

JSON

1

JSON

```
"invoice":{
  "invoice_number": "1",
  "invoice_date": "2019-02-17",
  "issued_by": "Mark",
  "bill_to": {
    "name": "Phelan",
    "address": {
      "number": "1",
      "street": "Main St",
      "city": "Dundalk"
    }
  },
  "item": [
    {
      "item_name": "Jacket",
      "price": "45.00",
      "quantity": "3"
    },
    {
      "item_name": "Jeans",
      "price": "40.00",
      "quantity": "3"
    }
  ],
  "payment_information": "No additional info"
}
```

- JavaScript Object Notation
 - A *lightweight* data-interchange format
 - Client <-> Server
 - Mobile apps / Rich Web apps
- Uses javascript syntax
- Can be used natively with javascript code
 - JSON <-> Objects

2

JSON Example

```
{  
  "to": "John",  
  "from": "Jane",  
  "subject": "Greeting",  
  "body": "Hi John"  
}
```

3

JSON Syntax Rules

Curly brace holders

Name value pairs (colon separated)

Comma separated pairs

Keys – double quoted strings

Values – string, number, object, array, Boolean, null

```
{  
  "to": "John",  
  "from": "Jane",  
  "subject": "Greeting",  
  "body": "Hi John"  
}
```

4

JSON Files / Structure

The file type for JSON files is

`".json"`

The Internet media type is

`"application/json"`

```
{
  "messages": [
    {
      "to": "John",
      "from": "Jane",
      "subject": "Greeting",
      "body": "Hi John"
    },
    {
      "to": "Mick",
      "from": "Joe",
      "subject": "Goodbye",
      "body": "See ya Mick"
    }
  ]
}
```

5

JSON Data Types



String

Must be contained in double quotes



Number

Integer or Floating point



JSON Object

E.g. `"message": { "to": "John", "from": "Jane", "text": "Hello" }`



Array

E.g. `{ "messages": ["Hello", "Goodbye", "Farewell"] }`



Boolean

E.g. `{ "sent": true }`



Null

E.g. `{ "subject": null }`

6

JSON Schema

7

JSON Schema

- **JSON Schema** is a vocabulary that allows you to annotate and validate **JSON documents**.
- A **JSON Schema** is itself **JSON** .
- The **JSON document** being validated or described is called the **instance**, and the document containing the description is called the **schema**.
- Latest *JSON Schema Specification* draft update was December 2020



8

Starting a Schema File

A schema is always placed in a separate JSON document. Currently you **cannot** reference the **schema** from the **instance document**.



```
"$schema": "http://json-schema.org/draft/2019-09/schema#"
```

9

Example JSON Schema

```
{
  "$schema": "http://json-schema.org/draft/2019-09/schema#",
  "title": "message",
  "type": "object",
  "properties": {
    "id": { "type": "string" },
    "to": { "type": "string" },
    "from": { "type": "string" },
    "subject": { "type": "string" },
    "body": { "type": "string" }
  }
}
```

10

JSON Schema Types

String

Length	<pre>{ "type": "string", "minLength": 2, "maxLength": 3 }</pre>
Regular Expression	<pre>{ "type": "string", "pattern": "^(\\([0-9]{3}\\))?[0-9]{3}-[0-9]{4}\$" }</pre>
Built-in Formats	<ul style="list-style-type: none">Dates and TimesEmail addressesIP addressesAnd others...

11

JSON Schema Types

Numeric

Integer	<pre>{ "type": "integer" }</pre>
Number	<pre>{ "type": "number" }</pre>
Multiples	<pre>{ "type": "number", "multipleOf": 10 }</pre>
Range	<pre>{ "type": "number", "minimum": 0, "exclusiveMaximum": 100 }</pre>

12

JSON Schema Types Object

Properties

key-value pairs

Required Properties

Use of "required" keyword

Property names

The names of properties can be validated against a schema irrespective of their values

Size

Use of "minProperties" and "maxProperties"

Dependencies

Property dependencies

declare certain other properties must be present if a given property is present

Schema dependencies

declare that the schema changes when a given property is present

13

JSON Schema Types Array

Items

*Each element in an array **may** be of a different type*

List validation

*a sequence of arbitrary length
where each item matches the same
schema*

Tuple validation

*a sequence of fixed length where
each item may have a different
schema*

Length

Use of "minItems" and "maxItems" keywords

Uniqueness

Use of the "uniqueItems" keyword e.g. array of numbers so that the same number is not repeated.

14

JSON Schema Types **Boolean & Null**

```
{ "type": "boolean" }
```

Value of **true** or **false** (not `"true"` or `"false"`)

```
{ "type": "null" }
```

Value of **null** (not `"null"` in double quotes)
*generally used to represent a missing value
– value must be null.*

15



<https://json-schema.org/>

Further Information

16