# *Web Programming III (420-C30-HR)*

# *Assignment 2 – node.js*

Date assigned: Wednesday, October 5, 2016

Date due: **Wednesday, October 26, 2016**

**Learning Objectives**

Upon successful completion of this assignment, the student will be able to:

* Create a functional web server using node.js
* Return static pages and data from a web server location depending on the request

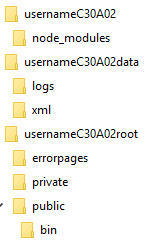
To do:

**General idea of the assignment:**

Create a node.js web server. The server will return all kinds of data (html, images, css, xml, etc). The server will also return data from a file when requested from a form.

**Details:**

You are creating a web server using **native** node.js modules. You can use downloaded modules for the last section.

1. You will need the following folder structure  
   

There may be other folders under those folders. The folders are used as follows:

1. usernameC30A02 – where the node application will be created. The node\_modules will be created by npm if you download any node modules (see later steps).
2. usernameC30A02data – where data files will be kept. This includes the data you will be writing to usernamedata.txt. The logs subfolder contains the logs for your webserver and the xml subfolder contains the xml files you will need (see provided files).
3. usernameC30A02root – HTML page home directory. All publicly accessible files will be in public and any subsequent subfolders. The bin folder in public will be for files which “execute” code (see below). The errorpages folder contains HTML copies of error pages.

The following files have been provided.

* In usernameC30A02root/public, 7 HTML files with various items such as images, CSS, etc. A PDF, text and XML file. A folder called images that the HTML files use, a folder called styles that the HTML files use for CSS. A newpath/evenmore path for a file along the path. Feel free to add more files to this folder as you require/wish. Note: I will be using different files for my testing…
* usernameC30A02root/errorpages is empty and will contain your error pages
* In usernameC30A02data/xml, are three XML files that are searchable. Books.xml, albums.xml and patients.xml

You assignment must be in the form of a node.js application/package complete with a package.json, README and a list of dependencies (you will be using at least one module which must be specified in the dependencies – see below).

The first thing that I will do when I mark your assignment is an npm install. You will NOT submit the node\_modules folder when you submit your assignment. The npm install must take care of downloaded all needed, non-core, node modules.

You must do the following when the request is received:

1. You will create a web server that runs on port 9000.
2. The web server must return static pages as follows:
   1. Return the index.html file from the usernameC30A02root/public folder (Except in the cases described below) when no file name is provided; i.e. localhost:9000.
   2. Return files from any path in the usernameC30A02root/public subfolder; i.e., localhost:9000/p/a/t/h/mypage.html
   3. Return index.html from any path in the usernameC30A02root/public subfolder when no file name is provided; i.e., localhost:9000/p/a/t/h/ \*\*this will require some work\*\*
   4. Be able to return files of the following types with the following extensions:

|  |  |
| --- | --- |
| Extensions | Mime type |
| .html | text/html |
| .css | text/css |
| .js | application/javascript |
| .png | image/png |
| .jpg/.jpeg | image/jpeg |
| .gif | image/gif |
| .pdf | application/pdf |
| .svg | image/svg+xml |
| .xml | text/xml |
| .txt | text/plain |

* 1. Use a JavaScript object of the extensions and the mime types for step (b)

NOTE: Remember the browser will automatically request (GET) any linked files when the HTML page is returned.

* 1. Return error 415 in a page you create if a file with a different extension is requested. The page is located in the usernameC30A02root/errorpages folder.
  2. Return error 404 in a page you create if a non-existent file is requested. The page is located in the usernameC30A02root/errorpages folder.
  3. Return error 500 in a page you create if an error is encountered in reading or opening the file. The page is located in the usernameC30A02root/errorpages folder.

1. When the path name is /bin/ the following must occur.
   1. Check the filename to make sure it is an xml file (has the extension xml). If the filename is not an XML file, return error 400 in a page you create. The page is located in the usernameC30A02root/errorpages folder.
   2. Check the filename to make sure it exists in the usernameC30A02**data/xml** folder. If not return error 404 page as above.
   3. If the filename does exist parse the query string. If there is no query string, return error 406 in a page you create. The page is located in the usernameC30A02root/errorpages folder.
   4. If the filename exists, read the file contents and convert it to a JSON object. You will need a node module to do this. I recommend that you use xml2js module ([https://github.com/Leonidas-fro­­­­­­m-XIV/node-xml2js](https://github.com/Leonidas-from-XIV/node-xml2js)) or xml2js-parser (<https://www.npmjs.com/package/xml2js-parser>). Both are documented at the links provided; both will give warnings when installed, but should work for our purposes.
   5. Once you read in the file, search the file for the record matching the record specified by the query string.
   6. If no record matches (either because the query string names are incorrect or the values do not match), return error 416 (Range not satisfiable).
   7. If a matching record is found, return the record nicely formatted in the response. Make the response an HTML file (You can do this by hand or use a node module to do this. I would recommend either jade (<https://www.npmjs.com/package/jade>) or mustache (<https://www.npmjs.com/package/mustache>) to do this).

For example: localhost:9000/bin/books.xml?title=Everyday Italian, would check to makes sure the file was xml (it is), check to see if it exists (it does) opens and reads the files books.xml to a json array, parses the query string to a json object, loops through the json array searching for the value in the query string matching the name from the query string.

1. When the request is a POST do the following when:
   1. Get the data from the post using event emitters.
   2. Create a user object and write the information as a comma separated file (CSV) to the file users.txt. The user data file is located in the usernameC30A02**data** folder.

TODO: MODIFY THIS SEGMENT SO THAT IT WRITES TO AND THEN READS FROM THE USER OBJECT

* 1. Return the object saved in an appropriate (nicely formatted) HTML page. You can do this with native node.js or use a template as above.
  2. If an error is encountered writing the object, return the error page with error code 520, a message to the user and the error object.
  3. You will have to create the HTML page which requests this information. It should have a form which uses the POST method. The form (and subsequent user object) has the following attributes: firstName, lastName, username, emailAddress and phoneNum. This form should be located in the appropriate location for HTML files.

**Other things**

1. Every request, good or bad, must be logged to the file web.log in the folder usernameC30A02**data/logs.**
2. You must create a user object as a node.js module. This must be in its own folder. Include the entire module and not just a file name within it.
3. I have provided a number of html files with embedded images, external CSS and JavaScript files. I have also provided a couple of text, pdf and XML files to test with as well.
4. There are marks for interface design. This only applies to the error pages and the returned page on the post.
5. The canonical list of HTTP Status Codes can be found here: <https://en.wikipedia.org/wiki/List_of_HTTP_status_codes>
6. Make the code efficient and modular; there are marks allocated for it.
7. You must use HTML5 and (proper) CSS techniques for all HTML files you create in this assignment.

**To submit**

When you have completed the assignment zip the three folders containing all the files for the assignment and copy it to the course page.