change your current directory to the newly created one:

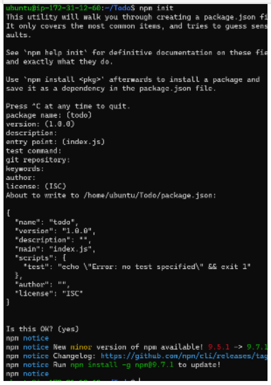
```
cd Todo
```



```
ubuntu@ip-172-31-12-60:~$ cd Todo
ubuntu@ip-172-31-12-60:~/Todo$
```

use the command `npm init` to initialise the project, so that a new file named `package.json` will be created. This file will normally contain information about your application and the dependencies that it needs to run. Follow the prompts after running the command. You can press `Enter` several times to accept default values, then accept to write out the `package.json` file by typing `yes`

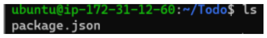
```
npm init
```



```
ubuntu@ip-172-31-12-60:~/Todo$ npm init
This utility will walk you through creating a package.json fi
It only covers the most common items, and tries to guess sens
aults.

See `npm help init` for definitive documentation on these fie
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: (todo)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/ubuntu/Todo/package.json:

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


Is this OK? (yes)
npm notice
npm notice New minor version of npm available! 9.5.1 -> 9.7.1
npm notice Changelog: https://github.com/npm/cli/releases/tag
npm notice Run npm install -g npm@9.7.1 to update!
npm notice
```

Run the command `ls` to confirm that you have package.json file created.

ls

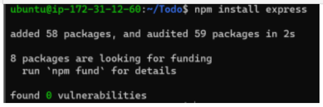


```
ubuntu@ip-172-31-12-60:~/Todo$ ls
package.json
```

INSTALL EXPRESSJS

To use express, install it using npm:

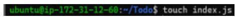
```
npm install express
```



```
ubuntu@ip-172-31-12-60:~/Todo$ npm install express

added 58 packages, and audited 59 packages in 2s

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

found 0 vulnerabilities
```

Now create a file `index.js` with the command below

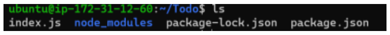
```
touch index.js
```



```
ubuntu@ip-172-31-12-60:~/Todo$ touch index.js
```

Run `ls` to confirm that your index.js file is successfully created

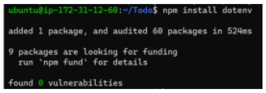
Ls



```
ubuntu@ip-172-31-12-60:~/Todo$ ls
index.js  node_modules  package-lock.json  package.json
```

Install the `dotenv` module

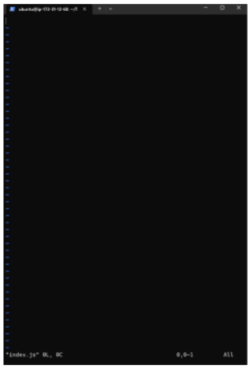
```
npm install dotenv
```



```
ubuntu@ip-172-31-12-60:~/Todo$ npm install dotenv

added 1 package, and audited 60 packages in 524ms

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

found 0 vulnerabilities
```

Open the index.js file with the command below

```
vim index.js
```

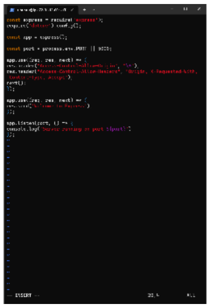


```
"index.js" 0L, 0C                0,0-1        All
```

Type the code below into it and save. Do not get overwhelmed by the code you see. For now, simply paste the code into the file.

```
const express = require('express');
require('dotenv').config();

const app = express();

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

app.use((req, res, next) => {
res.header("Access-Control-Allow-Origin", "\*");
res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");
next();
});

app.use((req, res, next) => {
res.send('Welcome to Express');
});
```

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

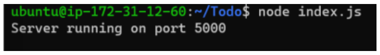


```
const express = require('express');
require('dotenv').config();

const app = express();

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

app.use((req, res, next) => {
res.header("Access-Control-Allow-Origin", "\*");
res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");
next();
});

app.use((req, res, next) => {
res.send('Welcome to Express');
});

app.listen(port, () => {
console.log(`Server running on port ${port}`)
});
-- INSERT --                                    20,4          All
```

Use `:w` to save in vim and use `:qa` to exit vim

start our server to see if it works. Open your terminal in the same directory as your index.js file and type:

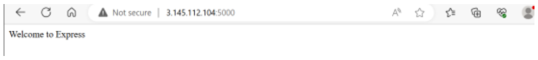
```
node index.js
```



```
ubuntu@ip-172-31-12-60:~/Todo$ node index.js
Server running on port 5000
```

created an inbound rule to open TCP port 5000.

```
http://<PublicIP-or-PublicDNS>:5000
```



Not secure | 3.145.112.104:5000

Welcome to Express

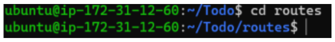
create `routes` that will define various endpoints that the `To-do` app will depend on. So let us create a folder `routes`

```
mkdir routes
```

```
ubuntu@ip-172-31-12-60:~/Todo$ mkdir routes
```

Change directory to `routes` folder.

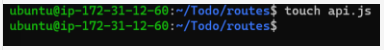
```
cd routes
```



```
ubuntu@ip-172-31-12-60:~/Todo$ cd routes
ubuntu@ip-172-31-12-60:~/Todo/routes$ |
```

Now, create a file `api.js` with the command below

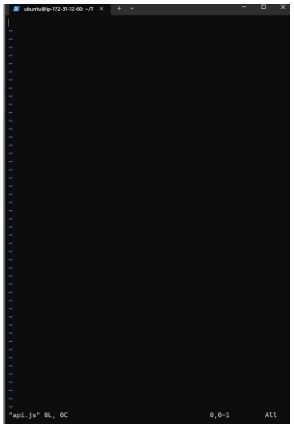
```
touch api.js
```



```
ubuntu@ip-172-31-12-60:~/Todo/routes$ touch api.js
ubuntu@ip-172-31-12-60:~/Todo/routes$
```

Open the file with the command below

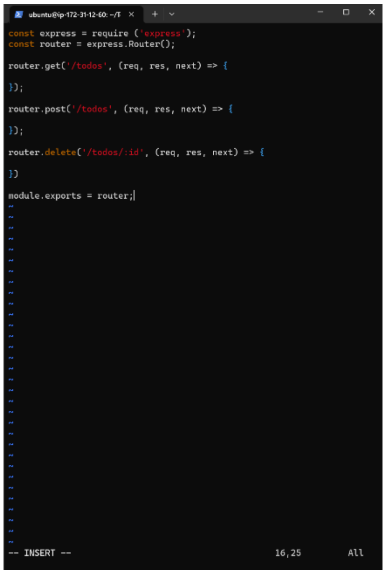
```
vim api.js
```



```
"api.js" 0L, 0C                                    0,0-1         All
```

Copy below code in the file.

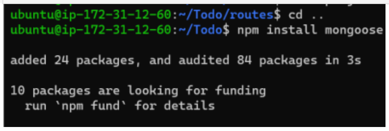
```
const express = require ('express');
const router = express.Router();

router.get('/todos', (req, res, next) => {

});

router.post('/todos', (req, res, next) => {

});

router.delete('/todos/:id', (req, res, next) => {

})

module.exports = router;
```



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

router.get('/todos', (req, res, next) => {

});

router.post('/todos', (req, res, next) => {

});

router.delete('/todos/:id', (req, res, next) => {

})

module.exports = router;
-- INSERT --                                16,25         All
```

Change directory back Todo folder with `cd ..` and install Mongoose

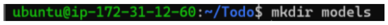
```
npm install mongoose
```



```
ubuntu@ip-172-31-12-60:~/Todo/routes$ cd ..
ubuntu@ip-172-31-12-60:~/Todo$ npm install mongoose

added 24 packages, and audited 84 packages in 3s

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

Create a new folder `models` :

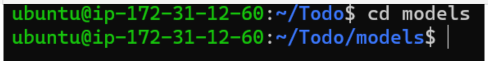
```
mkdir models
```



```
ubuntu@ip-172-31-12-60:~/Todo$ mkdir models
```

Change directory into the newly created ‘models’ folder with

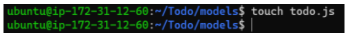
```
cd models
```



```
ubuntu@ip-172-31-12-60:~/Todo$ cd models
ubuntu@ip-172-31-12-60:~/Todo/models$
```

Inside the models folder, create a file and name it `todo.js`

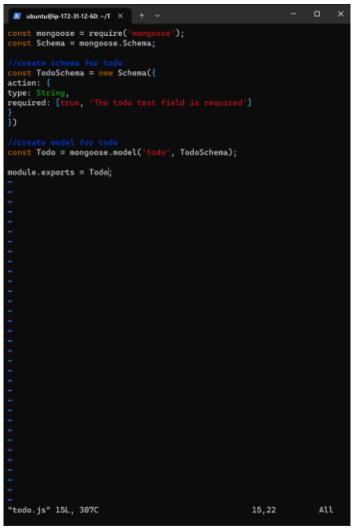
```
touch todo.js
```



```
ubuntu@ip-172-31-12-60:~/Todo/models$ touch todo.js
ubuntu@ip-172-31-12-60:~/Todo/models$
```

Open the file created with `vim todo.js` then paste the code below in the file:

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

//create schema for todo
const TodoSchema = new Schema({
action: {
type: String,
required: [true, 'The todo text field is required']
}
})

//create model for todo
const Todo = mongoose.model('todo', TodoSchema);

module.exports = Todo;
```



```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

//create schema for todo
const TodoSchema = new Schema({
action: {
type: String,
required: [true, 'The todo text field is required']
}
})

//create model for todo
const Todo = mongoose.model('todo', TodoSchema);

module.exports = Todo;
"todo.js" 15L, 307C                                   15,22         All
```

Update our routes from the file `api.js` in 'routes' directory to make use of the new model.

In Routes directory, open api.js with `vim api.js`, delete the code inside with `:%d` command and paste there code below into it then save and exit

```
const express = require ('express');
const router = express.Router();
const Todo = require('../models/todo');

router.get('/todos', (req, res, next) => {
```

```
//this will return all the data, exposing only the id and action field to the
client
Todo.find({}, 'action')
.then(data => res.json(data))
.catch(next)
});

router.post('/todos', (req, res, next) => {
if(req.body.action){
Todo.create(req.body)
.then(data => res.json(data))
.catch(next)
}else {
res.json({
error: "The input field is empty"
})
}
});

router.delete('/todos/:id', (req, res, next) => {
Todo.findOneAndDelete({"_id": req.params.id})
.then(data => res.json(data))
.catch(next)
})

module.exports = router;
```