Introduction to Object Oriented JavaScript

JavaScript with JSON

Capgemini

Lesson Objectives

JSON Object

JSON.stringify and JSON.parse

JSON Introduction

JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax.

It is commonly used for transmitting data in web applications (e.g., sending some data from the server to the client, so it can be displayed on a web page, or vice versa).

A JSON object can be stored in its own file, which is basically just a text file with an extension of .json, and a MIME type of application/json.

JSON is purely a data format — it contains only properties, no methods.

JSON requires double quotes to be used around strings and property names. Single quotes are not valid.

JSON Introduction

Even a single misplaced comma or colon can cause a JSON file to go wrong, and not work.

We can validate JSON using an application like JSONLint.

JSON can actually take the form of any data type that is valid for inclusion inside JSON, not just arrays or objects. So for example, a single string or number would be a valid JSON object.

Unlike in JavaScript code in which object properties may be unquoted, in JSON, only quoted strings may be used as properties.

JSON Type

Number : integer, real or floating point

String : double-quoted Unicode with backslashes

Boolean : true and false

Array : ordered sequence of comma-separated values enclosed in square brackets

Object : collection of comma-separated “key”:value pairs enclosed in curly braces

null

JSON Object Notation

A JSON object is an unordered set of name/value pairs

- A JSON object begins with { (left brace) and ends with } (right brace)
- Each name is followed by : (colon) and the name/value pairs are separated by , (comma) and enclosed with in quotes.

The JSON.parse function deserializes JSON text to produce a JavaScript value.

```
var data = {"Name":"Abcd", "age":55}

var dataparsed = eval(data);

console.log(dataparsed.Name);
console.log(dataparsed.age);
```

JSON Object Notation

The JSON.stringify function serializes a JavaScript value to JSON text.

```
function Employee(name, age, salary) {
    this.Name = name;
    this.age = age;
    this.salary = salary;
}

var employeeObject = new Employee('Abcd',25,5118);

console.log(employeeObject);
```

Demo

Demo1
Demo2

Lab

Lab 3

Summary

In this lesson we have learned about –

JSON Object
JSON.stringify and JSON.parse