LEARNING

jsonschema

#jsonschem

a

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: jsonschema

It is an unofficial and free jsonschema ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official jsonschema.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with jsonschema

Remarks

What is JSON Schema?

JSON Schema is a vocabulary that allows you to validate, annotate, and manipulate JSON documents.

A JSON Schema is itself a JSON document and requires a third party library to validate data against it.

It is currently a draft IETF standard allowing consistent expectation from implementations based on the standard specification.

Vocabularies

Published vocabularies

- JSON Schema Core the basic foundation of JSON Schema
- JSON Schema Validation the validation keywords of JSON Schema
- JSON Hyper-Schema the hyper-media keywords of JSON Schema

Proposed vocabularies requiring discussion, feedback and review

- JSON Schema API API documentation keyword proposal
- JSON Schema UI the UI keywords for consistent UI generation proposal

Implementations

A list of implementations is maintained in the GitHub repository. The list of supported languages currently lists validators for:

> ActionScript 3, C, C++, Clojure, Dart, Erlang, Go, Haskell, Java, JavaScript, .NET, PHP, Perl, Python and Ruby

There are also UI generators, data parsers, schema editors, documentation generators and IDE support.

Examples

Basic example validation schema

```json
{
    "title": "Person",
    "type": "object",
    "properties": {
        "firstName": {
            "type": "string"
        },
        "lastName": {
            "type": "string"
        },
        "age": {
            "description": "Age in years",
            "type": "integer",
            "minimum": 0
        }
    },
    "required": ["firstName", "lastName"]
}
```

Results

```
// Valid
{
    "firstName": "Jason",
    "lastName": "Voorhees"
}
// Valid
{
    "firstName": "Jason",
    "lastName": "Voorhees",
    "age": 47
}
// Invalid – no lastName
{
    "firstName": "Jason",
    "age": 47
}
// Invalid – age is not integer
{
    "firstName": "Jason",
    "lastName": "Voorhees",
    "age": "47"
}
```

Read Getting started with jsonschema online:
https://riptutorial.com/jsonschema/topic/10835/getting-started-with-jsonschema

Credits

S. No	Chapters	Contributors
1	Getting started with jsonschema	Anthropic, Community