HTTPAPI
Internet-Draft
Intended status: Informational
Expires: 8 May 2026

R. Polli
Par-Tec

4 November 2025

REST API Media Types
draft-ietf-httpapi-rest-api-mediatypes-08

Abstract

This document registers the following media types used in APIs on the IANA Media Types registry: application/openapi+json, and application/openapi+yaml.

About This Document

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

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-httpapi-rest-api-mediatypes/.

Discussion of this document takes place on the HTTPAPI Working Group mailing list (mailto:httpapi@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/httpapi/. Subscribe at https://www.ietf.org/mailman/listinfo/httpapi/. Working Group information can be found at https://datatracker.ietf.org/wg/httpapi/about/.

Source for this draft and an issue tracker can be found at https://github.com/ietf-wg-httpapi/mediatypes/labels/rest-api.

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 8 May 2026.

Table of Contents

1. Introduction

OpenAPI Specification [OAS] version 3 and above is a consolidated standard for describing HTTP APIs using the JSON [JSON] and YAML [YAML] data format.

To increase interoperability when processing API descriptions and leverage content negotiation mechanisms when exchanging OpenAPI description representations this specification registers the following media types: application/openapi+json and application/openapi+yaml.

1.1. Notational 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. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

The terms "content", "content negotiation", "resource", and "user agent" in this document are to be interpreted as in [HTTP].

2. Media Type registrations

This section describes the information required to register the above media types according to [MEDIATYPE].

2.1. The OpenAPI Media Types

The OpenAPI Specification Media Types convey semantics for OpenAPI Document (OAD) resources as defined in [OAS] for version 3.0 and above.

Those resources can be represented in [JSON] or [YAML]. Since there are multiple OpenAPI Specification versions, those media types support the version parameter.

The following example conveys the desire of a client to receive an OpenAPI Document resource based on the stated preferences:

1. openapi 3.1 in YAML

2. openapi 3.0 in YAML

3. any openapi version in JSON

```
Accept: application/openapi+yaml;version=3.1,
        application/openapi+yaml;version=3.0;q=0.5,
        application/openapi+json;q=0.3
```

2.1.1. Media Type application/openapi+json

The following information serves as the registration form for the application/openapi+json media type.

Type name: application

Subtype name: openapi+json

Required parameters: None

Optional parameters: version: its value is a string representing the OpenAPI Specification version. ; unrecognized parameters should be ignored

Encoding considerations: Same as "application/json"

Security considerations: See Section 4 of this document, "application/json" and [OAS]

Interoperability considerations: See Section 3 of this document, "application/json" and [OAS]

Published specification: this document, [OAS]

Applications that use this media type: HTTP

Fragment identifier considerations: [OAS] or the specific version of the OpenAPI document.

Additional information:

* Deprecated alias names for this type: "application/vnd.oai.openapi+json". This name is used, but not registered.

* Magic number(s): N/A

* File extension(s): json

* Macintosh file type code(s): N/A

Person and email address to contact for further information: See Authors' Addresses section.

Intended usage: COMMON

Restrictions on usage: None.

Author: See Authors' Addresses section.

Change controller: IETF

2.1.2. Media Type application/openapi+yaml

The following information serves as the registration form for the application/openapi+yaml media type.

Type name: application

Subtype name: openapi+yaml

Required parameters: N/A

Optional parameters: version: its value is a string representing the OpenAPI Specification version. ; unrecognized parameters should be ignored

Encoding considerations: Same as "+yaml" Structured Syntax Suffix

Security considerations: See Section 4 of this document, "+yaml" Structured Syntax Suffix and [OAS]

Interoperability considerations: See Section 3 of this document, "+yaml" Structured Syntax Suffix and [OAS]

Published specification: [OAS]

Applications that use this media type: HTTP

Fragment identifier considerations: [OAS] or the specific version of the OpenAPI document.

Additional information:

* Deprecated alias names for this type: "application/vnd.oai.openapi". This name is used, but not registered.
* Magic number(s): N/A
* File extension(s): Same as "application/yaml"
* Macintosh file type code(s): N/A

Person and email address to contact for further information: See Aut hors' Addresses section

Intended usage: COMMON

Restrictions on usage: None.

Author: See Authors' Addresses section

Change controller: IETF

3. Interoperability Considerations

Interoperability requirements for media type registrations are discussed in Section 4.6 of [MEDIATYPE] and in the Interoperability Considerations of the "+yaml" Structured Syntax Suffix.

3.1. Using the version parameter

The version parameter ought to be processed according with the associated OpenAPI Specification.

For example, when an OpenAPI 3.1 resource uses the patch version version=3.1.1, its value is expected to be ignored by tooling (see https://spec.openapis.org/oas/v3.1.0.html#versions).

4. Security Considerations

Security requirements for media type registrations are discussed in Section 4.6 of [MEDIATYPE]. and in the Security Considerations of the "+yaml" Structured Syntax Suffix.

4.1. General Considerations

OpenAPI documents are processed by a wide variety of tooling for numerous different purposes, such as client code generation, documentation generation, server side routing, and API testing. OpenAPI document authors must consider the risks of the scenarios where the OpenAPI document may be used.

An OpenAPI document describes the security schemes used to protect the resources it defines. The security schemes available offer varying degrees of protection. Factors such as the sensitivity of the data and the potential impact of a security breach should guide the selection of security schemes for the API resources. Some security schemes, such as basic auth and OAuth Implicit flow, are supported for compatibility with existing APIs. However, their inclusion in OpenAPI does not constitute an endorsement of their use, particularly for highly sensitive data or operations.

OpenAPI documents may contain references to external resources that may be dereferenced automatically by consuming tools. External resources may be hosted on different domains that may be untrusted. References in an OpenAPI document, or across OpenAPI documents may cause a cycle. Tooling must detect and handle cycles to prevent resource exhaustion.

Certain properties allow the use of Markdown which can contain HTML including script. It is the responsibility of tooling to appropriately sanitize the Markdown.

OpenAPI documents use [jsonschema] therefore share the security consideration of JSON Schema.

5. IANA Considerations

This specification defines the following new Internet media types [MEDIATYPE].

IANA is asked to update the "Media Types" registry at https://www.iana.org/assignments/media-types (https://www.iana.org/assignments/media-types) with the registration information provided in the sections below.

Media Type	Registration information section
application/openapi+json	Section 2.1.1 of this document
application/openapi+yaml	Section 2.1.2 of this document

Table 1

6. Normative References

[HTTP] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

[JSON] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

[jsonschema]
Wright, A., Andrews, H., Hutton, B., and G. Dennis, "JSON Schema", 28 January 2020, <https://json-schema.org/specification.html>.

[MEDIATYPE]
Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/rfc/rfc6838>.

[OAS] Darrel Miller, Jeremy Whitlock, Marsh Gardiner, Mike Ralphson, Ron Ratovsky, and Uri Sarid, "OpenAPI Specification 3.1.0", 15 February 2021, <https://spec.openapis.org/oas/latest>.

[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>.

[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>.

[YAML] Oren Ben-Kiki, Clark Evans, and Ingy dot Net, "YAML Ain’t Markup Language Version 1.2", 1 October 2021, <https://yaml.org/spec/1.2/spec.html>.

Acknowledgements

Thanks to Erik Wilde and David Biesack for being the initial contributors of this specification, and to Darrel Miller and Rich Salz for their support during the adoption phase.

In addition to the people above, this document owes a lot to the extensive discussion inside and outside the HTTPAPI workgroup. The following contributors have helped improve this specification by opening pull requests, reporting bugs, asking smart questions, drafting or reviewing text, and evaluating open issues:

Austin Wright, Ben Hutton and Jason Desrosiers.

FAQ

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

Q: Why this document? After all these years, we still lack a proper

media type for REST related document types. This has some security implications too (eg. wrt on identifying parsers or treat downloads)

Change Log

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

RFC EDITOR PLEASE DELETE THIS SECTION.

Since -00

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

* Split YAML registrations in a separate I-D.

Since -04

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

* Split JSONSCHEMA registrations in a separate I-D.

Author’s Address

Roberto Polli
Par-Tec
Italy
Email: robipolli@gmail.com