

JSON

Authored by Chris Bohnet

Agenda

1. What is JSON?
2. What does JSON look like?
3. JSON Request Headers
4. JSON Responses
5. How is JSON different from SOAP APIs?
6. JSON API End-to-end data flow

What is JSON?

*JavaScript Object
Notation*

It is a format used to transfer data between clients and servers.

It's a way to store and persist the data that's communicated.

It's commonly used to read data from a web server and display data on a web page with HTTP RESTful APIs.

It is text-based.

It is written in key-value pairs.

It is both human-readable and machine-understandable.

JSON is considered a subset of the JavaScript language but any language can use it.

What does JSON look like?

Request Body

The JSON request body is a collection of key-value pairs defining the data being exchanged.

It has a specific set of syntax rules:

Double quotes describe strings . `{"name": "Chris"}`

Numbers are displayed without additional syntax . `{"age": 25}`

Boolean (true / false) values can also be defined. `{"hasChildren": true}`

Curly brackets are used for a collection of key-value pairs (objects).

Employee: `{"name": "Chris",
 "age": 25,
 "status": "single",
 "city": "Omaha"}`

Square brackets define lists or arrays. `{"numbers": [1,2,3]}`

JSON Request Example

```
{  
  "menu": {  
    "id": "01",  
    "category": "dinner",  
    "menu": {  
      "menuitems": [  
        { "value": "steak", "price": 12.00 },  
        { "value": "chicken", "price": 10.00 },  
        { "value": "fish", "price": 15.00 }  
      ]  
    }  
  }  
}
```

JSON Request Headers

Request Header

In an HTTP request, there is a header section that is used to indicate a number of operating parameters.

To indicate the data format usage of JSON, the Content-Type parameter must be set to application/json.

In PHP,

```
header('Content-Type: application/json');
```

In Express,

```
app.get('/hello',function(req, res) {  
  res.type('json');  
  res.send('{ "name":"Hello"}');  
})
```

In JAVA,

```
public ResponseEntity<String> bar() {  
  final HttpHeaders httpHeaders= new HttpHeaders();  
  httpHeaders.setContentType(MediaType.APPLICATION_JSON);  
  return new ResponseEntity<String>("{\"test\":  
    \"jsonResponseExample\"}", httpHeaders, HttpStatus.OK);  
}
```

JSON Responses

In general ...

- JSON itself offers many patterns to structuring JSON responses.
- However, there is no standard with naming the responses or the types of responses.
- Some models from complex to simple have been created in attempt to define rules to follow such as JSend, JSON API, OData JSON Protocol or HAL.
- Amongst these, JSend provides the most simplistic adaptation.

Introducing JSend ...

- Focuses on application-level messaging.
- Has 4 main levels: status, message, code and data.
- Defines 3 types of responses: success, failure or error.

```
{
  status : "success",
  data : {"post" : {"id":2, "title":"Book", "body":"About it" }}
}
```

```
{
  status : "fail",
  data : {"title":"A title is required"}
}
```

```
{
  status : "error",
  message : "Unable to connect to database"
}
```

How is JSON different than SOAP APIs?

JSON API

Support for all browsers.

Easy to read and write.

Straightforward syntax.

JavaScript eval() function parsing.

Easy to create and manipulate.

Supported by most backend technologies

Can transmit and serialize data using a network connection.

Can be used with modern languages.

Uses text that can be converted to JavaScript object to JSON and sends it to the server.

SOAP API

Documents are transportable across systems using XML.

XML separates the data from HTML.

XML data is typeless.

XML has namespace support.

Offers ways to display data.

Is more secure compared to JSON.

Supports various encoding formats.

Simplifies platform change process.

Less efficient from the needed use of tag names.

JSON Request vs SOAP/XML Request

SOAP/XML's use of tags make for a more cumbersome message format relative to JSON.

JSON API Payload

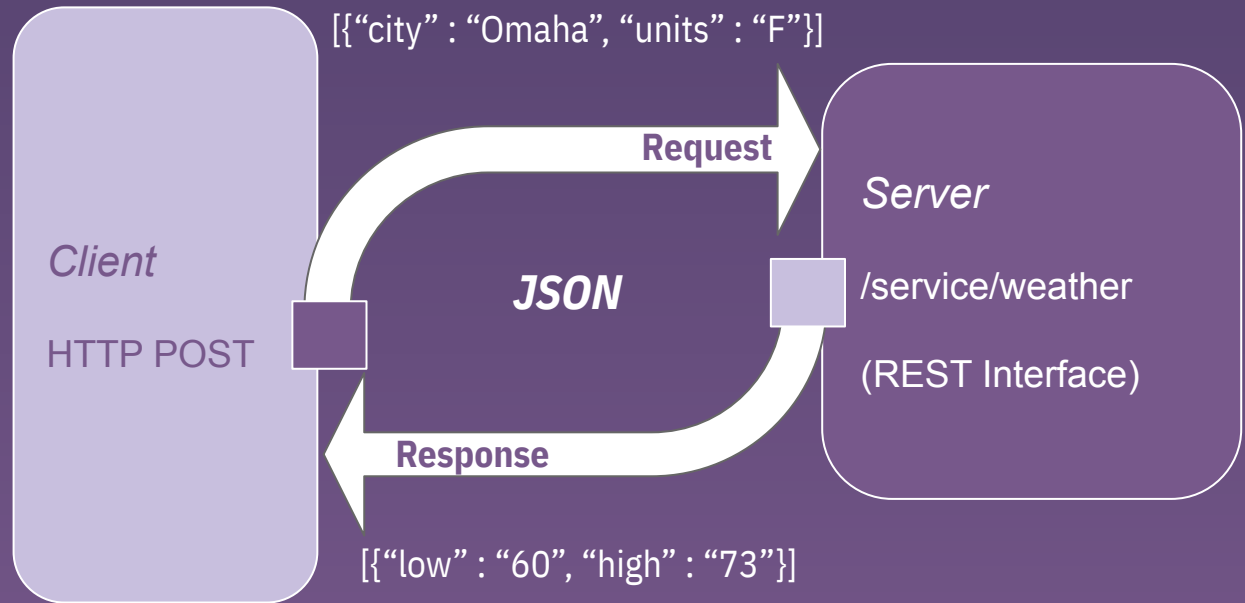
```
{
  "employees": [
    {"id": 01, "name": "Brian"},
    {"id": 02, "name": "George"},
    {"id": 03, "name": "Sam"}
  ]
}
```

SOAP API Payload

```
<employees>
  <employee>
    <id>01</id>
    <name>Brian</name>
  </employee>
  <employee>
    <id>02</id>
    <name>George</name>
  </employee>
  <employee>
    <id>03</id>
    <name>Sam</name>
  </employee>
</employees>
```

JSON API End-to-End Data Flow Diagram

JSON/REST/HTTP



JSON, “Introducing JSON” as referenced at <https://www.json.org/json-en.html>.

AU Courant Technology, “Category Archives: devops”, as referenced at <https://aucouranton.com/category/devops/>.

MSN, “Handle content-types - Azure Logic Apps, as referenced at <https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-content-type>.

Techeplanet, “Express Set Content Type, as referenced at <https://techeplanet.com/express-set-content-type/>.

Technick, “Simple API JSON Response Format, as referenced at https://technick.net/guides/software/software_json_api_format/.

Github, “JSend”, as referenced at <https://github.com/omniti-labs/jsend>.

Codeblogmoney, “JSON Example with Data Types Including JSON Array”, as referenced at <https://codeblogmoney.com/json-example-with-data-types-including-json-array/>.

Guru99, “JSON vs XML: What’s the Difference?”, as referenced at <https://www.guru99.com/json-vs-xml-difference.html#8>.

API2Cart, “JSON vs XML: API Data Interfaction”, as referenced at <https://api2cart.com/api-technology/json-vs-xml-api-data-interaction/>.

Resources