JSON BASICS

**JavaScript Object Notation [JSON]**

**1. Examples of Data Formats:**

**String**

String str1 = "123praveen200.12";

String str2 = "EmpID=123|EmpNM=praveen|EmpSal=200.12";

**XML**

<employee>

<emp-id>123</emp-id>

<emp-name>Praveen</emp-name>

<emp-salary>200.12</emp-salary>

</employee>

**JSON**

{"EmpID":123, "EmpNM":"Praveen", "EmpSal":200.12}

* Like XML, JSON also is a "Programing Language & Platform Independent Language" which helps to store and transport data
* However compared to XML, it’s a lightweight, easy for applications to parse and generate by avoiding complicated parsing and translations
* JSON is a "text format" but uses conventions that are familiar to programmers of the C-family of languages (C, C++, C#, Java, JavaScript, Perl, Python, etc.,). Hence JSON is an "ideal data interchange language"
* JSON, as the name implies, which consists of data similar to "Object Notation of JavaScript". It's an extension of JavaScript scripting language and this format was specified by "Douglas Crockford in 2006"
* Hence if we receive data from a server in JSON format, we can directly use it like any other JavaScript object
* The filename extension of JSON is “.json"
* MIME type (Content Type) of JSON is "application/json"

**2. JSON Syntax**

JSON syntax is derived from JavaScript object notation syntax:

- Data is in "name:value" pairs

- Data is separated by "commas"

- "Curly braces" hold objects

- "Square brackets" hold arrays

**i. JSON Data**

JSON data is written as name/value pairs. A name/value pair consists of

- a field name (**Should be** in double quotes)

- followed by a colon

- followed by a value

**Ex:** "employee-name" : "Praveen D"

**ii. JSON Values**

 In JSON, values must be one of the following data types

1. String

2. Number

3. Boolean

4. NULL

5. an Object (JSON object)

6. an Array

* In JSON,

- String values must be written with double quotes

- Numbers must be an integer/decimal values

- Boolean values must be true/false

- JSON NULL values must be null

* **Ex:-**

{ "name":"Praveen D",

"age":33,

"isEmployed":true,

"girlFriend":null

}

**iii. JSON Objects**

* Values in JSON can be objects
* JSON Objects are

- surrounded by curly braces {}

- JSON object data is written in "key:value" pairs

- Each "key:value" pair is separated by a comma

- Keys must be String and Values must be a valid

- JSON data type (String, Number, Object, Array, Boolean or null)

* **Ex:-**

{

"employee":{ "name":"Praveen D",

"age":33,

"isEmployed":true,

"girlFriend":null

}

}

* Values in a JSON object can be another JSON object

{

"employee": {

"name":"Praveen D",

"age":33,

"isEmployed":true,

"girlFriend":null,

"cars": {

"car1":"GM",

"car2":"BMW",

"car3":"Audi"

}

}

}

**JSON Arrays**

Values in JSON can be arrays

JSON Arrays are

- surrounded by "Square Brackets []"

- JSON Arrays values is separated by a comma

- Array values must be a valid JSON data type

- (String, Number, Object, Array, Boolean or null)

Example 1:-

{

"employees":[ "Praveen", "Rekha", "Malleshwar" ]

}

Example 2:-

{

"name":"Praveen",

"age":33,

"cars":[ "GM", "BMW", "Audi" ]

}

Values in an array can also be another array, or even another JSON object:

{

"name":"Praveen",

"age":33,

"cars": [

{ "name":"GM",

"models":[ "Aveo", "Beat", "Cruze" ]

},

{ "name":"Audi",

"models":[ "A3", "A7" ]

}

]

}

**JSON v/s XML**

Both JSON and XML can be used to store & get the data from a web server

However,

- XML has to be parsed with an XML parser

- JSON can be parsed by a standard JavaScript function

Hence, XML is much more difficult to parse than JSON

JSON is a ready-to-use JavaScript object

XML Data cannot consist of Arrays

JSON Data can consist of Arrays

In General,

- Web Applications interact with each other by exchanging data using XML

- Mobile Apps interact with Web Applications by exchanging data using JSON