Chapter 15.5: JSON

CS 80: Internet Programming

Instructor: Mark Edmonds

JSON describes data

- XML let us specify a data format
- There's another extremely relevant data format: JSON
- JSON stands for **J**ava**S**cript **O**bject **N**otation

JSON desribes data

- In Javascript, basically everything is an object, so we are literally defining an object with JSON
- A simple example:
  - Suppose you wanted a list of employees, each with a first and last name. What would the XML look like?

JSON describes data

- Possibily something like this:

```
<!-- XML to represent a list of employees-->
<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>
```

JSON describes data

- JSON follows a `{"attribute": "value", ... }` model for representing data
  - `[]` can be the value, indicating an array
  - Let's see an example of what we did above

JSON describes data

```
// json to represent a list of employees
{"employees":[
  {"firstName":"John", "lastName":"Doe"},
  {"firstName":"Anna", "lastName":"Smith"},
  {"firstName":"Peter", "lastName":"Jones"}
]}
```

Why is JSON good?

- The web runs on Javascript
- There's a magical parsing function that converts JSON into an actual Javascript object:
  - `JSON.parse(json_text)` where `json_text` is valid JSON text (e.g. the text above)

Why is JSON good?

- Very similar to XML: human readable, hierarchical, widely usable, can be used with Ajax (soon to be covered)
- Dissimilar from XML: no end tag, shorter, quicker to read and write, can use arrays
- **The biggest difference: XML has to be parsed by a specialized XML parser. JSON is parsed with one simple Javascript function call**
- Here's an example of loading JSON into an object:

Example: `simple_json.html`

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title>Simple JSON example</title>
```

```
7    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/
        jquery.min.js"></script>
8    <script>
9    // run when the document is ready
10   $(document).ready(function(){
11     // create JSON data
12     var text = '{"name":"John Johnson","street":"Oslo West 16","phone
          ":"555 1234567"}';
13
14     // parse it into an object
15     var obj = JSON.parse(text);
16
17     // access members/attributes of the new object
18     document.getElementById("demo").innerHTML =
19       obj.name + "<br>" +
20       obj.street + "<br>" +
21       obj.phone;
22   });
23
24  </script>
25 </head>
26
27 <body>
28
29   <h2>JSON Object Creation in JavaScript</h2>
30
31   <p id="demo"></p>
32
33 </body>
34
35 </html>
```

JSON Values

- number (int or float)
- string (double-quoted)
- boolean
- array (indicated with square brackets)
- object (another pair of curly braces)
- null (no content)

JSON Values

- How would you load and print `employees` JSON object above?

Example: `employee_json.html`

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title>Employee JSON example</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/
     jquery.min.js"></script>
  <script>
    // run when the document is ready
    $(document).ready(function() {
      // create JSON text
      var data = '{"employees": [ {"firstName": "John","lastName": "Doe
         " }, {"firstName": "Anna", "lastName": "Smith"}, {"firstName":
          "Peter","lastName": "Jones" }]}';

      // parse JSON text into object
      var employees_obj = JSON.parse(data);

      // setup document
      document.writeln("<h2>Employees list</h2>");
      document.writeln("<p>");
      // loop over every employee
      for (var i = 0; i < employees_obj.employees.length; i++) {
        // print employee info
        document.writeln(employees_obj.employees[i].firstName + " " +
            employees_obj.employees[i].lastName + "<br>");
      }
      document.writeln("</p>");
    });
  </script>
</head>

<body>
</body>

```

```
</html>
```

JSON & JavaScript

- Because of JSON's deep connection with Javascript objects, we can actually declare Javascript objects using JSON syntax without needing to parse
  - Javascript natively supports the ability to create objects using JSON syntax
  - Example for our employees:

Example: `employee_js.html`

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title>Employee JSON example</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/
     jquery.min.js"></script>
  <script>
    // run when the document is ready
    $(document).ready(function() {
      // create JS object using JSON syntax
      var employees = [{ "firstName": "John","lastName": "Doe" }, {"
         firstName": "Anna", "lastName": "Smith"}, {"firstName": "Peter
         ","lastName": "Jones" }];

      // setup document
      document.writeln("<h2>Employees list</h2>");
      document.writeln("<p>");
      // loop over every employee
      for (var i = 0; i < employees.length; i++) {
        // print employee info
        document.writeln(employees[i].firstName + " " + employees[i].
           lastName + "<br>");
      }
      document.writeln("</p>");
    });
  </script>
</head>

```

```
<body>
</body>

</html>
```