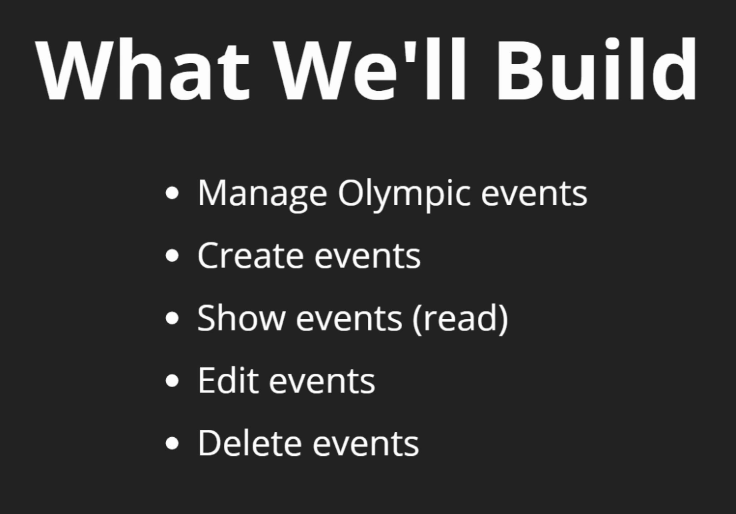
What we’ll build:



1. Let’s create a new Directory named “node-crud’
2. And Initiate Git init from cmd line
3. Let’s start node application with ‘npm init’
4. That will create our ‘package.json’ file
5. Let’s install ‘exress’
6. npm install - - save express
7. Open sublime with subl . command
8. Hide that menu with Ctrl+shift+P Toggle Menu
9. Create a new file named server.js as we’ll call that file the start file.

10.Command palette newfile + enter file name

11. // grab our dependencies

// configure our application

// set the routes

// start our server

This is how we layout our server.js to Start.

12. We import ‘express’ and store it in const variable

And we start an instance of express named ‘app’

const express = require('express'),

app = express();

13. Then we create ‘port’ variable that will grab the port our server is supposed to listen to, and if that port doesn’t exist in our environment variables, we’ll create a default one

const express = require(‘express’),

app = express(),

port = process.env.PORT || 8080;

14. Down in the ‘set the routes’ we create a home page route, with ES5 style

app.get(‘/’, function (req, res) {

res.send(‘Hello I am the App!’);

});

With ES6 style:

app.get(‘/’, (req, res) => {

res.send(‘Hello I am the App!’);

});

15. And we’ll start the server ES6 style:

App.listen(port, () => {

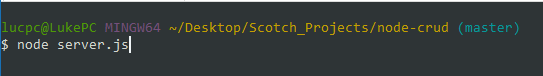
Console.log(`App listening on http://localhost:${port}`);

});

We are using the ‘`’ back tick in the template syntax is because we are allowed to do variables ${port} in those template syntaxes.

That’s all we need to start the server and see our app in our browser now!

16. Back in the Terminal we could start our server with:

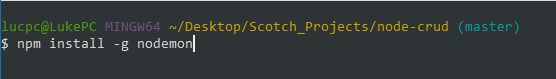


But since we’re developing let’s use a package named:

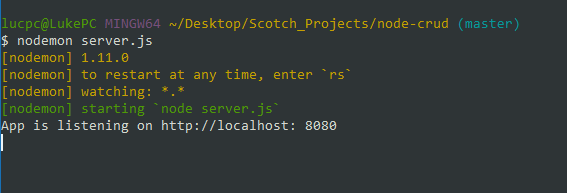
‘nodemon’

‘nodemon’ will watch for any file changes, and update our server in case of those file changes.

That way we won’t have to close/restart our server constantly.



Then once installed lets use ‘nodemon’ to start the server with:



The server is started, and we can see the message that we console.log() in our code, and we can also go see in the browser @ localhost:8080

