Exercise after Assignment 0

In your folder (which I have directly at the top level – that is my folder is C:\myNode)

1. Create a folder **firstApp**
	1. Inside that folder create a file with the code:

	console.log('Hello World');
	2. Save the file as app.js
	3. Open the terminal
	4. Run app.js by typing **node app** in when you are in the right folder.
2. Create a folder **server1**
	1. Inside that folder create a file with the code:

	const http = require('http');

const server = http.createServer((req,res) => {console.log(req.url);

 res.end('Hello World of Node')

 })

 server.listen(3000)

* 1. Save the file as app.js
	2. Open the terminal if nec
	3. Run the app
	4. Open a browser and go to the url [**http://localhost:3000/**](http://localhost:3000/)
	5. See what happens in the browser and then look at the terminal and you will see that the **/** from the end of <http://localhost:3000/> has been logged.
	6. Now at the browser go to the url <http://localhost:3000/nodeRocks/>
	7. Now go back to the terminal and you will see that the server app has retrieved information from the req (request).
	8. You can imagine that you have a website with multiple pages (myWebSite.com/somePage) and that our server retrieves the URL as above, and uses the url module to find out what page the request came from, and responds appropriately. Wow! We will implement this in the next app.
	9. This example and the next is from greg Lim's book "Beginning Node.js", which is not in the ACM Library.
1. Create a folder **server2**
	* 1. Create and save a file app.js with the following code:

Modify the code in server1 to be as follows:
(NOTE: If we weren't on localhost then we would need to massage the url a bit to get the file.)

const http = require('http');

const server = http.createServer((req,res) => {

  if (req.url === '/about')

     res.end('The about page')

  else if (req.url === '/contact')

     res.end('The contact page')

  else if (req.url ==='/')

     res.end('The home page')

  else {

     res.writeHead(404);

     res.end('Page not found')

  }

  })

  server.listen(3000)

* 1. Run the app (so that the server is listening)
	2. Test it in your browser by going to <http://localhost:3000/about> etc.
	3. Note: we have devoted port 3000 to servicing this particular web site. There is nothin magic about 3000.
	4. The writeHead methods writes the status code. (Of course, 404 is the infamous 'page not found' code, and 200 is the code for successful response.
1. You can check the Status Code of any page in the developper tools – here are screen shots of my going to nytimes.com on both Chrome and Firefox ---- and using the Networks tab.

Chrome:


Firefox: NOTE: In Firefox you need to go to the Networks tab and then *reload* the page to get this info.



BTW – In Firefox (only the start of the GETs shows) look at all the files which were downloaded by the NYTimes website – a total of 36 requests! For some reason Chrome shows 112 requests (possibly from a reloaded page. *Of course, as a developper you will also be interested in seein*g *how fast your pa*g*es load/*

In your browsers – in both Chrome and Firefox- go to the URL localhost:3000/yuck and see what the Network tab shows. (Remember that in FF you need to reload the page after you are on the Networks tab.)