

Web Science

Quiz 1: February 28, 2023

Enter your answers directly into this document (with the exception of #2 and #3).

All answers should be In Your Own Words, using complete sentences with proper spelling and grammar.

Save this document as either a Word docx or a PDF. For all questions other than #2 and #3, you will not receive any credit for answers not placed in this document.

When finished with the quiz, put everything you wrote (this document, all code, etc.) in your personal GitHub repo in a folder named **quiz1**. You must test your code on your VM, served with Node. No Apache!

1. **Short answers** (25 points): (Answer in complete sentences, explain your answers)

- a. (5) What are the potential advantages of using TypeScript's type system as opposed to typeless JavaScript?

Using TypeScript allows the programmer to ensure that the original variable type remains consistent throughout its use. Since Typescript can determine the type of variable at a given point in the code, it can help with debugging when it comes to calling functions that do not exist. For example, if a variable is initially created as a string, but is given a number value when pulling information as a form, the substring method should not work on that variable and TypeScript will help catch that mistake.

- b. (7) What is a package.json file? What is it used for? How is it created?

A package.json file is used to determine which packages are installed and show clearly which file npm start will run. Running the command npm init on a project will create the package.json and package-lock.json files. The "main" key value pair is one of the more important elements of the package.json file as it lets other users know which file to run when hosting the app through Node. Another important element is the dependencies. In lab3 when working on building an internal API, "express" and "require" were installed to complete the functionality of the program, and these two can be shown in the package.json file to show that these were used as well as the version of the dependency.

- c. (3) Of the following permutations, identify which are valid and which are invalid for shipping/installing a Node application: 1. (package.json && package-lock.json); 2. package.json only; 3. package-lock.json only

Both the package.json and package-lock.json are needed for shipping/installing a node application. Therefore 1 is valid and 2 and 3 are invalid. Package-lock.json gives the node relationships that the app uses but does not provide information such as which file to run. Package.json does not have all the explicit node relationships of package-lock.json, which means both files are needed.

Note: the above is true when there are dependencies used in the app. If there were no dependencies installed (no express, no require, etc) then permutation 2 is acceptable.

- d. (10) Describe **in detail** the sequence(s) of transaction(s) for a frontend to request data from some external entity via Node.

Requests between the client and the external entity must be completed through the Node.js server. The user starts at the front end and sends a request for the information. Assuming the Node.js server has an internal API, the request processing should formulate a get request from using the url and API key if one exists. Once the get request is formulated, the Node server sends a request through the url for the information. If there is an internal API, the API would then use the provided information in the get request to similarly formulate a get request for the external entity. The node server then externally sends this request to the external location. When the external location returns its response back to node, the internal api processes this response and then returns its response to the frontend along with its own code that contains the data if successful or an error code if not.

2. **Coding question:** (60 points) Here is a free API that does not require any API keys:
<http://universities.hipolabs.com/> – the documentation for which can be found here:
<https://github.com/Hipo/university-domains-list>

Create a new input box (or extend an already created input box) that can accept the name of a university and returns the API's information about that university. If the user fails to input a valid university, return the information for RPI. You might want to extend a recently due lab...

Creativity matters; you need to really integrate this new information into your app. Make it feel like it is a meaningful, conscious, intentioned feature of your app. How you do that is up to you. Don't make it look like some random afterthought. Go beyond the minimum (but remember that creativity doesn't have to be visual). If you need to, write a short README.md file that tells me what I should consider for creativity. (creativity: 30 points of the 60 available for this question)

You may use any and all open source libraries you want for this coding question, so long as you cite them in a README.md file.

3. (15) Ensure the package.json file for Q2 has no errors when I run npm install.

- 4. Extra credit (+5):** Why do your URLs need /node/ in order for Node to serve web pages on your VM?

When the url is not under /node/ the web page is running on apache and not through Node.js. The app needs to be under node in order for the server to recognize the app is being run through node. This acts as the equivalent location for localhost port 3000 when running an app locally on a personal machine.