

Note: this example cannot be done on the Raspberry Pi as there is not a version of the CloudFoundry command line tool available for the Raspberry Pi.

On your laptop or workstation with the prerequisite Node, NPM and cf software installed open a command or terminal window and create a new directory to work in.

enter the **npm init** command

complete the prompts. Ensure you enter **server.js** as the entry point when prompted

Once you have answered all the questions you will have the package.json file for your project. If you entered any values incorrectly whilst being prompted you can edit the package.json file and make any corrections.

Create a new file called **server.js**. In the video I use the Atom editor (<https://atom.io>) to create the file, but you can use any editor you are comfortable using. If using a text editing application not specifically aimed at developers please ensure any smart text substitution features are turned off, as these will convert some character, such as the double quote (") character, to look nicer when printed to screen or on paper. Code interpreters and compilers do not recognise these converted characters and will give a syntax error.

Enter the following code, which implements the basic server:

```
1 var express = require("express");
2 var app = express();
3
4 var serverPort = 3000;
5 var serverHost = 'localhost';
6
7 var server = app.listen(serverPort, serverHost, function() {
8   var host = server.address().address;
9   var port = server.address().port;
10  console.log('Listening at http://%s:%s', host, port);
11 });
```

As the code requires the express package, you need to install this package to your local system using the npm command. The --save option will automatically update package.json for you.

npm install express --save

test your code by running:

npm start

The server should start without any errors being displayed. However, this server is not much use as it doesn't return anything. To exit the server press and hold the Ctrl key then hit c (Ctrl-c).

Add a route to the server so a browser can get a return from the server. Add the following code above the line containing `var server = app.listen(.....`

```
1 app.get('/', function(req, res) {
2   res.send('Hello Coursera');
3 });
```

Your application should now be:

```
1 var express = require("express");
2 var app = express();
3
4 var serverPort = 3000;
5 var serverHost = 'localhost';
6
7 app.get('/', function(req, res) {
8   res.send('Hello Coursera');
9 });
10
11 var server = app.listen(serverPort, serverHost, function() {
12   var host = server.address().address;
13   var port = server.address().port;
14   console.log('Listening at http://%s:%s', host, port);
15 });
```

Save the modified server.js file and run the server using the **npm start** command

On the same machine running the server launch a browser and enter <http://127.0.0.1:3000> in the address bar. You should see Hello Coursera displayed in the browser, showing the server and route are working.