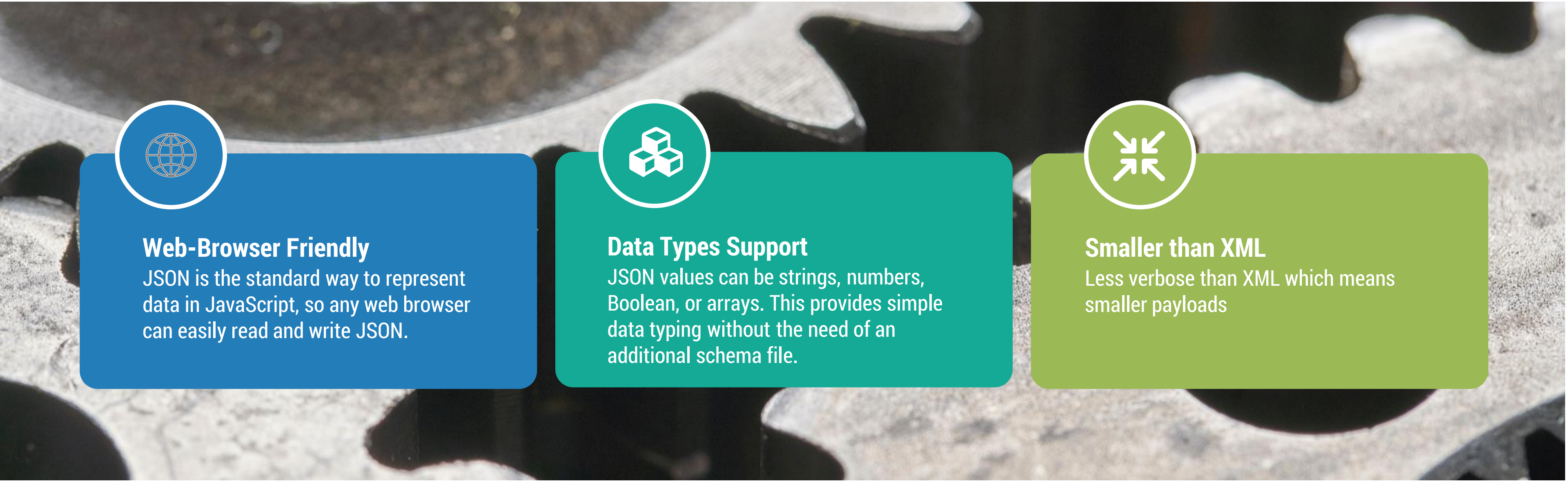
JSON

Business To Business Integration

JSON

JavaScript Object Notation



Web-Browser Friendly

JSON is the standard way to represent data in JavaScript, so any web browser can easily read and write JSON.

Data Types Support

JSON values can be strings, numbers, Boolean, or arrays. This provides simple data typing without the need of an additional schema file.

Smaller than XML

Less verbose than XML which means smaller payloads

Introducing JSON

Enclosed with curly braces

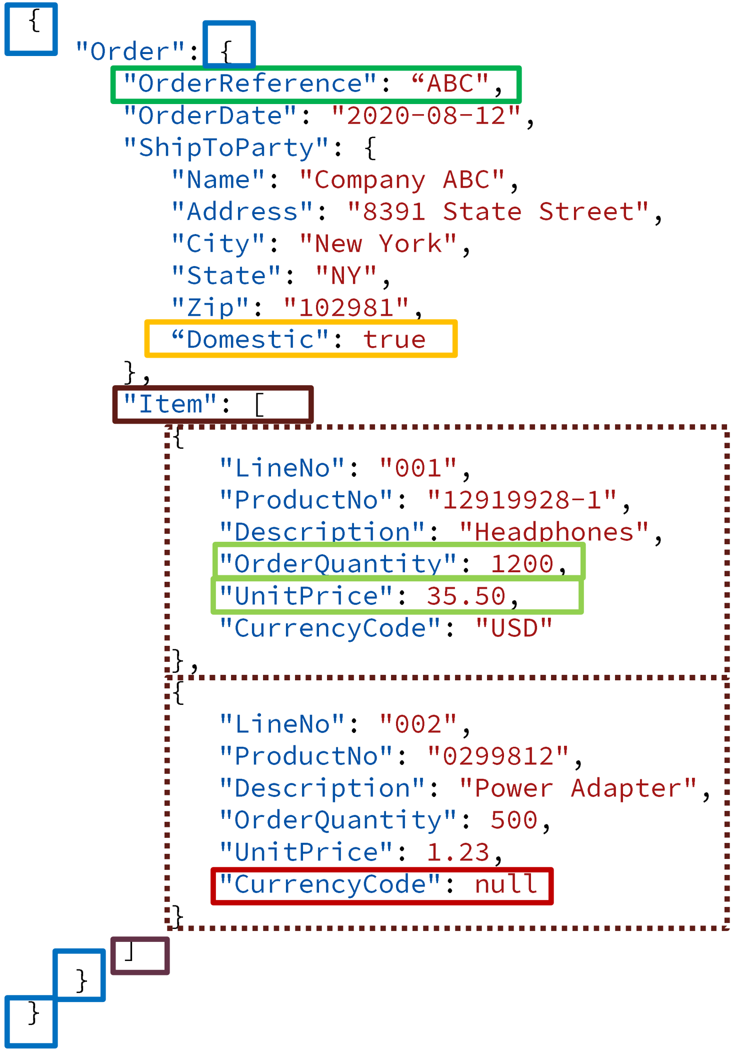
Attribute - Value Pairs

Number data type

Boolean data type (true/false)

Null data type

Array



```
{
    "Order": {
      "OrderReference": “ABC",
      "OrderDate": "2020-08-12",
      "ShipToParty": {
         "Name": "Company ABC",
         "Address": "8391 State Street",
         "City": "New York",
         "State": "NY",
         "Zip": "102981",
         “Domestic": true
      },
      "Item": [
        {
           "LineNo": "001",
           "ProductNo": "12919928-1",
           "Description": "Headphones",
           "OrderQuantity": 1200,
           "UnitPrice": 35.50,
           "CurrencyCode": "USD"
        },
        {
           "LineNo": "002",
           "ProductNo": "0299812",
           "Description": "Power Adapter",
           "OrderQuantity": 500,
           "UnitPrice": 1.23,
           "CurrencyCode": null
        }
      ]
   }
}
```

Data Types in JSON

Data Type	Description
String	A sequence of characters which can include letters, numbers, or other symbols. JSON uses Unicode which supports international characters. Strings are included by double quotation marks. Examples: “CAR” or “TEST123!” or “こんにちは”
Number	A valid number format that may optionally include decimal places. Examples: 100 or 100.23
Boolean	Boolean attributes are used to specify true or false. In JSON, the supported values for Boolean are lower case true or false.
Array	An ordered list of zero or more values which can be of any type. Arrays are surrounded by square brackets and each repeating value is separated by a comma. Example [“dog”,”cat”,”fish”]
Object	An unordered collection of name value pairs. The overall JSON document is an object, but an individual JSON attribute value can also be an object.
Null	Null means the absence of a value. To explicitly specify that an attribute has no value, lower case null is used for the value.

JSON Schema

Starts with $schema attribute

Defines a title for the object

Defines which attributes are required

Defines each property

```
"$schema": "http://json-schema.org/schema#",
  "title": "Product",
  "type": "object",
  "required": ["id", "name", "price"],
  "properties": {
    "id": {
      "type": "number",
      "description": "Product identifier"
    },
    "name": {
      "type": "string",
      "description": "Name of the product"
    },
    "price": {
      "type": "number",
      "minimum": 0
    },
    "tags": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "stock": {
      "type": "object",
      "properties": {
        "warehouse": {
          "type": "number"
        },
        "retail": {
          "type": "number"
        }
      }
    }
  }
}
```

JSON Conforming with Schema

$schema references Schema file

Property names and types conform to schema specifications

```
{
 “$schema” : “https://schemalocation.com”
  "id": 1,
  "name": "Foo",
  "price": 123,
  "tags": [
    "Bar",
    "Eek"
  ],
  "stock": {
    "warehouse": 300,
    "retail": 20
  }
}
```