

On how to convert XML Data to JSON Format
XMLデータをJSON形式に変換する方法について

Feb. 22, 2024

Tourism Technology Study Group in JTREC, Lead
Kazuyoshi Itagaki

1. はじめに

- 鈴木理事の提示された”Draft Development of Classes related to Sustainability Claims”にある ebXML by RDM→API by JSON-LD Transformation（次頁参照）は、当プロジェクトの主要テーマの一つである。
- しかるに、Transformationは深い技術的な内容を持っており、容易に理解・習得できるものでないことが分かった。
- そこで、まずJSON形式の基本事項から始めて、JavaScriptオブジェクト → JSON文字列への変換、XML から JSONへの変換と探ってみたが、実用的な域にはまだ至っていない。
- ここでは、簡単にその内容を紹介するにと止め、諸賢のご教示を請う次第である。

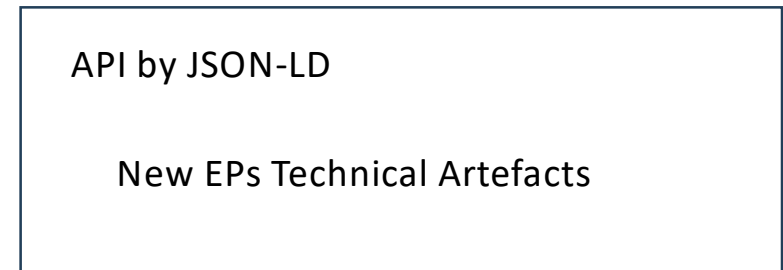
4. Project Scope

Development Steps of API Transformation of EPs Technical Artefacts



XML Naming and Design Rules and RDM Rules

Transformation (not yet developed)



JSON and Open API

Naming and Design Rules

2. JSON の操作

JSON の操作

- JavaScript Object Notation (JSON) は、構造化データを表現するための標準のテキストベースの形式で、JavaScript のオブジェクト構文に基づいています。ウェブアプリケーションでデータを転送する場合によく使われます（例えば、複数のデータをサーバーからクライアントへ送信して、ウェブページ上に表示する場合などで、その逆もあります）。
- 文字列をネイティブオブジェクトへ変換することは「デシリアライズ」(deserialization) と呼ばれており、ネイティブオブジェクトをネットワークを通して転送できるように文字列へ変換することは「シリアライズ」(serialization) と呼ばれています。

JSON の構造

- JSON は JavaScript オブジェクトにとっても似ている形式の文字列です。JSON では通常の JavaScript オブジェクトと同様な基本データ型（文字列、数値、配列、論理型やその他のリテラル型）を使うことができます。これにより、階層的にデータを構成することができます。

3. JSON文字列 → オブジェクトへの変換

- json_strがJSON文字列です。これをJSON.parse()に渡すことでオブジェクトに変換されます。あとは通常のオブジェクトの扱いと同じで、ドットで繋げながら取得したい要素にアクセスしています。
次にオブジェクト→JSON文字列への変換を見ていきましょう。

- JavaScriptコード

```
var json_str = '{"id":1, "name":"tanaka", "attribute":{"gender":"male",  
"phone_number":"xxxxxxxxxxx", "birth":"1991/01/01"}}';  
var obj = JSON.parse(json_str)  
console.log(obj.name)  
console.log(obj.attribute.birth)
```

- 結果

```
tanaka  
1991/01/01
```

4. JavaScriptオブジェクト → JSON文字列への変換

- objはJavaScriptのオブジェクトです。これをJSON.stringify()に渡してJSON文字列に変換しています。typeofで型を見てみると、stringになっているのが分かります。
- 普段JavaScript内で使用するときにはオブジェクトが良いですが、サーバサイドとデータのやり取りをしたい場合は文字列にして送ると非常に便利です。
- JavaScriptコード

```
var json_obj = '{"id":1, "name":"tanaka", "attribute":{"gender":"male",  
"phone_number":"xxxxxxxxxxx", "birth":"1991/01/01"}}';  
var json_str = JSON.parse(json_obj)  
console.log(json_str)  
console.log(type of json_str)
```

- 結果

```
{"id":1,"name":"tanaka","attribute":{"gender":"male","phone_number":"xxxxxxxxxxx","birth":"1991/01/01"}}  
string
```

- 出典：<https://products.sint.co.jp/topsic/blog/json>

5. XML to JSON converter (Online Converter)

- Free tool to convert XML data to equivalent JSON format. XML elements are converted to JSON keys, and element values are converted to corresponding JSON values.
 - XMLデータを同等のJSON形式に変換する無料のonlineツール。XML要素はJSONキーに変換され、要素値は対応するJSON値に変換されます。
 - XML attributes are converted to respective JSON keys with the prefix "-". In such cases, the XML element values are converted to their respective JSON data using the "#text" JSON key.
 - XML属性は接頭辞「-」を持つそれぞれのJSONキーに変換されます。そのような場合、XML要素値は「#text」のJSONキーを用いてそれぞれのJSONデータに変換されます。
-
- 出典：<https://www.site24x7.com/ja/tools/xml-to-json.html>

For example

XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<bookstore>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price> </book>
  <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price> </book>
</bookstore>
```

JSON

```
{
  "bookstore": {
    "book": [
      {
        "-category": "children",
        "title": {
          "-lang": "en",
          "#text": "Harry
Potter"
        },
        "author": "J K.
Rowling",
        "year": "2005",
        "price": "29.99"
      },
      {
        "-category": "web",
        "title": {
          "-lang": "en",
          "#text": "XQuery
Kick Start"
        },
        "author": [
          "James McGovern",
          "Per Bothner",
          "Kurt Cagle",
          "James Linn",
          "Vaidyanathan
Nagarajan"
        ],
        "year": "2003",
        "price": "49.99"
      }
    ]
  }
}
```


6. Use a JavaScript library called "xml-js" 「xml-js」というJavaScriptのライブラリを使用

- Installing xml-js

```
npm install xml-js
```

- Get data from API
- APIからデータを取得

```
const response = await fetch(`${domain}/db.php?Command=TitleLookup&TID=${tid}`);
```

- Convert XML to text
- XMLをテキストに変換

```
const dataText = await response.text();
```

- Using the xml-js library with the code below to output dataJsonText in console.log.
- 以下のようなコードでxml-jsライブラリを使用してdataJsonTextをconsole.logで出力

```
const convert = require('xml-js');  
const dataJsonText = convert.xml2json(dataText, {compact: true, spaces: 4});
```

6. Store JSON object in dataJson dataJsonにJSONオブジェクトを格納

- Storing JSON objects in dataJson
- dataJsonへのJSONオブジェクトの格納

```
const dataJson = JSON.parse(dataJsonText)
```

- Extracting the part with the necessary data
- 必要なデータがある部分を取り出す

```
const dataComment = dataJson.TitleLookupResponse.TitleItems.TitleItem.Comment._text;
```

- 出典：<https://zenn.dev/peishim/articles/920aa4d2648548>