JSON

Syed Gillani

[What is JSON]

- A lightweight text based data-interchange format
- Completely language independent
- Based on a subset of the JavaScript Programming Language
- Easy to understand, manipulate and generate
    - **It is NOT:**

    ✓ Overly Complex

    ✓ A “document” format

    ✓ A markup language

    ✓ A programming language

[Why JSON]

- Straightforward syntax
- Easy to create and manipulate
- Can be natively parsed in JavaScript using **eval()**
- Supported by all major JavaScript frameworks
- Supported by most backend technologies

Contains only two main types of elements:

✓ Objects (key->value)

✓ Arrays of ordered objects

[JSON Object Syntax]

- Unordered sets of name/value pairs
- Begins with **{** (left brace)
- Ends with **}** (right brace)
- Each name is followed by **:** (colon)
- Name/value pairs are separated by **,** (comma)

e.g. {"employee_id": 1234567}

[JSON Example]

employee_id	name	hire_date	location	consultant
..	..	...	..	..
...	..	...	..	..
..	..	..	..	..
..	..	..	..	...

```
var employeeData = {
 "employee_id": 1234567,
 "name": "Jeff Fox",
 "hire_date": "1/1/2013",
 "location": "Norwalk, CT",
 "consultant": false
};
```

[Array in JSON]

- An ordered collection of values
- Begins with **[** (left bracket)
- Ends with **]** (right bracket)
- Name/value pairs are separated by **,** (comma)

```
"random_nums": [ 24,65,12,94 ]
```

[Array in JSON]

```
var employeeData = {
 "employee_id": 1236937,
 "name": "Jeff Fox",
 "hire_date": "1/1/2013",
 "location": "Norwalk, CT",
 "consultant": false,
  "random_nums": [ 24,65,12,94 ]
};
```

[Data Types]

- **General**:
  - ✓ Sequence of one or more Unicode characters
  - ✓ Wrapped in “double quotes”
  - ✓ Backslash escapement

- **Numbers**:
  - ✓ Integer
  - ✓ Real
  - ✓ Scientific
  - ✓ No octal or hex
  - ✓ No NaN or Infinity – Use **null** instead.

- **Boolean**:
  - ✓ True and False
  - ✓ Null: representing nothing or a value as nothing

[Therefore JSON is]

- Lacks namespaces (as utilised in XML and RDF)
- No inherent validation (XML has DTD and XSD..)

- This Makes it:

✓ Lighter and faster than XML/RDF

✓ Less syntax and no semantics

✓ Can easily be parsed in an object-oriented language (e.g. Java, Javascript)

It is extensibly used in Today's Web

[Fin]

★Acknowledgments: *The Web*