

Building Blocks for HTTP APIs
Internet-Draft
Intended status: Standards Track
Expires: 3 January 2026

C. Vasters
Microsoft Corporation
2 July 2025

JSON Structure: Alternate Names and Descriptions
draft-vasters-json-structure-alternate-names-00

Abstract

This document is an extension to JSON Structure Core. It defines three annotation keywords, `altnames`, `altnums`, and `descriptions`, which allow schema authors to provide alternative identifiers, display names, and multi-variant descriptions for types, properties, and enumeration values.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://json-structure.github.io/alternate-names/draft-vasters-json-structure-alternate-names.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-vasters-json-structure-alternate-names/>.

Source for this draft and an issue tracker can be found at <https://github.com/json-structure/alternate-names>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 January 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Conventions	3
1.2. Keywords	3
1.2.1. The altnames Keyword	3
1.2.2. The altnums Keyword	4
1.2.3. The descriptions Keyword	5
1.3. Enabling the Annotations	7
1.4. Security and Interoperability Considerations	7
1.5. IANA Considerations	8
2. Normative References	8
Acknowledgments	8
Author's Address	8

1. Introduction

This document is an extension to JSON Structure Core [JSTRUCT-CORE]. It defines three annotation keywords, `altnames`, `altnums`, and `descriptions`, which allow schema authors to provide alternative identifiers, display names, and multi-variant descriptions for types, properties, and enumeration values.

These annotations facilitate mapping between internal schema identifiers and external data representations (e.g., JSON keys that do not conform to identifier rules) and support internationalization by enabling localized labels.

1.1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Keywords

This section defines the alternate names and symbols annotations.

1.2.1. The altnames Keyword

The altnames keyword provides alternative names for a named type or property. Alternate names are not subject to the identifier syntax restrictions imposed on name and MAY be used to map internal schema names to external representations.

The value of altnames MUST be a map where each key is a purpose indicator and each value is a string representing an alternate name.

- * The key "json" is RESERVED. When used, it specifies the property key to use for a property when encoded in JSON. This allows for JSON keys that do not conform to identifier rules.
- * Keys starting with the prefix "lang:" (e.g., "lang:en", "lang:fr") are RESERVED for localized display names. The suffix after the colon specifies the language code. The language code MUST conform to [RFC4646].
- * Other keys are allowed for custom usage, provided they do not conflict with the reserved keys or prefixes.

The altnames keyword MAY be included in any schema element that has an explicit name (i.e., named types or properties). When present, the alternate names provide additional mappings that schema processors MAY use for encoding, decoding, or user interface display.

Example:

```
{
  "Person": {
    "type": "object",
    "altnames": {
      "json": "person_data",
      "lang:en": "Person",
      "lang:de": "Person"
    },
    "properties": {
      "firstName": {
        "type": "string",
        "altnames": {
          "json": "first_name",
          "lang:en": "First Name",
          "lang:de": "Vorname"
        }
      },
      "lastName": {
        "type": "string",
        "altnames": {
          "json": "last_name",
          "lang:en": "Last Name",
          "lang:de": "Nachname"
        }
      }
    },
    "required": ["firstName", "lastName"]
  }
}
```

1.2.2. The altnums Keyword

The altnums keyword provides alternative representations (symbols) for enumeration values defined by a type using the enum keyword. Alternate symbols allow schema authors to map internal enum values to external codes or localized display symbols.

The value of altnums MUST be a JSON object (map) where each key is a purpose indicator and each corresponding value is an object mapping each enumeration value (as defined in the enum array) to its alternate symbol.

- * The key "json" is RESERVED and MUST be used to specify alternate symbols for JSON encoding.
- * Keys beginning with "lang:" (e.g., "lang:en", "lang:es") are RESERVED for providing localized alternate symbols.

- * Additional keys are permitted for custom usage, subject to no conflicts with reserved keys or prefixes.

The `altnums` keyword **MUST** be used only with schemas that include an `enum` keyword. When present, it provides alternative representations for each enumeration value that schema processors **MAY** use for encoding or display purposes.

```
{
  "Color": {
    "type": "string",
    "enum": ["RED", "GREEN", "BLUE"],
    "altnums": {
      "json": {
        "RED": "#FF0000",
        "GREEN": "#00FF00",
        "BLUE": "#0000FF"
      },
      "lang:en": {
        "RED": "Red",
        "GREEN": "Green",
        "BLUE": "Blue"
      },
      "lang:de": {
        "RED": "Rot",
        "GREEN": "Grn",
        "BLUE": "Blau"
      }
    }
  }
}
```

1.2.3. The descriptions Keyword

The `descriptions` keyword provides multi-variant descriptions as an alternative to the `description` keyword. It allows schema authors to provide localized or context-specific descriptions for types, properties, or enumeration values.

The value of `descriptions` **MUST** be a map where each key is a purpose indicator and each value is a string representing a description.

- * Keys beginning with `"lang:"` (e.g., `"lang:en"`, `"lang:fr"`) are **RESERVED** for localized descriptions. The suffix after the colon specifies the language code. The language code **MUST** conform to [RFC4646].

- * Other keys are allowed for custom usage, provided they do not conflict with the reserved keys or prefixes.

The descriptions keyword MAY be included in any schema element that can have a description. When present, the descriptions provide additional mappings that schema processors MAY use for user interface display.

Example:

```
{
  "Person": {
    "type": "object",
    "descriptions": {
      "lang:en": "A person object with first and last name",
      "lang:de": "Ein Person-Objekt mit Vor- und Nachnamen",
      "lang:fr": "Un objet personne avec prnom et nom de famille",
      "lang:cn": "一个有名字和姓氏的人象",
      "lang:jp": "名前と姓を持つ人物オブジェクト"
    },
    "properties": {
      "firstName": {
        "type": "string",
        "descriptions": {
          "lang:en": "The first name of the person",
          "lang:de": "Der Vorname der Person",
          "lang:fr": "Le prnom de la personne",
          "lang:cn": "人的名字",
          "lang:jp": "人の名前"
        }
      },
      "lastName": {
        "type": "string",
        "descriptions": {
          "lang:en": "The last name of the person",
          "lang:de": "Der Nachname der Person",
          "lang:fr": "Le nom de famille de la personne",
          "lang:cn": "人的姓氏",
          "lang:jp": "人の姓"
        }
      }
    },
    "required": ["firstName", "lastName"]
  }
}
```

1.3. Enabling the Annotations

Alternate names and symbols annotations can be enabled in a schema or meta-schema by adding the JSONSchemaAlternateNames key to the \$uses clause when referencing the extended meta-schema:

```
{
  "$schema": "https://json-structure.org/meta/extended/v0/#",
  "$id": "myschema",
  "$uses": [
    "JSONStructureAlternateNames"
  ],
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "altnames": {
        "lang:fr": "Nom"
      }
    }
  }
}
```

The annotation are enabled by default in the validation meta-schema:

```
{
  "$schema": "https://json-structure.org/meta/validation/v0/#",
  "$id": "myschema",
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "altnames": {
        "lang:fr": "Nom"
      }
    }
  }
}
```

1.4. Security and Interoperability Considerations

Alternate names and symbols annotations do not affect the validation of instance data. They are purely metadata and MUST be ignored by validators that do not support this extension. However, applications that rely on alternate names for mapping or localization MUST implement appropriate safeguards to ensure that the alternate identifiers are used consistently.

1.5. IANA Considerations

This document has no IANA actions.

2. Normative References

[JSTRUCT-CORE]

Vasters, C., "JSON Structure Core", n.d., <<https://json-structure.github.io/core/draft-vasters-json-structure-core.html>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.

[RFC4646] Phillips, A. and M. Davis, "Tags for Identifying Languages", RFC 4646, DOI 10.17487/RFC4646, September 2006, <<https://www.rfc-editor.org/rfc/rfc4646>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

Acknowledgments

TODO acknowledge.

Author's Address

Clemens Vasters
Microsoft Corporation
Email: clemensv@microsoft.com