Google | 100 طويق

JavaScript VI: JSON

أكاديمية طويق
TUWAIQ ACADEMY

وزارة الاتصالات وتقنية المعلومات
MINISTRY OF COMMUNICATIONS AND INFORMATION TECHNOLOGY

الاتحاد السعودي للأمن السيبراني والبرمجة والدرونز
SAUDI FEDERATION FOR CYBERSECURITY, PROGRAMMING & DRONES

مهارات المستقبل

stc

Objective

- JSON & XML
- JSON syntax, objects and methods
- Session Storage
- The Web Storage API

Sending Objects Between a Web Server and Clients

❖ HTTP allows information to be transferred across the internet. You need to keep in mind these 2 important facts about HTTP:

1- HTTP is a stateless protocol—it doesn't see any link between two requests being successively carried out on the same connection. Cookies and the Web Storage API are used to store necessary state information.

2- HTTP transfers text (not objects or other complex data structure). JSON converts data structures, like objects, into text that can be transferred using HTTP.

JavaScript Object Notation (JSON)

XML and JSON are commonly used to convert JavaScript objects into a format that can be transferred with HTTP. This is because, as mentioned previously, HTTP can only transfer text across the web. This me answe can not transfer JavaScript objects between a webserver and a client.

What is XML?

- eXtensible Markup Language (XML) is used to annotate text or add additional information.
- Tags are used to annotate data.
- These tags are not shown to the end-user, but are needed by the ‘machine’ to read and subsequently process the text correctly.

```
<book id="bk101">
<author>Gambardella, Matthew</author>
<title>XML Developer's Guide</title>
<genre>Computer</genre>
<price>44.95</price>
<publish_date>2000-10-01</publish_date>
<description>An in-depth look at creating applications with
 XML.</description>
</book>
```

Note that the tags are one the left and right of the data you want to markup

What is JSON?

- JSON, or JavaScript Object Notation, is a syntax for converting objects, arrays, numbers, strings, Booleans into a format that can be transferred between the web server and the client. Like XML, JSON is language in dependent.

- JSON Is text and any JavaScript object can be converted into JSON, which can then be sent to the server. Any JSON data received from the server can also be converted into JavaScript objects.

- JSON is quicker to read and write than XML. It also doesn’t use end tags, can use arrays and can be parsed by a standard JavaScript function.

JSON Syntax

- Data are in key/value pairs.
    - The key must be a string, enclosed in double quotes, while the value can be a string, a number, a JSON object, an array, a Boolean or null.
- Property names must be double-quoted strings.
- Property names must be double-quoted strings
- Data are separated by commas
- Objects are held by curly braces - { }
- Arrays are held by square brackets - [ ]
- The file type for JSON files is ".json" and the MIME type for JSON text is "application/json".

JSON Objects

In the example below, note that the key is a string in double quotes and the value can be a string, number, JSON object, array, boolean or null.

```
let myObj = { "name":"Jason", "age":30, "car":null };
let x = myObj.name;
```

The object value is accessed using the dot (.) notation. It could also have been accessed using the square brackets ([]) notation:

```
let myObj = { "name":"Jason", "age":30, "car":null };
let x = myObj["name"];
```

JSON Objects

The dot/bracket notation can be used to modify any value in a JSON object:

```
myObj.age = 31;          //myObj already declared beforehand
myObj[“age”] = 31;
```

The delete keyword can be used to delete properties from the object:

```
delete myObj.age;      //myObj already declared beforehand
```

JSON Objects

You can also save an array of objects:

```
arrayofPersonObjects = [{ "name": { "first": "Tom", "last": "Smith" },
"age": "21", "gender": "male", "interests": "Programming" },
    { "name": { "first": "Jack", "last": "Daniels" }, "age": "19", "gender":
"male", "interests": "Gaming" }
];
```

Or use the for-in loop to loop through an object’s properties

```
myObj = { "name":"Jason", "age":30, "car":null };
for (x in myObj) {
    document.getElementById("demo").innerHTML += x;
}
```

JSON Objects

Use the bracket notation to access property values in a for-in loop:

```
myObj = { "name":"Jason", "age":30, "car":null };
for (x in myObj) {
  document.getElementById("demo").innerHTML += myObj[x];
}
```

JSON Objects

Example of a JSON object within a JSON object:

```
myObj = {
    "name":"Jason",
    "age":30,
    "cars": {
        "car1":"Ford",
        "car2":"BMW",
        "car3":"VW"
    }
 }
```

Use the dot/bracket notation to access nested JSON objects:

```
x = myObj.cars.car2;
//or
x = myObj.cars["car2"];
```

JSON Methods

JSON.parse()

You receive this text from a web server:

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

By parsing, the data becomes a JavaScript object:

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

and you can use the object in your page:

```
<p id="demo"></p>

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

JSON Methods

JSON.stringify()

You receive this text from a web server:

```
let obj = { "name":"Jason", "age":30, "city":"New York"};
```

Using JSON.stringify() converts the object to a string:

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

And that can be sent to a server:

```
let obj = { "name":"Jason", "age":30, "city":"New York"};
let myJSON = JSON.stringify(obj);
document.getElementById("demo").innerHTML = myJSON;
```

The Web Storage API

We can store information using:

1. *sessionStorage* — stores state information for each given origin for as long as the browser is open.
2. *localStorage* — stores state information for each given origin even when the browser is closed and reopened.

sessionStorage

To add the key value pair {“totalPersonObjs”,1} to sessionStorage:

```
sessionStorage.setItem("totalPersonObjs", 1);
```

Retrieve a value from sessionStorage:

```
let total = parseInt(sessionStorage.getItem("totalPersonObjs"));
```

Summary

- JSON & XML
- JSON syntax, objects and methods
- sessionStorage
- The Web Storage API

Resources

- https://www.any-api.com/
- https://rapidapi.com/marketplace
- https://github.com/public-apis/public-apis