# **Experiment No. 07**

Title: To implement data handling with JSON

Batch:SY-IT(B3) Roll No.:16010423076 Experiment No.:7

**Aim**: To Implement data handling with JSON.

Resources needed: Notepad++, Web Browser

#### Theory:

JSON stands for JavaScript Object Notation. JSON is a **text format** for storing and transporting data. JSON is "self-describing" and easy to understand

- JSON stands for JavaScript Object Notation
- JSON is a lightweight data-interchange format
- JSON is plain text written in JavaScript object notation
- JSON is used to send data between computers
- JSON is language independent \*

### Why Use JSON?

- The JSON format is syntactically similar to the code for creating JavaScript objects.
   Because of this, a JavaScript program can easily convert JSON data into JavaScript objects.
- Since the format is text only, JSON data can easily be sent between computers, and used by any programming language.
- JavaScript has a built in function for converting JSON strings into JavaScript objects: **JSON.parse**()
- JavaScript also has a built in function for converting an object into a JSON string: JSON.stringify()

Both JSON and XML can be used to receive data from a web server.

#### JSON Example

```
{"employees":[
    { "firstName":"John", "lastName":"Doe" },
    { "firstName":"Anna", "lastName":"Smith" },
    { "firstName":"Peter", "lastName":"Jones" }
}
```

#### JSON.stringify()

- When sending data to a web server, the data has to be a string.
- Convert a JavaScript object into a string with JSON.stringify().
- Stringify a JavaScript Object

Imagine we have this object in JavaScript:

```
const obj = {name: "John", age: 30, city: "New York"};
```

Use the JavaScript function JSON.stringify() to convert it into a string.

```
const myJSON = JSON.stringify(obj);
```

The result will be a string following the JSON notation.

myJSON is now a string, and ready to be sent to a server:

## **Example**

```
const obj = {name: "John", age: 30, city: "New York"};
const myJSON = JSON.stringify(obj);
```

#### JSON.parse()

A common use of JSON is to exchange data to/from a web server. When receiving data from a web server, the data is always a string. Parse the data with JSON.parse(), and the data becomes a JavaScript object.

## **Example - Parsing JSON**

Imagine we received this text from a web server:

```
'{"name":"John", "age":30, "city":"New York"}'
```

#### Use the JavaScript function JSON.parse() to convert text into a JavaScript object:

```
const obj = JSON.parse('{"name":"John", "age":30, "city":"New York"}');
```

Make sure the text is in JSON format, or else you will get a syntax error.

#### **Use the JavaScript object in your page:**

#### **Example**

```
<script>
```

```
document.getElementById("demo").innerHTML = obj.name;
</script>
```

Date objects are not allowed in JSON. If you need to include a date, write it as a string.

You can convert it back into a date object later:

#### **Example**

Convert a String into date

```
const text = '{"name":"John", "birth":"1986-12-14", "city":"New York"}';
const obj = JSON.parse(text);
obj.birth = new Date(obj.birth);
```

document.getElementById("demo").innerHTML = obj.name + ", " + obj.birth;

#### **Storing Data**

When storing data, the data has to be a certain format, and regardless of where you choose to store it, *text* is always one of the legal formats.

JSON makes it possible to store JavaScript objects as text.

## Example

## Storing data

```
// Storing data:
const myObj = {name: "John", age: 31, city: "New York"};
const myJSON = JSON.stringify(myObj);
localStorage.setItem("testJSON", myJSON);

// Retrieving data:
let text = localStorage.getItem("testJSON");
let obj = JSON.parse(text);
document.getElementById("demo").innerHTML = obj.name;
```

## **JSON Server**

#### **Sending Data**

If you have data stored in a JavaScript object, you can convert the object into JSON, and send it to a server:

#### **Example**

```
const myObj = {name: "John", age: 31, city: "New York"};
const myJSON = JSON.stringify(myObj);
window.location = "demo_json.php?x=" + myJSON;
```

## **Receiving Data**

If you receive data in JSON format, you can easily convert it into a JavaScript object:

#### **Example**

```
const myJSON = '{"name":"John", "age":31, "city":"New York"}';
const myObj = JSON.parse(myJSON);
document.getElementById("demo").innerHTML = myObj.name;
```

#### JSON HTML

#### **HTML Table**

Make an HTML table with data received as JSON:

## Example

#### **HTML Drop Down List**

Make an HTML drop down list with data received as JSON:

#### **Example**

```
const dbParam = JSON.stringify({table:"customers",limit:20});
const xmlhttp = new XMLHttpRequest();
xmlhttp.onload = function() {
  const myObj = JSON.parse(this.responseText);
  let text = "<select>"
  for (let x in myObj) {
    text += "<option>" + myObj[x].name + "</option>";
  }
  text += "</select>"
  document.getElementById("demo").innerHTML = text;
  }
}
xmlhttp.open("POST", "json_demo_html_table.php", true);
xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xmlhttp.send("x=" + dbParam);
```

#### **Activity:**

- 1. Convert JSON objects into string using JSON.sringify().
- 2. Replace any data in JSON object JSON.repalce()
- 3. Valid JSON sting into JSON using JSON.parse()

#### **Results:** (Program printout with output)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JSON Operations Demo</title>
  <style>
    body {
       font-family: "Arial", sans-serif;
       margin: 40px;
       text-align: center;
       background-color: #f8f8f8;
       color: #333:
    h2 {
       font-size: 28px;
       text-transform: uppercase;
       letter-spacing: 1px;
       font-weight: bold;
```

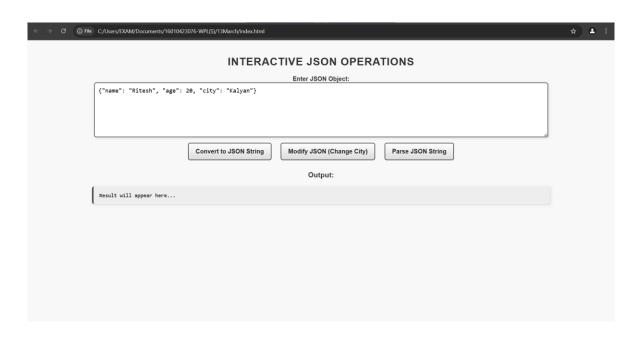
```
margin-bottom: 20px;
textarea {
  width: 80%;
  height: 120px;
  font-size: 16px;
  padding: 10px;
  border: 2px solid #444;
  border-radius: 6px;
  outline: none;
  background: #fff;
  transition: 0.3s;
  box-shadow: 2px 2px 6px rgba(0, 0, 0, 0.1);
textarea:focus {
  border-color: #000;
  box-shadow: 3px 3px 8px rgba(0, 0, 0, 0.2);
button {
  margin: 10px;
  padding: 12px 18px;
  font-size: 16px;
  cursor: pointer;
  border: 2px solid #333;
  background: linear-gradient(to bottom, #fff, #ddd);
  color: #222;
  border-radius: 6px;
  font-weight: bold;
  transition: all 0.3s ease;
button:hover {
  background: #333;
  color: #fff;
  transform: translateY(-2px);
}
pre {
  background: #eee;
  padding: 15px;
  border-radius: 6px;
  text-align: left;
  width: 80%;
  margin: auto;
  font-size: 14px;
```

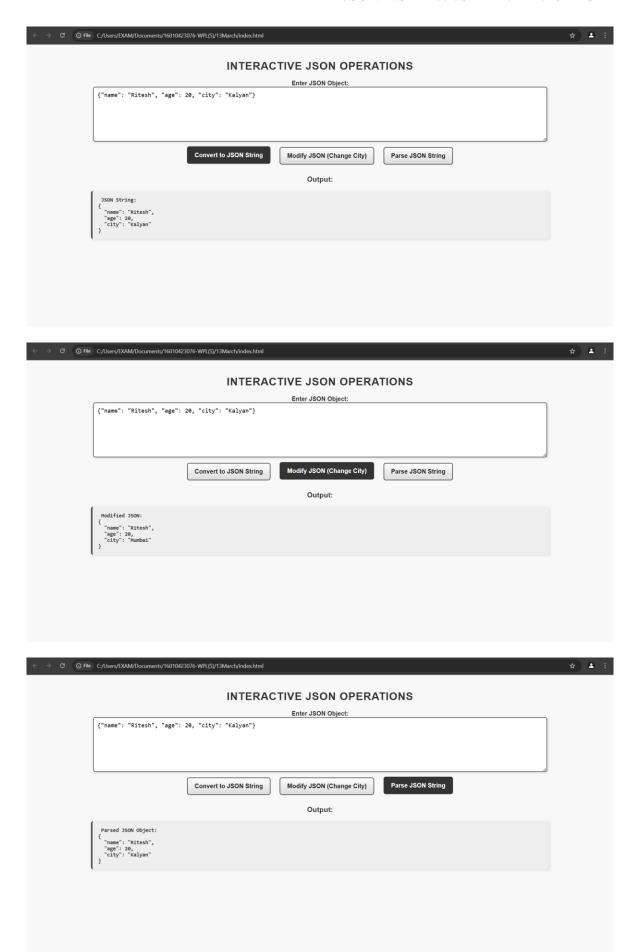
```
font-weight: bold;
       border-left: 4px solid #444;
       box-shadow: 2px 2px 6px rgba(0, 0, 0, 0.1);
       transition: 0.3s;
    pre:hover {
       background: #ddd;
       box-shadow: 3px 3px 8px rgba(0, 0, 0, 0.2);
  </style>
</head>
<body>
  <h2>Interactive JSON Operations</h2>
  <label for="jsonInput"><strong>Enter JSON Object:</strong></label>
  <br>>
  <textarea id="jsonInput">{"name": "Ritesh", "age": 20, "city": "Kalyan"}</textarea>
  <br>
  <button onclick="convertToJSONString()">Convert to JSON String</button>
  <button onclick="replaceDataInJSON()">Modify JSON (Change City)</button>
  <button onclick="parseJSONString()">Parse JSON String/button>
  <h3>Output:</h3>
  Result will appear here...
  <script>
    function getJSONInput() {
       try {
         return JSON.parse(document.getElementById("jsonInput").value);
       } catch (error) {
         document.getElementById("output").innerText = "□ Invalid JSON input!";
         throw new Error("Invalid JSON input");
    }
    function convertToJSONString() {
       let jsonObject = getJSONInput();
       let jsonString = JSON.stringify(jsonObject, null, 2);
       document.getElementById("output").innerText = " JSON String:\n" + jsonString;
    function replaceDataInJSON() {
       let jsonObject = getJSONInput();
      jsonObject.city = "Mumbai"; // Modify JSON (Changing city)
```

```
document.getElementById("output").innerText = " Modified JSON:\n" + JSON.stringify(jsonObject, null, 2);
}

function parseJSONString() {
    let jsonString = document.getElementById("jsonInput").value;
    try {
        let parsedObject = JSON.parse(jsonString);
        document.getElementById("output").innerText = " Parsed JSON Object:\n" + JSON.stringify(parsedObject, null, 2);
    } catch (error) {
        document.getElementById("output").innerText = " \subseteq Error: Invalid JSON String!";
    }
}
</script>

</body>
</html>
```





#### **Questions:**

- 1. Why Jason is better than xml?
  - JSON is better than XML because it is simpler, lighter, and faster. JSON uses a smaller size and is easier to read and write, while XML has a lot of extra tags that make it bulkier. JSON is also faster to parse in programming languages like JavaScript because it directly works as a JavaScript object, whereas XML needs extra processing. JSON is widely used in web APIs and modern applications because of its efficiency and ease of use.
- 2. Write difference between JSON and Javascript JSON (JavaScript Object Notation) is just a data format used to store and transfer data, while JavaScript is a programming language used to build websites and applications. JSON is like a simple text file containing structured data, whereas JavaScript can run logic, interact with users, and manipulate web pages. JSON is often used in JavaScript, but it can also be used in Python, Java, and other languages.

#### **Outcomes:**

CO3: Apply JavaScript and JSON for web application development

**Conclusion:** (Conclusion to be based on the outcomes achieved)

From this experiment, I learned how to work with JSON in JavaScript, including converting objects to strings, modifying data, and parsing JSON. I also understood why JSON is preferred over XML and how it differs from JavaScript. JSON is an essential tool for handling data in web development, making it easy to share and manipulate information efficiently.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

## **References:**

## **Books/ Journals/ Websites:**

- "Web technologies: Black Book", Dreamtech Publications
- <a href="http://www.w3schools.com">http://www.w3schools.com</a>

