

Weekly reflection

This week I decided it was important to really start applying myself to this course. I took on Json and learned how the input and output streams work and how to serialize an object.

Topic Researched: JSON/QCJSON

JSON is something that I had never heard of in this class and decided it would be a good starting point for me. I went to W3 schools to complete their JavaScript JSON tutorials. Here is what I discovered about JSON:

- JSON stands for JavaScript Object Notation
- JSON is a lightweight data-interchange format
- JSON is language independent *
- JSON is "self-describing" and easy to understand

So the basic premise is this: JSON can do much of what XML documents can do but with less information. In the example online it shows how a JSON object is only 3-4 lines of code that records several instances of an object called "employee" while the XML alternative is several lines and much of it is repetitive.

While this information was helpful I realized I would not be spending a lot of time with JavaScript and should find some more relevant examples. I found a site called [Json-Simple](#) that showed me a version of JSON that I could use inside Eclipse. I followed some tutorials and copied some code into my Git-hub for reference. I found it very easy to use.

QCJSON however I found the API a little more confusing. Having little understanding. Which is ironic since it was made to be a simpler version of JSON. More specifically, it appears to be a collection of classes built around dealing with a Json string/object. It comes with Json input/output classes that handle both coming and going information, as long as you get that information as a hashmap it can create the object. But it also needs to be an object that can be constructed using that hashmap.

New vocabulary:

Hash Map: a data table structured around Key – value pairs.

SocketInputStream:

Links

JSON.SIMPLE and QCJSON examples

https://github.com/Stout-Matthew/understanding_portfolio/blob/master/understand_portfolio/json_understand_one.java

- The above example has the following methods:
 - my_JSON_simple_example_1() -- this is my code, it shows how to create a JSON.simple object
 - online_example_json_simple() – this is internet code that shows how to create an object
 - qcJson_Example_Simple_1() – this is my code on how to serialize QCjson and export/import objects

Sharing videos

https://www.youtube.com/watch?v=sdj07_rTs60

Relevant team meeting

While we did have a team meeting this week, I don't feel there was anything important that I could point you towards for myself.

Additional Code sources:

JSON.simple

[https://code.google.com/p/json-simple/wiki/EncodingExamples#Example_1-1 - Encode a JSON object](https://code.google.com/p/json-simple/wiki/EncodingExamples#Example_1-1_-_Encode_a_JSON_object)

QCJSON

<http://www.quickconnectfamily.org/qcjson/docs/org/quickconnect/json/package-summary.html>