

Primer on JSON Schema

Why JSON Schema?

1. Specifying the JSON data structure

Using a spreadsheet to define the JSON structure was too cumbersome. JSON Schema is a better approach since it is designed to be human readable. There are also tools that can take the JSON schema and make it even easier to read and edit.

For example, this the JSON Schema editor I used: <https://www.jsonschema.net/>

The screenshot displays the JSON Schema editor interface. On the left, the 'JSON' tab is active, showing a JSON schema definition for 'The Root Schema'. The schema includes a 'required' array with 'UHC' and a 'properties' object with 'UHC' and 'items'. The 'UHC' property is an array of objects, and the 'items' property is an object with various fields like 'collectDate', 'collectTime', '_profCorp', '_batchID', '_hash', '_batchProcessDate', '_batchProcessTime', '_imageCapture', '_urlSite', and 'allClaims'. On the right, the 'Schema' tab is active, showing a hierarchical tree view of the schema. The tree starts with 'http://example.com/root.json' and branches into '#/properties/UHC' and '#/properties/UHC/items'. The '#/properties/UHC/items' branch further branches into various properties like 'collectDate', 'collectTime', '_profCorp', '_batchID', '_hash', '_batchProcessDate', '_batchProcessTime', '_imageCapture', '_urlSite', and 'allClaims'. Each property in the tree has a 'required' status and a 'key' value.

2. Automating the validation of JSON data

The structures of our various JSON data are getting more complicated and constantly evolving. Checking the JSON structure manually is just too time consuming and error prone.

Since JSON Schema is an open standard with lots support from the open source community, it is a good technology for what we need now.

Once the schema is defined, you can use a validator to validate the JSON data using tools such as <https://jsonschemalint.com>

Below is a screen shot of the validator.

JSON Schema Lint

About

Samples

Reset

Save as Gist

JSON

draft-07

Schema :: JSON, draft-07

Format

```
{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://example.com/root.json",
  "type": "object",
  "title": "The Root Schema",
  "required": [
    "UHC"
  ],
  "properties": {
    "UHC": {
      "$id": "#/properties/UHC",
      "type": "array",
      "title": "UHC output schema",
      "description": "Data output schema for UHC Payer ",
      "default": null,
      "items": {
        "$id": "#/properties/UHC/items",
        "type": "object",
        "required": [
          "dateOfServiceBegin",
          "codeADA",
          "serviceDescription",
          "sedation/analgesia-each subsequent 15 minute increment",
          "toothNumber",
          "submittedFee",
          "coveredFee",
          "deductible",
          "benefitPaid"
        ],
        "properties": {
          "dateOfServiceBegin": {
            "type": "string",
            "format": "date-time",
            "minimum": "2018-10-26"
          },
          "codeADA": {
            "type": "string",
            "enum": [
              "D99243"
            ]
          },
          "serviceDescription": {
            "type": "string",
            "enum": [
              "intravenous moderate (conscious)"
            ]
          },
          "sedation/analgesia-each subsequent 15 minute increment": {
            "type": "string",
            "enum": [
              "01-32"
            ]
          },
          "submittedFee": {
            "type": "number",
            "minimum": 185.00
          },
          "coveredFee": {
            "type": "number",
            "minimum": 39.00
          },
          "deductible": {
            "type": "number",
            "minimum": 0.00
          },
          "benefitPaid": {
            "type": "number",
            "minimum": 29.00
          }
        }
      }
    }
  }
}
```

Message

Schema is valid according to draft-07.

Document :: JSON

Format

```
{
  "dateOfServiceBegin": "2018-10-26",
  "codeADA": "D99243",
  "serviceDescription": "intravenous moderate (conscious)",
  "sedation/analgesia-each subsequent 15 minute increment": "01-32",
  "submittedFee": 185.00,
  "coveredFee": 39.00,
  "deductible": 0.00,
  "benefitPaid": 29.00
}
```

Message

Document validates against the schema, spec version draft-07.

Below is a validation example, the validator detects an incorrect date formatting issue

JSON

draft-07

Format

Document :: JSON

Format

```
{
  "UHC": [
    {
      "collectDate": "2019/01-02",
      "collectTime": "13:50:41",
      "_profCorp": "",
      "_batchID": "Conumser-batch0",
      "_hash": "some hash here",
      "_batchProcessDate": "2018-12-16",
      "_batchProcessTime": "17:56:32+00:00",
      "_dateStarting": "",
      "_dateEnding": "",
      "_imageCapture": "True",
      "_imageFolder": "@paul Aetna",
      "_urlSite": "https://www.dbp.com",
      "allClaims": [
        {
          "claimCheck": [
```

Field	Error	Value
.UHC[0].collectDate	should match format "date"	"2019/01-02"

Below is a validation example where a required field is missing

The screenshot shows a JSON document being validated. The document contains the following fields:

```
{
  "_batchID": "Conumser-batch0",
  "_hash": "some hash here",
  "_batchProcessDate": "2018-12-16",
  "_batchProcessTime": "17:56:32+00:00",
  "_dateStarting": "",
  "_dateEnding": "",
  "_imageCapture": "True",

  "_urlSite": "https://www.dbp.com",
  "allClaims": [
    {
      "claimCheck": [
        {
          "_profCorp": "ConsumerHealth, Inc.",
          "_insCo": "UNITED HEALTHCARE/DBP",
          "_clinicID": "28",
          "_subID": "949068445",
          "_patientName": "GONZALEZ BARBA,MARIA",
          "_subFirstName": ""
        }
      ]
    }
  ]
}
```

The validation tool has identified an error in the first element of the `allClaims` array. The error message is: "should have required property '_imageFolder'". The value of the field is the entire object shown above.

Field	Error	Value
.UHC[0]	should have required property '_imageFolder'	{ "collectDate": "2019-01-0: "collectTime": "13:50:41" " profCorp": ""

3. Speed up development cycle

We want to enable each developer to able to test and iterate their own code as quickly as possible. Json Schema allows each developer to independently checking their own JSON data structures without depending on other team members.

4. Ease integration with our automated collection process

By ensuring a standard interface, we are able to more easily integrate the collectors developed into our system.

Other resources

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

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