

Workgroup: Network Working Group
Internet-Draft:
draft-polli-rest-api-mediatypes-00

Published: 15 September 2021

Intended Status: Informational

Expires: 19 March 2022

Authors: R. Polli

Digital Transformation Department, Italian Government

REST API mediatypes

Abstract

This document register the following media-types used in APIs on the IANA MEdia Types registry: text/yaml, application/yaml, application/openapi+json, and application/openapi+yaml

Note to Readers

RFC EDITOR: please remove this section before publication

Discussion of this draft takes place on the HTTP working group mailing list (httppapi@ietf.org), which is archived at <https://lists.w3.org/Archives/Public/ietf-httpapi-wg/>.

The source code and issues list for this draft can be found at <https://github.com/ioggstream/draft-polli-rest-api-mediatypes>.

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 19 March 2022.

Copyright Notice

Copyright (c) 2021 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1. Introduction](#)
 - [1.1. Notational Conventions](#)
 - [1.2. The OpenAPI Media Types](#)
- [2. Security Considerations](#)
- [3. IANA Considerations](#)
 - [3.1. application/yaml](#)
 - [3.2. text/yaml](#)
 - [3.3. application/openapi+json](#)
 - [3.4. application/openapi+yaml](#)
- [4. Normative References](#)
- [Appendix A. Acknowledgements](#)
- [FAQ](#)
- [Change Log](#)
- [Author's Address](#)

1. Introduction

OpenAPI [[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 specifications and leverage content negotiation mechanisms when exchanging OpenAPI resources this specification registers the following media-types: text/yaml, application/yaml, application/openapi+json and application/openapi+yaml.

The

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.

This document uses the Augmented BNF defined in [[RFC5234](#)] and updated by [[RFC7405](#)].

1.2. The OpenAPI Media Types

The OpenAPI Media Types convey OpenAPI specification files as defined in [[oas](#)] for version 3.0.0 and above.

Those files can be serialized in JSON or [[yaml](#)]. Since there are multiple OpenAPI Specifications versions, those media-types support the version parameter.

The following examples conveys the desire of a client to receive an OpenAPI resource preferably in the following order:

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. Security Considerations

Security requirements for both media type and media type suffix registrations are discussed in Section 4.6 of [[MEDIATYPE](#)].

3. IANA Considerations

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

3.1. application/yaml

Type name: application

Subtype name: yaml

Required parameters: None

Optional parameters: None; unrecognized parameters should be ignored

Encoding considerations: Same as [[JSON](#)]

Security considerations: see [Section 2](#) of this document

Interoperability considerations: None

Published specification: (this document)

Applications that use this media type: HTTP

Fragment identifier considerations: Same as for application/json
[[JSON](#)]

Additional information:

Deprecated alias names for this type: application/x-yaml

Magic number(s): n/a

File extension(s): yaml, yml

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: n/a

3.2. text/yaml

Type name: text

Subtype name: yaml

Required parameters: None

Optional parameters: None; unrecognized parameters should be ignored

Encoding considerations: Same as [[JSON](#)]

Security considerations: see [Section 2](#) of this document

Interoperability considerations: None

Published specification: (this document)

Applications that use this media type: HTTP

Fragment identifier considerations: Same as for application/json
[[JSON](#)]

Additional information:

Deprecated alias names for this type: text/x-yaml

Magic number(s): n/a

File extension(s): yaml, yml

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: n/a

3.3. application/openapi+json

Type name: application

Subtype name: openapi+json

Required parameters: None

Optional parameters: version; unrecognized parameters should be
ignored

Encoding considerations: Same as [[JSON](#)]

Security considerations: see [Section 2](#) of this document

Interoperability considerations: None

Published specification: (this document)

Applications that use this media type: HTTP

Fragment identifier considerations: Same as for application/json
[[JSON](#)]

Additional information:

Deprecated alias names for this type: n/a

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: n/a

3.4. application/openapi+yaml

Type name: application

Subtype name: openapi+yaml

Required parameters: None

Optional parameters: version; unrecognized parameters should be ignored

Encoding considerations: Same as [[JSON](#)]

Security considerations: see [Section 2](#) of this document

Interoperability considerations: None

Published specification: (this document)

Applications that use this media type: HTTP

Fragment identifier considerations: Same as for application/json [[JSON](#)]

Additional information:

Deprecated alias names for this type: n/a

Magic number(s): n/a

File extension(s): yaml, yml

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: n/a

4. Normative References

[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/info/rfc8259>>.

[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/info/rfc6838>>.

[oas] Darrel Miller, ., Jeremy Whitlock, ., Marsh Gardiner, ., Mike Ralphson, ., Ron Ratovsky, ., and . Uri Sarid, "OpenAPI Specification 3.0.0", 26 July 2017.

[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/info/rfc2119>>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.

[RFC7405] Kyzivat, P., "Case-Sensitive String Support in ABNF", RFC 7405, DOI 10.17487/RFC7405, December 2014, <<https://www.rfc-editor.org/info/rfc7405>>.

[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/info/rfc8174>>.

[yaml] Oren Ben-Kiki, ., Clark Evans, ., and . Ingy doet Net, "YAML Ain't Markup Language Version 1.2", 1 October 2002, <<https://yaml.org/spec/1.2/spec.html>>.

Appendix A. Acknowledgements

This specification was born from a thread created by James Manger and the subsequent discussion here <https://github.com/httpwg/http-extensions/issues/885>.

FAQ

Q: Why this document? After all these years, we still lack a proper media-type for yaml. This has some security implications too (eg. wrt on identifying parsers or treat downloads)

Q: Why application/yaml and text/yaml Browsers and libraries implementations treats them differently. For example Google Chrome will display pages with Content-Type: text/yaml and to download pages with Content-Type: application/yaml.

Change Log

RFC EDITOR PLEASE DELETE THIS SECTION.

Author's Address

Roberto Polli
Digital Transformation Department, Italian Government
Italy

Email: robipolli@gmail.com