Last Updated: 3/13/15

Topic Name: JSON

Brief Introduction:

JSON stand for “JavaScript Object Notation” and is an easier alternative to XML for formatting data. It’s a lightweight data-interchange format according to W3Schools. Two examples of what typical JSON objects and XML look like are as follows:

JSON Example

```
{"employees":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
]}
```

XML Example

```
<employees>
    <employee>
        <firstName>John</firstName> <lastName>Doe</lastName>
    </employee>
    <employee>
        <firstName>Anna</firstName> <lastName>Smith</lastName>
    </employee>
    <employee>
        <firstName>Peter</firstName> <lastName>Jones</lastName>
    </employee>
</employees>
```

You can easily see how much easier JSON looks and feels compared to XML.
I have found, however, that JSON is not that easy to parse in Java for some reason. You have to go look for third party libraries in order to get the job done very easily – and trust me it’s not that easy.
Once you *are* able to finally manipulate JSON then it becomes much better and feels smooth in comparison to handling XML data. In JavaScript it is very easy to work with and here is an example to demonstrate:

JSON with JavaScript

```
<script>
var text = '{"name":"John Johnson","street":"Oslo West
16","phone":"555 1234567"}'
```

```
var obj = JSON.parse(text);

document.getElementById("demo").innerHTML =
obj.name + "<br>" +
obj.street + "<br>" +
obj.phone;
</script>
```

Comparing JSON to XML in handling looks like this:
Using XML

- Fetch an XML document
- Use the XML DOM to loop through the document
- Extract values and store in variables

Using JSON

- Fetch a JSON string
- JSON.Parse the JSON string

So much easier! (If you are working with JavaScript);

Teaching Description:

None yet.

Teaching Examples:

None yet.

Files to View:

None yet.