BÀI HỌC: JSON

1. Tổng quan

JSON (JavaScript Object Notation) là một tiêu chuẩn mở dựa trên văn bản nhẹ được thiết kế cho trao đổi dữ liệu có thể đọc được của con người. Các quy ước được sử dụng bởi JSON được các lập trình viên biết đến, bao gồm C, C ++, Java, Python, Perl, v.v.

- ✓ JSON là viết tắt của JavaScript Object Notation.
- ✓ Định dạng do Douglas Crockford chỉ định.
- ✓ Nó được thiết kế để trao đổi dữ liệu có thể đọc được của con người.
- ✓ Nó đã được mở rộng từ ngôn ngữ kịch bản JavaScript.
- ✓ Phần mở rộng của tên tệp là .json.
- ✓ Kiểu JSON Internet Media là application / json.
- ✓ Định danh Loại thống nhất là public.json.

2. Sử dụng JSON

- ✓ JSON được sử dụng trong khi viết các ứng dụng dựa trên JavaScript bao gồm các tiện ích mở rộng của trình duyệt và các trang web.
- ✓ Định dạng JSON được sử dụng để tuần tự hóa và truyền dữ liệu có cấu trúc kết nối mạng.
- ✓ JSON chủ yếu được sử dụng để truyền dữ liệu giữa máy chủ và các ứng dụng web.
- ✓ Các dịch vụ web và API sử dụng định dạng JSON để cung cấp dữ liệu công khai.
- ✓ JSON có thể được sử dụng với các ngôn ngữ lập trình hiện đại.

3. Đặc điểm của JSON

- ✓ JSON dễ dàng đọc và viết
- ✓ JSON là một định dạng trao đổi dựa trên text plain.
- ✓ JSON độc lập với ngôn ngữ.

Một ví dụ JSON

Ví dụ sau cho thấy cách sử dụng JSON để lưu trữ thông tin liên quan đến cuốn sách (book):

```
{
"book": [
  {
    "id":"01",
    "language": "Java",
    "edition": "third",
    "author": "Herbert Schildt"
    },
    {
    "id":"07",
    "language": "C++",
    "edition": "second"
    "author": "E.Balagurusamy"
  }]
}
```

File html có nội dung sau:

```
<html>
<head>
     <title>JSON example</title>

     <script language="javascript">

          var object1 = { "language": "Java", "author": "herbert schildt" };
          document.write("<h1>JSON with JavaScript example</h1>");
          document.write("<br />Object 1:");
          document.write("<h3>Language = " + object1.language + "</h3>");
          document.write("<h3>Author = " + object1.author + "</h3>");

          var object2 = { "language": "C++", "author": "E-Balagurusamy" };
          document.write("<br />Object 2:");
          document.write("<h3>Language = " + object2.language + "</h3>");
          document.write("<h3>Author = " + object2.author + "</h3>");

          document.write("<hr />");
          document.write(object2.language + " programming language can be studied " +
                "from book written by " + object2.author);
          document.write("<hr />");

     </script>

</head>

<body>
</body>

</html>
```

Kết quả chạy chạy trên trình duyệt:

JSON with JavaScript example

Object 1:

Language = Java

Author = herbert schildt

Object 2:

Language = C++

Author = E-Balagurusamy

C++ programming language can be studied from book written by E-Balagurusamy

4. Cú pháp JSON

Cú pháp JSON về cơ bản được coi là tập hợp con của cú pháp JavaScript; nó bao gồm những điều sau:

- ✓ Dữ liệu được biểu diễn dưới dạng khóa / giá trị.
- ✓ Dấu ngoặc nhọn chứa các đối tượng và mỗi tên được theo sau bởi ':' (dấu hai chấm), tên / giá trị
- ✓ Các cặp được phân tách bằng dấu, (dấu phẩy).
- ✓ Dấu ngoặc vuông chứa các mảng và các giá trị được phân tách bằng dấu, (dấu phẩy).

Dưới đây là một ví dụ đơn giản:

```
{
"book": [
  {
    "id":"01",
    "language": "Java",
    "edition": "third",
    "author": "Herbert Schildt"
    },
    {
    "id":"07",
    "language": "C++",
    "edition": "second"
    "author": "E.Balagurusamy"
  }]
}
```

JSON hỗ trợ hai cấu trúc dữ liệu sau:

- ✓ Tập hợp các cặp tên / giá trị: Cấu trúc dữ liệu này được hỗ trợ bởi các ngôn ngữ lập trình.
- ✓ Danh sách có thứ tự các giá trị: Nó bao gồm mảng, danh sách, vectơ hoặc chuỗi, v.v.

5. Kiểu dữ liệu của JSON

Kiểu	Mô tả
Number	Kiểu dữ liệu số
String	Kiểu dữ liệu xâu ký tự
Boolean	Kiểu dữ liệu có 2 giá trị true và false
Array	Kiểu mảng
Value	Giá trị
Object	Kiểu đối tượng
null	Trống

6. So sánh JSON và XML

- Cả JSON và XML đều có thể được sử dụng để nhận dữ liệu từ máy chủ web.
- Các ví dụ JSON và XML sau đây đều xác định một đối tượng nhân viên, với một mảng gồm 3 nhân viên:

Ví dụ JSON

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

```
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

Ví dụ XML

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

JSON giống như XML bởi vì

- Cả JSON và XML đều "tự mô tả" (con người có thể đọc được)
- Cả JSON và XML đều phân cấp (giá trị trong giá trị)
- Cả JSON và XML đều có thể được phân tích cú pháp và được sử dụng bởi rất nhiều ngôn ngữ lập trình
- Cả JSON và XML đều có thể được tìm nạp (fetch) bằng XMLHttpRequest

JSON không như XML vì

- JSON không sử dụng thẻ kết thúc
- JSON ngắn hơnJSON đọc và viết nhanh hơn
- JSON có thể sử dụng mảng
- Sự khác biệt lớn nhất là:XML phải được phân tích cú pháp bằng trình phân tích cú pháp XML.
- JSON có thể được phân tích cú pháp bởi một hàm JavaScript tiêu chuẩn.

Tại sao JSON tốt hơn XML

- XML khó phân tích cú pháp hơn nhiều so với JSON.
- JSON được phân tích cú pháp thành một đối tượng JavaScript sẵn sàng sử dụng.
- Đối với các ứng dụng AJAX, JSON nhanh hơn và dễ dàng hơn so với XML:
- Sử dụng XML
    - Tìm nạp tài liệu XML
    - Sử dụng DOM XML để lặp qua tài liệu
    - Trích xuất các giá trị và lưu trữ trong các biến
- Sử dụng JSON
    - Tìm nạp một chuỗi JSON
    - JSON.Parse chuỗi JSON

7. Các hàm JSON

Hàm JSON.parse

Sử dụng hàm JavaScript JSON.parse () để chuyển đổi một chuỗi thành một đối tượng JavaScript:

Giả sử có chuỗi: '{ "name":"John", "age":30, "city":"New York"}'

Để chuyển chuỗi này thành đối tượng, chúng ta sử dụng:

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

JSON.stringify()

Sử dụng hàm JavaScript JSON.stringify () để chuyển đối tượng JavaScript thành một chuỗi.

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

var myJSON = JSON.stringify(obj);

8. Một số hạn chế của JSON

- Không có lược đồ. Một mặt, điều đó có nghĩa là chúng ta hoàn toàn có thể linh hoạt để trình bày dữ liệu theo bất kỳ cách nào chúng ta muốn. Mặt khác, điều đó có nghĩa là chúng ta có thể vô tình tạo ra dữ liệu dạng sai rất dễ dàng.
- Chỉ có một loại số: định dạng dấu phẩy động chính xác kép IEEE-754. Điều đó khá thú vị, nhưng nó chỉ đơn giản có nghĩa là bạn không thể tận dụng các loại số đa dạng và sắc thái có sẵn trong nhiều ngôn ngữ lập trình.
- Không có loại ngày. Thiếu sót này có nghĩa là các nhà phát triển phải sử dụng chuỗi biểu thị ngày, dẫn đến sự khác biệt về định dạng hoặc phải biểu diễn ngày ở dạng mili giây kể từ kỷ nguyên (ngày 1 tháng 1 năm 1970).
- Không có chú thích. Không thể chú thích dẫn đến có thể có sự hiểu nhầm trong đoạn mã.

Tham khảo:

https://www.tutorialspoint.com/json/json_tutorial.pdf
https://www.tutorialspoint.com/json/pdf/json_quick_guide.pdf
https://www.infoworld.com/article/3222851/what-is-json-a-better-format-for-data-exchange.html
https://www.w3schools.com/js/js_json_intro.asp