Usage Guidelines

Do not forward this document to any non-Infosys mail ID. Forwarding this document to a non-Infosys mail ID may lead to disciplinary action against you, including termination of employment.

Contents of this material cannot be used in any other internal or external document without explicit permission from ETA@infosys.com.

Introduction to JSON

Education, Training and Assessment
We enable you to leverage knowledge anytime, anywhere!

Infosys® | Building Tomorrow's Enterprise

Copyright Guideline

Confidential Information

- This Document is confidential to Infosys Limited. This document contains information and data that Infosys considers confidential and proprietary ("Confidential Information").
- Confidential Information includes, but is not limited to, the following:
  - Corporate and Infrastructure information about Infosys
  - Infosys' project management and quality processes
  - Project experiences provided included as illustrative case studies
- Any disclosure of Confidential Information to, or use of it by a third party, will be damaging to Infosys.
- Ownership of all Infosys Confidential Information, no matter in what media it resides, remains with Infosys.
- Confidential information in this document shall not be disclosed, duplicated or used – in whole or in part – for any purpose other than reading without specific written permission of an authorized representative of Infosys.
- This document also contains third party confidential and proprietary information. Such third party information has been included by Infosys after receiving due written permissions and authorizations from the party/ies. Such third party confidential and proprietary information shall not be disclosed, duplicated or used – in whole or in part – for any purpose other than reading without specific written permission of an authorized representative of Infosys.

Course Information

Course Code: ERJEEML965

Course Name: Introduction to JSON

Document Number: 007

Version Number: 1.0

Session Plan

- Introduction to JSON
- JSON implementations
- JSON Data types
- JSON Schema
- JSON in Java
- JSON with Ajax
- Debugging JSON

Introduction to JSON

Education, Training and Assessment
We enable you to leverage knowledge anytime, anywhere!

Infosys® | Building Tomorrow's Enterprise

JavaScript Object Notation

- JSON is a light weight data interchange format created by Douglas Crockford.
- It follows JavaScript Object literal notation.
- It is language-independent, with parsers available for many languages.
- It is an alternative to XML format
- JSON is used to exchange data between server and web application

JavaScript Object Literals

- Array literals are formed with square brackets:

```
var Names= ["Paul","John","George","Ringo"];
```

- This is equivalent to:

```
var Names= new Array("Paul","John","George","Ringo");
```

- Object literals are formed with curly brackets:

```
var Omega={
"Country":"India",
"YearFormed":1947
}
```

Omega is an object with two properties Country and YearFormed

JSON Syntax

- JSON uses Javascript object literal syntax

```
"name":"value"
```

- JSON data is written in the form of name/value pairs

- JSON object is placed in curly braces
- Example:

```
{"BookName":"JSON" , "Price":200}
```

- The above example is a single JSON object with two fields as "BookName" and "Price"

JSON Arrays

- JSON Arrays are written in square brackets

- Example:

```
"Books": [
            {"BookName":"JSON" , "Price":200},
            {"BookName":"Ajax" , "Price":350}
          ]
```

- The above example is a single JSON Array called Books which contains two objects

JSON in JavaScript

- As JSON follows JavaScript object literal syntax, we can directly use JSON in JavaScript.
- Example:

```
var book=  {"BookName":"JSON" , "Price":200};
```

- Use simple JavaScript eval() function to parse JSON in JavaScript

```
var jsonData=eval(book);
```

- Now the values can be accessed as

```
jsonData.BookName
jsonData.Price
```


As eval() function has security concerns, it is safe to use JSON.parse() method to parse JSON in JavaScript.

Example

```
<html>
<head>
<script>
var book=  {"BookName":"JSON" , "Price":200};
var jsonData=eval(book);
Console.log(jsonData.BookName);
Console.log(jsonData.Price);
</script>
</head>
</html>
```

A JSON object is created called book with two properties BookName and Price

Evaluate the JSON object using eval() function

Access the JSON object properties using dot operator

JSON Implementations

Education, Training and Assessment
We enable you to leverage knowledge anytime, anywhere!

Infosys | Building Tomorrow's Enterprise

JSON Implementations

- We have used JSON as a data interchange format
- But there are other alternate implementations of JSON
- JSON can also used for
  - Dependency Management
  - Storing metadata

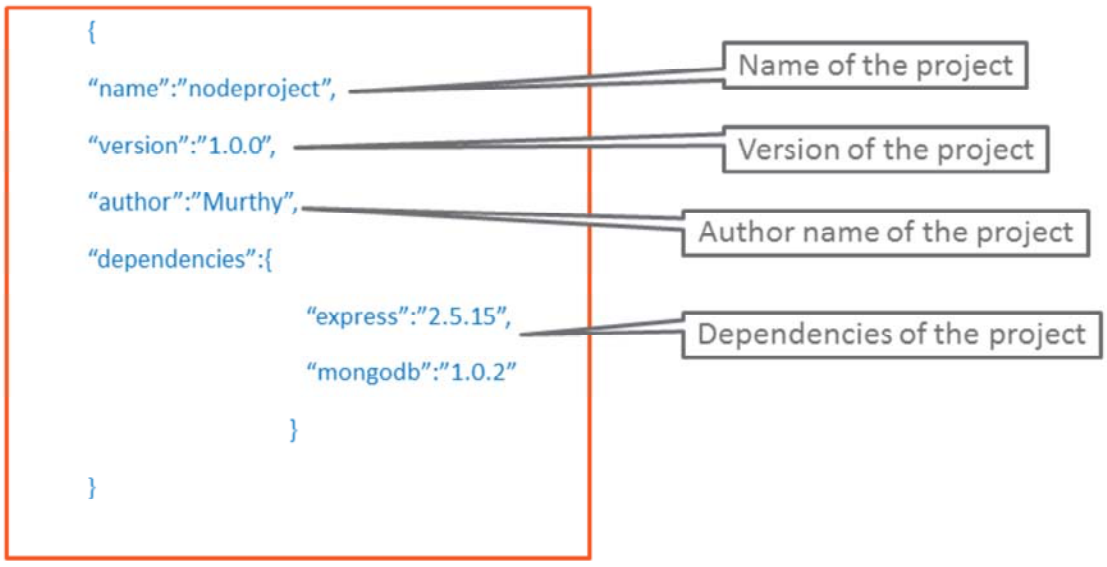
Dependency Management

- In real time scenarios, most of the frameworks we use for creating projects will have dependencies also to be loaded
- These dependencies will be usually stored in a separate file in JSON format
- Example:
  - Node.js uses JSON for tracking dependencies.
  - For every Node.js project, there will be a package.json file that hold all the dependencies

Metadata…(1)

- JSON is also used to store metadata of projects
- Prior to JSON, metadata was either stored in a text file or in a language specific file such as config.php for PHP, config.js for JavaScript etc.,
- Now we use config.json in most of the languages to store metadata
- Example:
  - In Node.js, package.json will be the metadata file
  - Sample package.json format is given in the next slide

Metadata…(2)



Comparison of JSON with XML and YAML …(1)

- JSON

Advantages	Disadvantages
Simple Syntax, less markup	Only handful data types supported
Easy to use as it is subset of Javascript	Doesn't support object reference

Comparison of JSON with XML and YAML …(2)

- JSON Example

```
{
        "name":"George",
        "age":40
        "address":{
                "city":"Bangalore",
                "state":"Karnataka"
        }
}
```

Comparison of JSON with XML and YAML …(3)

- XML

Advantages	Disadvantages
Generalized markup	Relatively wordy compared to JSON
Xml schema for datatype, Can create new datatypes.	
XSLT for transformation into different output Xpath for extracting information from nested structures	

Comparison of JSON with XML and YAML …(4)

- XML Example

```
<name>George</name>
<age>40</age>
<address>
        <city>Bangalore</city>
        <state>Karnataka</state>
</address>
```

Comparison of JSON with XML and YAML …(5)

- YAML – Yet Another Markup Language
- The purpose of YAML, is to signify typical data types in human-readable notation which is similar to JSON
- YAML is superset of JSON, with many more capabilities (lists, casting, etc.)

Limitations:

- YAML doesn't handle escaped Unicode characters
- Therefore, JSON can be parsed by YAML parsers

- **If JSON isn't enough, consider YAML**

Comparison of JSON with XML and YAML …(6)

- YAML Example

```
#An Employee Record
Name:George
Age:40
Address:
        city:Bangalore
        state:Karnataka
```

JSON Data Types

Education, Training and Assessment
We enable you to leverage knowledge anytime, anywhere!

Infosys | Building Tomorrow's Enterprise

JSON Data Types

Data Types	Representation
Number	Double precision floating point number
String	Set of Unicode characters
Boolean	True or False
Array	An ordered sequence of values
Object	An unordered collection of key-value pairs
null	empty

JSON Schema

Education, Training and Assessment
We enable you to leverage knowledge anytime, anywhere!

Infosys | Building Tomorrow's Enterprise

JSON Schema

- JSON schema specifies the structure of the JSON data
- Latest version of JSON schema is V4
- It is clear and human readable documentation
- It is useful for automated testing
- It is useful for validating client submitted data

JSON Schema Keywords … (1)

- Following are the list of keywords to be used in the JSON schema
  - $schema – Specifies that the schema is written using draft v4 specification
  - title – Specifies the title of the schema
  - description – Specifies the description of the schema
  - type – specifies the first constraint of the JSON data; it has to be a JSON data
  - properties – specifies various properties to be used in the schema; values and their data types
  - required – specifies the mandatory properties to be defined in the schema
  - minimum – specifies the minimum value for a property defined in the schema

JSON Schema Keywords … (2)

- exclusiveMinimum – if it is set to true, the instance is valid if its value is strictly greater than the minimum value
- Maximum – specifies the maximum value for a property defined in the schema
- exclusiveMaximum - if it is set to true, the instance is valid if its value is strictly lower than the maximum value
- maxLength – specifies the maximum number of characters to be used for a string property
- minLength – specifies the minimum number of characters to be used for a string property
- Pattern – specifies the pattern for a string property
- multipleOf – numbers entered should be multiples of the given number

Example…(1)

- Schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Product",
  "description": "Product details",
  "type": "object",
"properties": {
    "id": {
      "description": "unique product identifier",
      "type": "integer"
    },
"name": {
      "description": "Product Name",
      "type": "string"
    },
"price": {
      "type": "number",
      "minimum": 0,
      "exclusiveMinimum": true
    },
"tags": {
      "type": "array",
      "items": {
        "type": "string"
      },
      "minItems": 1,
      "uniqueItems": true
    }
  },
  "required": ["id","name","price"]
}
```

Example…(2)

- Data

```
[  {
    "id": 1,
    "name": "An ice sculpture",
    "price": 11.50,
    "tags": ["cold", "ice"]
    },
  {
    "id": 3,
    "name": "A blue mouse",
    "price": 25.50,
      }  ]
```

JSON in Java

Education, Training and Assessment
We enable you to leverage knowledge anytime, anywhere!

Infosys | Building Tomorrow's Enterprise

Java JSON Libraries

- There are different JSON libraries available in Java.
- Following are few famous libraries
  - JSON.simple
  - JSON-taglib
  - Jackson
  - Google Gson

JSON.simple

- It is a simple Java library for JSON
- This library contains following classes

JSONObject	JSONArray	JSONStringer	JSONWriter
JSONTokener	JSONException	JSONString	

JSONObject

- It is an unordered collection of key-value pairs.
- It has get() and opt() to access the values by key
- It has put() methods for adding or replacing values by key

JSONArray

- It is an ordered sequence of values.
- It has get and opt() methods to access values by index
- It has put() methods for adding or replacing values
- The values can be of any type

Other Classes

Class	Definition
JSONStringer	Produces JSON text
JSONWriter	For writing JSON text to streams
JSONTokener	Accepts string as input and extracts characters and tokens from it
JSONException	Thrown when a syntax error occurs in JSON data
JSONString	Represents JSON string value. It contains methods to convert data to string.

Writing JSON data

```
public class WriteJSON {

public static void main(String[] args) {

 JSONObject obj = new JSONObject();

Obj.put("name", "Kishore");

Obj.put("age", new Integer(30));

 JSONArray list = new JSONArray();

List.add("msg 1");

List.add("msg 2");

List.add("msg 3");

 obj.put("messages", list);
```

```
try {

FileWriter file = new FileWriter("d:\\test.json");

file.write(obj.toJSONString());

file.flush();

file.close();

 } catch (IOException e) {

e.printStackTrace();

}

 System.out.print(obj);

    }

}
```

Reading JSON data

```
public class ReadJSON {

public static void main(String[] args) {

 JSONParser parser = new JSONParser();

 try {

 Object obj = parser.parse(new FileReader("d:\\test.json"));

 JSONObject jsonObject = (JSONObject) obj;

 String name = (String) jsonObject.get("name");

System.out.println(name);

 long age = (Long) jsonObject.get("age");

System.out.println(age);
```

```
// loop array

JSONArray msg = (JSONArray)
jsonObject.get("messages");

Iterator<String> iterator = msg.iterator();

while (iterator.hasNext()) {

System.out.println(iterator.next());

} } catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {e.printStackTrace();

} catch (ParseException e)
{e.printStackTrace();}

 } }
```

JSON-taglib

- JSON-taglib is a JSP 2.0 library for processing JSON data in a JSP page
- <json:object> is used to create a JSON object in a jsp page
- Example

```
<json:object>
<json:property name="a" value="10"/>
<json:property name="b" value="20"/>
</json:object>
```

Jackson

- Jackson is a High-performance JSON processor Java library.
- Jackson data binding will help to convert Java object to / from JSON.
- Following APIs are used to write and read JSON data
  - JsonGenerator – Write to JSON.
  - JsonParser – Parse JSON.

GSON

- Gson is a Java library which is used to convert Java objects to JSON
- It contains following methods
  - toJSON() -> converts Java object to JSON
  - fromJSON() -> converts JSON to Java object
- Gson can also convert pre-existing unmodifiable objects
- Example:

```
Gson g=new Gson();
```

GSONBuilder

- We can use GsonBuilder class to create Gson instance with different configuration settings instead of using default
- Example

```
Gson g=new GsonBuilder()
        .setDateFormat(dateFormat.LONG)
        .setVersion(1.0)
        .create();
```

JSON with Ajax

Education, Training and Assessment
We enable you to leverage knowledge anytime, anywhere!

Infosys | Building Tomorrow's Enterprise

JSON with Ajax

- Ajax is a set of several technologies which handles asynchronous communication
- Ajax can receive JSON responses also.

Ajax code

```
var XHR;
if (window.ActiveXObject) {//For Microsoft Browsers
XHR=new ActiveXObject("Microsoft.XMLHTTP");
}
else if (window.XMLHttpRequest){//For Mozilla and Non Microsoft Browsers
XHR=new XMLHttpRequest();
}
XHR.open("GET", "data.json" ,true);
XHR.onreadystatechange = function(){displayMessage(XHR);};
XHR.send();
```

Parsing JSON Response

- After receiving JSON response in Ajax application, it has to be converted into JSON object in JavaScript.
- To convert a JSON text into an JSON object, use the eval() function.
- eval() invokes the JavaScript compiler
- Since JSON is a proper subset of JavaScript, the compiler will correctly parse the text and produce an object structure
- Eval() is not recommended in all cases as it can evaluate any JavaScript expression
- Instead of eval(), we can use JSON.parse() the parse the JSON data exactly

Example to parse json response in ajax

```
function displayMessage(XHR)        {
        if(XHR.readyState==4)                {
                if(XHR.status==200)                        {
                        var d=JSON.parse(XHR.responseText);
                        document.getElementById("error").innerHTML=d[0].name;
                        document.getElementById("error").style.display="block";
                }else{
                        document.getElementId("error").innerHTML=XHR.statusText;
                        document.getElementById("error").style.display="block";
                } } }
```

Cross Domain Ajax Calls

- In some scenarios, we would like to retrieve data from different domains
- In such cases, we will get cross domain error
- Browsers use a policy called as same domain policy which is a security measure followed in order to restrict one domain from accessing information from another domain
- Same domain policy looks for three things in incoming request;host, port and protocol
- If anyone of them is different from the existing domain, the request will not be completed and a cross domain error will be returned

JSONP

- We can use JSONP(JSON with padding) to get around the same domain policy.
- One exception under the same origin policy is the <script> tag. So scripts can be passed across domains.
- JSONP uses this exception in order to pass data across domains as a script by adding padding to make the JSON object look like a script.
- With JSONP, we pass the JSON data as a parameter to a function; thereby, we pad our object into a function callback.

Example

```
Mycallback({
"empid":1.
"name":"Murthy"
});
```

Wrap json data with any name. In this example, data is wrapped with name called as Mycallback

- In the fetch url , we need to mention a callback parameter as

  http://www.example.com/index.jsp?callback=Mycallback

- In this example, we are padding the employees object into mycallback function and we have to use mycallback function in order to retrieve employees object

In JS frameworks, we mention datatype parameter as jsonp to make cross domain calls.

Debugging JSON

Education, Training and Assessment
We enable you to leverage knowledge anytime, anywhere!

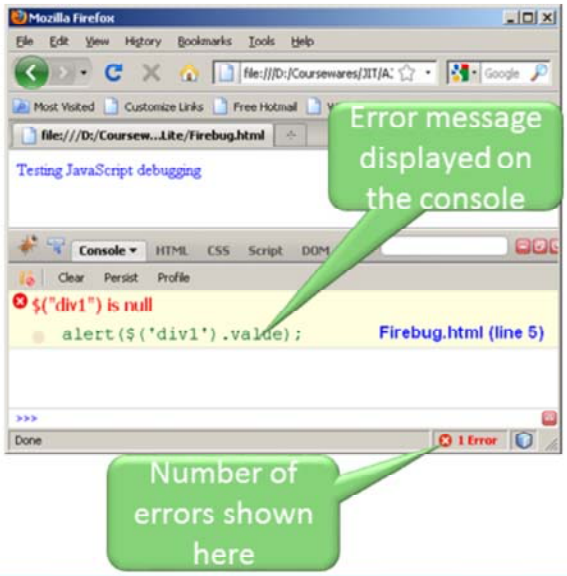
Infosys® | Building Tomorrow's Enterprise

Debugging JSON

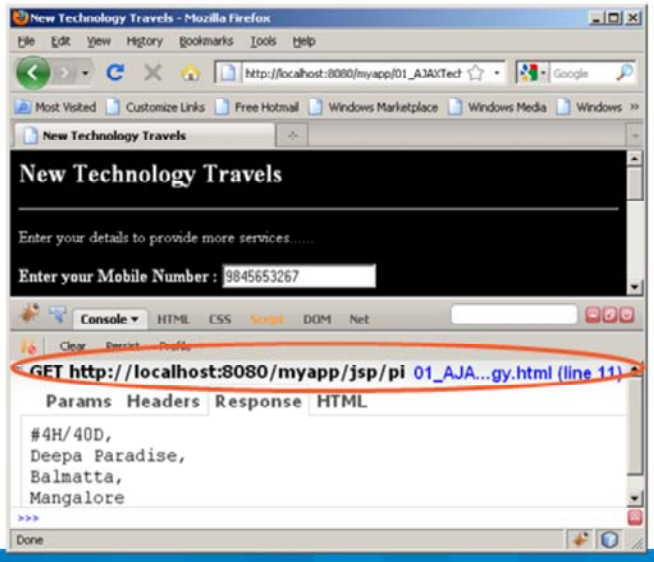
- Almost all of the top browsers, such as Mozilla Firefox, Google Chrome, Safari, and Internet Explorer, have powerful debugging tools that help us understand the requests that are being made, and the responses that are coming back.
- JSON could either be part of the request, or be part of the response.
- Firebug is a very popular web development toolkit that is available for Mozilla Firefox.
- Firebug is an external plugin and has to be installed on the browser

JavaScript Errors

- After the Firebug is installed successfully, press F12 to open Firebug which displays a small window right below the page.
- Clicking on a error will take the control to the actual line causing the error in the code



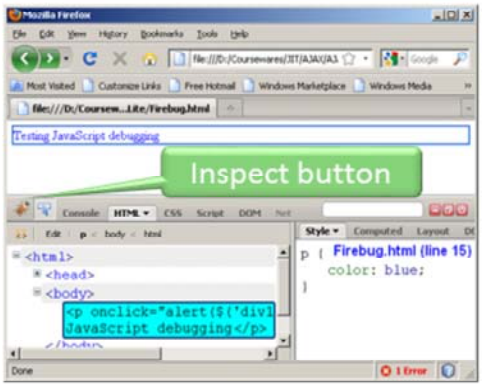
XMLHttpRequest Monitoring



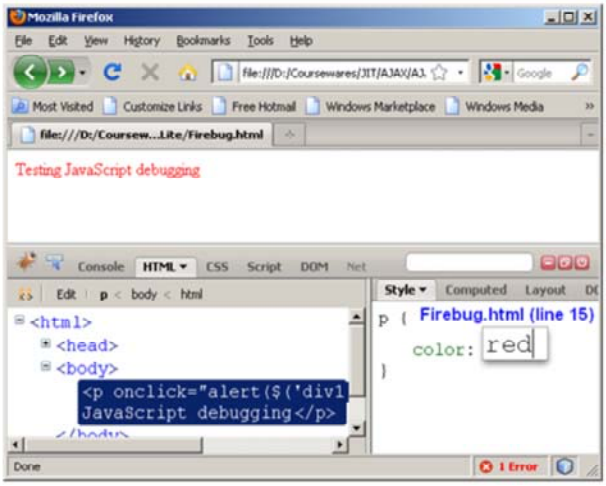
- Track the Ajax call from the page.
- The Ajax failure is indicated by highlighting it in red

Inspect HTML

- Clicking on the inspect button will allow user to mouseover any element on the page and highlights the corresponding code
- This will help in determining the code, causing display of a specific element in a massive webpage



Edit Code Live



- The greatest advantage of Inspect is associated with edit. Firebug allows editing both HTML and CSS live!

We can also debug JavaScript by setting break points. The keys F11, F10 and shift F11 can be used to step into, step over and step out respectively. We can also have a watch of property value changes in **DOM** option, Monitor Network Activity by determining the time taken to load every js and image files by the application in **Net** option etc.

More on this is at – http://getfirebug.com/whatisfirebug

Validating JSON

- There are so many online tools available to validate JSON data.
- Jsonlint.com is one of the famous web based json validator.
- Jsonschemalint.com is another web based json validator where we can validate json data with schema
- There are several json validators available for different programming languages
- For example, json-schema-validator(Java), JSV(JavaScript),json.net(.net) etc.,

Validating JSON data using jsonlint

Formatting JSON

- JSONLint is not just an online JSON validator, it also helps us format JSON and makes it look pretty.
- Often JSON feeds are big in size, and an online editor that provides a tree structure to traverse through the JSON object is always helpful.
- **JSON Editor Online** is an online editor to format the big JSON objects, as it provides an easy to navigate tree

Summary

- Introduction to JSON
- JSON implementations
- JSON Data types
- JSON Schema
- JSON in Java
- JSON with Ajax
- Debugging JSON


Thank You
© 2012 Infosys Limited, Bangalore, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.
Infosys
Building
Tomorrow's Enterprise