

-npm init ==> to create package.json::

-create folders ::

- routes
- controllers
- config
- models
- views

-npm install express::

-fire up the express server::

```
const express = require('express');
const app = express();
const port = 8000; // When we deploy the app on production its port should be 80
```

```
app.listen(port, function(err){
  if(err){
    console.log(`There is an error in running the server`);
    return;
  }
  console.log(`Server is running fine on port ${port}`);
});
```

-Now, get rid of firing the server. So go to package.json

=> under script => add "start" key and make its value as "nodemon index.js"

=> So now onwards just write "npm start" on terminal to fire up the server.

-Now initialise git

=>git init

-add gitignore file to ignore node_modules folder as this folder makes lot of weightage in space.
Then stage and commit everything.

-setup the routes::

- require express in it.
- then take router = express.Router()
- then export the router(to be accessible from our app)

- require the router in app.

- use middleware to set if there is any request url start with '/' then go to router::

```
const router = require('./routes/index')
app.use('/',router); //use express router
```

- once you get into routes then write HTTP request methods. Once we identified the URL then we have to call appropriate controller function.

```
router.get('/',controller.home);
```

Check the example ::

main app::

```
const express = require('express');
const app = express();
const port = 8000; // When we deploy the app on production its port should be 80
const router = require('./routes/index')

//use express router
app.use('/',router);

app.listen(port, function(err){
```

```
    if(err){
      console.log(`There is an error in running the server`);
      return;
    }
    console.log(`Server is running fine on port ${port}`);
  })
```

routes -> index.js

```
const express = require('express');
const { appendFile } = require('fs');
const controller = require('../controllers/index');

//router helps us to separate routes and controller
const router = express.Router();

// console.log("router loaded");
router.get('/', controller.home);

module.exports = router; //external file can use it
```

controller->index.js

```
module.exports.home=function(req,res){
  res.end("<h1>Hi from server</h1>")
}
```