Documentation Packet Week Ending [Oct 8th]

|  |  |
| --- | --- |
| Team: \_\_\_\_\_\_\_ | Student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Goals:

* Establish and document standards for Notebook entries for your team.
* Create JSON files to store Javascript Objects on the hard disk.
* Read JSON files in NodeJS as parse into a Javascript Object
* Convert Javascript Objects to a string to save in a JSON file
* Use data from a JSON file to populate EJS Templates with data
* Read Query Parameters with ExpressJS
* Save data passed by Query Parameters to a JSON file

# Included Documentation

* Team Documentation: Notebook Entry Format (pg. 2)
* Team Website Upgrade: Load Team Profiles from JSON (pg. 2)
* Team Website Upgrade: Save Query Parameters to JSON (pg. 2)
*  Analysis: Team Composition

# Required Documentation

*  Team Documentation: Notebook Entry Format
*  Analysis: Team Composition
*  Team Website Upgrade: Load Team Profiles from JSON
* Team Website Upgrade: Save Query Params to JSON



* Pencil Svg Png Icon Free Download (#376363 ... Codecademy “Learn Javascript” course, “Functions” Chapter
  + Includes the Lesson, both Projects, and the Quiz

# Changes:

* Grading Rubric: Checkmark is for credit. Cross mark is for missed.

# Events:

* Oct. 4th: Weekly Review
* Oct. 5th: [Lesson] Working with JSON Files
* Oct. 7th: [Lesson] Reading Query Parameters

# Team Documentation: Notebook Entry Format

Consult with your team to determine a set of rules for formatting Notebook entries to ensure that all Notebook entries look as though they were made by the same person at the same time. Create a list of rules and instructions each team member must follow while creating a Notebook entry. Use examples as you explain. Some example ideas:

* How wide are the margins?
* What goes in the Header/Footer?
* What style will all code use?
* How are bulleted lists and steps handled?
* Where are steps, headers, images, and code blocks indented compared to each other
* When is the text and images for a step considered “enough detail”?

# Team Website Upgrade: Load Team Profiles from JSON

Create a JSON file in your team website project folder and populate it with “profiles”. Each “profile” has the team member’s name as the object key, and the value is an object with all the properties that describe that team member (name, bio text, image link, title, etc.). Example:

{

“Lump” : {

“name” : “Lump”,

“bio” : “<p>I\’m pretty dumb.</p>”,

“title” : “Professional Freeloader” }

}

When your website server launches, it should load the JSON file and parse the text into a Javascript object. Save this object to a variable.

When a user connects to the endpoint for a team member’s profile, send the team member’s profile information from the JSON file to the EJS template to populate the web page.

Team Website Upgrade: Save Query Parameters to JSON

Create another JSON file in your team website server files. It contains one property that is an empty array called “comments”:

{ “comments”: [] }

Create a new endpoint in your team website called “feedback”. This endpoint is expecting two Query Parameters called “name” and “adjective”. Example:

<http://localhost:8080/feedback?name=Lump&adjective=dumb>

Using the request and response objects of the app.get(‘/feedback’) callback function, determine what the Query Parameters are available and save their values as an object:

{ name: “Lump”, // The object will end up

adjective: “Dumb” } // looking like this

Open the JSON file you created for this step, push() the object into the comments array, save the file, and respond to the user with a success message. If the user did not provide all the Query Parameters required, respond with the one they missed.

# Reflection

**What was the most impactful thing you learned this week?** **How can/will you use it in the future?**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**What did you do *wrong* this week that you could do better in the future, and how?**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

# Grading

|  |  |  |  |
| --- | --- | --- | --- |
|  | Complete | Followed Instructions | |
| Reflection | ❑ | ❑ | |
|  | Complete | Followed Instructions | |
| C:\Users\csmith\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\6867D38F.tmp Analysis: Team Composition | ❑ | ❑❑ | |
|  | Submitter | | Errors |
| C:\Users\csmith\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\6867D38F.tmp “Team Documentation: Notebook Entry Format” |  | | ❑❑❑ |
|  | Complete | | |
| Eye Icon - Free Download at Icons8 Team Website Upgrade: Load Team Profiles from JSON | ❑❑ | | |
|  | Complete | | |
| Eye Icon - Free Download at Icons8 Team Website Upgrade: Save Query Parameter to JSON | ❑❑ | | |
|  | Complete | | |
| Pencil Svg Png Icon Free Download (#376363 ... Codecademy | ❑❑ | | |
|  | Undamaged | | |
| DocPac Condition | ❑ | | |

# Useful Google Searches:

* How to use Headers/Footers/Styles in Microsoft Word (search one at a time)
* Reading JSON files in NodeJS
* Writing to JSON files in NodeJS
* Javascript array push()
* Add elements to a Javascript array
* Access properties of a Javascript Object
* How to parse JSON into Javascript Objects
* JSON.parse()
* How to convert Javascript Objects into JSON string
* JSON.stringify()
* Reading Query Parameters with ExpressJS