JSON-based Apparel Fabric Query Micro-program Application

Liu Xueying\textsuperscript{1,a}, Zhao Yejun\textsuperscript{1}

\textsuperscript{1}Information Center, Beijing Institute of Fashion Technology, Beijing, China
\textsuperscript{a} Corresponding author: 18810551653@163.com

Abstract. With the continuous development of information technology, the application market has also gradually innovated. Lightweight applications are favored by people because of its small footprint, easy to use, and fast running. JSON (JavaScript Object Notation) is considered by developers to be an ideal data exchange language because of its simplicity and ease of use, and is widely used in Web data exchange processing. JSON is used to cache data, which is convenient for storing complex data, and is more conducive to client parsing and use.

1. Introduction to JSON

The Internet Corporation (W3C) released the XML (Extensible Markup Language) extensibility markup language in 1998. It describes the various data in a structured way. It is not limited to specific networks, platforms and programming languages. XML is a subset of SGML (Standard Generalized Markup Language), similar to HTML.

XML syntax rules are simple, but they are strictly required and belong to the "tree" structure. The root element is very important and is the parent of all elements, and all elements must be properly nested and the tags must appear in pairs. Therefore, the XML document occupies a large space, the format is cumbersome, the data is highly repetitive, the parsing consumes a lot of system resources, and the execution efficiency is low.

The JSON (JavaScript Object Notation) syntax is a language that is very similar to the C family syntax, and is used to transfer and generate data. Compared to XML, JSON is an easier and more user-friendly web service client format that is easier to parse in C-series languages. It can quickly convert a set of data from a client page in WEB development to a string and pass the string to the back-end. And because the JSON format is written in plain text, it's easier to read and makes the writer look simpler.

1.1 JSON syntax format

The JSON syntax is a subset of the JavaScript object syntax. The JSON syntax has four basic data types and two structures. The basic data types are String, Number, Boolean, and Null. The two structure types of JSON are: object structure and array structure. The object structure is a collection of name/value pairs, each object is contained in "{}", the object name and value are separated by ":" (colon), the name and value are caused by "" (double quotation marks), name/value Use "," (comma) to separate.

E.g:
```
{"key1": value1,
 "key2": value2, …}
```
An arrangement is an ordered list of values, an ordered collection of values. The array structure is enclosed in "["]”, separated by ""," (comma) 
E.g: 
[{"key1" : “value11, value12·…"},
 {“key2" : “value21, value22·…"},
 ...]

1.2 The advantages of JSON
Simplifying the data access process is a distinct advantage of json data transfer compared to XML. The symbols used in the JSON data format (such as arrays, strings, objects, etc.) can be run in the JavaScript engine without parsing. This makes the program more and more convenient to access data than DOM technology.

JSON not only makes parsing more convenient, but also has better compatibility during development. It can get data by accessing object properties and traversing arrays. At the same time, JSON has better readability than XML and has the basic characteristics of structured data. In actual use, such as Google Maps, use a JSON solution instead of XML. Another advantage of JSON is its simplicity. Unlike XML, in XML, tags must appear in pairs to satisfy tag compliance. In JSON, all of these requirements can be met with simple parentheses. On the content side, you can dynamically change the JSON data as needed. This gives the user a lot of flexibility. In the data exchange with more fields, traditional XML takes a long time in this process. In contrast, JSON processing is faster.

1.3 JSON data structure usage example in clothing label query system
E.g:
var fabric ={
 "cotton": {
 "Source": "Natural plant fiber",
 "Benefits": "easy to keep warm, soft, close-fitting, moisture-absorbing, breathable",
 "Disadvantages": "easy to shrink, easy to wrinkle, often ironed. Alkali-resistant, not acid-resistant. If it is acid, it will corrode and cause loopholes."
 }
 "Mulberry silk": {
 "Source": "With mulberry silk as raw material, the production process is not polluted, so it is a highly respected green product in the world. It is called “Fiber Queen”.",
 "Advantages": "Good air permeability, good hygroscopicity and good drape. Soft, bright and comfortable to wear. Fabric texture is elegant and luxurious."
 "Disadvantages": "Easy to hook. Poor resistance to salt, salt in sweat will corrode it. Poor light resistance."
 }
}

2. Data access interface architecture
The architecture of the data access interface in this article is as follows:
Figure 1

As shown in Figure 1, the data access interface in the system completes the middle layer design connecting the client layer and the server side data layer. When the client issues a query request on the browser, the query parameters are passed as a get request to the data access interface, and then the process of querying the parameters. This layering can be extended to ease the pressure of data queries during the processing of data. In the data access process, the query parameters are passed to the data layer through the stored procedure, and the query result is returned to the stored procedure call interface. In order to make the data results returned to the client more simplified, the data access interface will post-process after accepting the returned results. The system encapsulates the result data into the data object through the object data, encapsulates the object into a JSON string through the JSON wrapper module, and finally returns the JSON string to the client. The client performs front-end parsing after receiving the data, and finally displays the client layer on the mobile port or web port in a specific style.

3. JSON data-based query method

3.1 Preprocessing of JSON data

Take the data file in the query system as an example:

```javascript
var fabric = {
    "cotton": {
        "Source": "Natural plant fiber",
        "Benefits": "easy to keep warm, soft, close-fitting, moisture-absorbing, breathable",
        "Disadvantages": "easy to shrink, easy to wrinkle, often ironed. Alkali-resistant, not acid-resistant. If it is acid, it will corrode and cause loopholes."
    },
    "Mulberry silk": {
        "Source": "With mulberry silk as raw material, the production process is not polluted, so it is a highly respected green product in the world. It is called "Fiber Queen".",
        "Advantages": "Good air permeability, good hygroscopicity and good drape. Soft, bright and comfortable to wear. Fabric texture is elegant and luxurious."
        "Disadvantages": "Easy to hook. Poor resistance to salt, salt in sweat will corrode it. Poor light resistance."
    }
}
```

var safety = {
"A": "formaldehyde content (mg / kg) ≤ 20, PH value: 1.0 ~ 7.5, color fastness: washable ≥ 3-4 grade, acid-resistant perspiration ≥ 3-4 grade, alkali-resistant sweat stain ≥ 3 grade 4 Grade, dry friction ≥ 3-4, saliva ≥ 4, no odor, banned decomposable aromatic dyes banned.
"B": "Formaldehyde content (mg / kg) ≤ 75, PH value: 4.0 ~ 8.5, color fastness: washable ≥ 3, acid and sweat resistance ≥ 3, alkali resistance ≥ 3, dry friction ≥ 3, no odor, disable the decomposable aromatic dyes disabled. Belong to direct contact with skin textiles,
"C": "Formaldehyde content (mg / kg) ≤ 300, PH value: 4.0 ~ ~ 9, color fastness: washable ≥ 3, acid resistance, sweat resistance ≥ 3, alkali-resistant perspiration ≥ 3, Resistance to friction ≥ 3, no odor, disable decomposable aromatic amine dyes. Belong to non-contact skin textiles"
;
Since the storage form of JSON data is called JSONtree, the above example is expressed in the form of a tree as follows:

![Figure 2](image)

When the client layer issues a query request, it queries the JSONTREE in the database. JSON data query requests are usually divided into path queries and keyword queries.

The path query is an exact query, which is filtered layer by layer in JSONTREE according to the established path. The path is found by the “key--value--key” of the upper and lower layers, and each key is separated by ".". Similar to finding a file on a disk by a computer user, the result is only one or not found.

Perform a keyword query in JSONTREE based on the given key. In some cases, different levels of variables may have the same key. Therefore, there will be multiple qualified query results, which are required to meet the user's needs through the screening results.

3.2 Query system example
The client code is as follows:

```html
<html>
<link href="style.css" type="text/css" rel="stylesheet">
<head>
<title>Clothing information inquiry</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="data_src.js"></script>
```
The middle layer code is as follows:

```javascript
function get1()
{
  // Code to handle material input and features
}
```

```javascript
function get2()
{
  // Code to handle size input and check
}
```

```javascript
function get3()
{
  // Code to handle standard and interpretation
}
```

```javascript
function get4()
{
  // Code to handle safety category and description
}
```
The system uses the path query method to obtain the value of the character filled in the input box of the client through `document.getElementById("id")`, then traverse the JSON tree in the JSON data file, and get the array from the query path, according to the given Key to query, return the query results.

4. Conclusion

Because of its simple writing, strong readability and low resolution, the JSON data format has been widely used by developers and has been developed with many tools and usage methods. In the actual application process, we also need to pay attention to the details of browser adaptability and Chinese character processing. When using the JSON data format, you must write strings in strict accordance with the JSON format requirements. Property names must use double quotes. The array form must be written exactly as specified.

Compared with relational databases and XML markup languages, JSON objects are more scalable and independent based on file format, and other systems can easily query data from JSON objects. In relational databases, data modeling is limited by the number of permissions on the management system. When the amount of data is large, the retrieval process becomes very slow. In JSON, data can be added at any time. Even if the content changes, there is no need to redesign. JSON is more practical and efficient for extracting complex data. In the future work, JSON objects will be used for data storage and research.

References

[1] Gong Jianhua. Application of JSON format data in web development [J]. Office Automation, 2013(20):46-48.

[2] Hu Qianli. Comparing the application of XML and JSON in the Web [J]. SME Management and Technology (late issue), 2011(7):268-270.

[3] Qiu Xiaohua, Qin Shuanshuan, Qiu Guo. Research on XML and JSON Data Transmission Format Based on WEB Development[J]. Information Technology and Informatization, 2017(4):123-125.

[4] Qu Zhan, Li Wei. Application of JSON in Ajax Data Exchange[J]. Journal of Xi’an Shiyou University: Natural Science, 2011, 26(1): 95-98.

[5] Tan Xiangwei. Using JSON to Realize Data Interaction between Android Client and Web Server[J]. Journal of Fujian Computer, 2013, 29(2): 166-167.

[6] Zhang Yifan, Liu Pingzeng, Ma Hongjian, et al. A Distributed System Architecture Based on JSON[J]. Chinese Journal of Agricultural Mechanization, 2015, 36(5): 255-257.

[7] Yang Shulin, Hu Jieping. JSON Data Exchange Format and Its Application in Data Verification[J]. Journal of Beijing Institute of Graphic Communication, 2008, 16(4): 56-58.

[8] Hu Xiaofeng. Application Analysis of JSON and XML in Network Data Transmission[J]. Computer Programming Techniques & Maintenance, 2010(10): 77-78.

[9] Ni Ruixi. A Heterogeneous Data Query Method Based on JSON[J]. Radiocommunication Technology, 2013, 39(1): 73-76.

[10] Gu Fangzhou, Shen Bo. Application Research of JSON Data Exchange Format in Heterogeneous System Integration[J]. Railway computer application. 2012, 21(2):1-4.
[11] Xu Baolei, Luo Jiang, Pan Gang. Application Research of Data Exchange Technology Based on JSON[J]. Software Guide, 2017, 16(10): 173-175.

[12] Zhang Huzhen, Qu Gansong, Hu Ruiqi. Data exchange model based on JSON[J]. Computer Engineering and Design, 2015(12):3380-3384.

[13] Li Jinlai, Wu Tao. Data Interaction between Mobile Terminal and Server-side Heterogeneous Database Based on JSON Data Exchange[J]. Modern computer, 2016(16):55-58.

[14] Wang Kuisheng, Wang Xiaobo. Data Interaction between Website Client and Server Using JSON[J]. Software Guide, 2010, 09(3): 147-149.

[15] Gao Jing, Duan Huichuan. Research on JSON data transmission efficiency[J]. Computer Engineering and Design, 2011, 32(7): 2267-2270.