api-catalog: a well-known URI and link relation to help discovery of APIs

Abstract

This document defines the "api-catalog" well-known URI and link relation. It is intended to facilitate automated discovery and usage of the APIs published by a given organisation or individual. A request to the api-catalog resource will return a document providing information about, and links to, the publisher's APIs.

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://ietf-wg-httpapi.github.io/api-catalog/draft-ietf-httpapi-api-catalog.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-httpapi-api-catalog/.

Discussion of this document takes place on the Building Blocks for HTTP APIs 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/.

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

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 9 January 2025.

Copyright Notice

Copyright (c) 2024 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
   1.1. Terminology
   1.2. Goals and non-goals
   1.3. Requirements Language
2. Using the 'api-catalog' well-known URI
3. Link relations
4. Accounting for APIs distributed across multiple domains
5. Internal use of api-catalog for private APIs
6. The API Catalog
7. Conformance to RFC8615
   7.1. Path prefix
   7.2. Supported URI schemes
   7.3. Registration of the api-catalog well-known URI
8. IANA Considerations
   8.1. The api-catalog well-known URI
   8.2. The api-catalog link relation
   8.3. the api-catalog Profile URI
9. Security Considerations
10. References
   10.1. Normative References
   10.2. Informative References
Appendix A. Example API Catalog document
   A.1. Using Linkset with RFC8615 relations
   A.2. Using Linkset with bookmarks
Appendix B. Acknowledgements
Author's Address
1. Introduction

An organisation or individual may publish Application Programming Interfaces (APIs) to encourage requests for interaction from external parties. Such APIs must be discovered before they may be used - i.e., the external party needs to know what APIs a given publisher exposes, their purpose, any policies for usage, and the endpoint to interact with each APIs. To facilitate automated discovery of this information, and automated usage of the APIs, this document proposes:

*a well-known URI, 'api-catalog', as a reference to the URI of an API Catalog document describing a Publisher's API endpoints.

*a link relation, 'api-catalog', of which the target resource is the Publisher's API Catalog document.

1.1. Terminology

*'Publisher' - an organisation, company or individual that publishes one or more APIs for usage by external third parties.

1.2. Goals and non-goals

The primary goal is to facilitate the automated discovery of a Publisher's public API endpoints, along with metadata that describes the purpose and usage of each API, by specifying a well-known URI [RFC8615] that returns an API catalog document. The API catalog document is primarily machine-readable to enable automated discovery and usage of APIs, and it may also include links to human-readable documentation.

Non-goals: this document does not mandate paths for API endpoints. i.e., it does not mandate that my_example_api's endpoint should be example.com/.well-known/api-catalog/my_example_api, nor even to be hosted at example.com (although it is not forbidden to do so). This document does not mandate a specific format for the API catalog document, although it does suggest some existing formats and provide a recommendation.

1.3. Requirements Language

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.
2. Using the 'api-catalog' well-known URI

The api-catalog well-known URI is intended for HTTP(S) servers that publish APIs. As the key aim is to facilitate discovery and usage of APIs, a Publisher supporting this URI:

*SHOULD publish the ./well-known/api-catalog URI at a predictable location. For example as companies typically own a .com TLD, a predictable location for the company 'example' would be https://www.example.com/.well-known/api-catalog

*SHALL resolve an HTTP(S) GET request to ./well-known/api-catalog and return an API catalog document (as described in Section 6).

*SHOULD resolve an HTTP(S) HEAD request to ./well-known/api-catalog with a response including a Link header with the relation(s) defined in Section 3.

The location (URL) of the API Catalog document is decided by the Publisher: the ./well-known/api-catalog URI provides a convenient reference to that URL.

3. Link relations

'*api-catalog': the 'api-catalog' link relation identifies a target resource that represents a list of APIs available from the Publisher of the context resource. The target resource URI may be ./well-known/api-catalog, or any other URI chosen by the Publisher. For example, the Publisher 'example.com' could include the api-catalog link relation in the HTTP header and/or content payload when responding to a request to https://example.com:
**"item"** [RFC9264]. When used in an API Catalog document, the 'item' link relation identifies a target resource that represents an API that is a member of the API Catalog.

### 4. Accounting for APIs distributed across multiple domains

A Publisher ('example') may have their APIs hosted across multiple domains that they manage: e.g., at example.com, developer.example.com, apis.example.com, apis.example.net etc. They may also use a third party API hosting provider which hosts APIs on a distinct domain.

To account for this scenario, it is recommended that:

* the Publisher publish the api-catalog well-known URI at a predictable location, e.g. example.com/.well-known/api-catalog.

* the Publisher also publish the api-catalog well-known URI at each of their API domains e.g. apis.example.com/.well-known/api-catalog, developer.example.net/.well-known/api-catalog etc.

* an HTTP GET request to any of these URIs should return the same result, namely, the API Catalog document.

* Since the physical location (URL) of the API Catalog document is decided by the Publisher, and may change, it is RECOMMENDED that the Publisher choose one of their instances of .well-known/api-catalog as a canonical reference to the location of the latest API Catalog. The Publisher's other instances of ./well-known/api-catalog SHOULD redirect to this canonical instance of ./well-known/api-catalog, using HTTP Status Code 308 Permanent Redirect [RFC9110], to ensure the latest API Catalog is returned.
As illustration, if the Publisher's primary API portal is apis.example.com, then apis.example.com/.well-known/api-catalog should resolve to the location of the latest API Catalog document. If the Publisher is also the domain authority for example.net, which also hosts a selection of their APIs, then a request to www.example.net/.well-known/api-catalog SHOULD return a redirect as follows.

Client request:

GET /.well-known/api-catalog HTTP/1.1
User-Agent: curl/7.16.3 libcurl/7.16.3 OpenSSL/0.9.7l zlib/1.2.3
Host: www.example.net

Server response:

HTTP/1.1 308 Permanent Redirect
Content-Type: text/html; charset=UTF-8
Location: http://apis.example.com/.well-known/api-catalog
Content-Length: 356

<!DOCTYPE HTML>
<html>
<head>
    <title>Permanent Redirect</title>
    <meta http-equiv="refresh" content="0; url=https://apis.example.com/.well-known/api-catalog">
</head>
<body>
</body>
</html>

5. Internal use of api-catalog for private APIs

A Publisher may wish to use the api-catalog well-known URI on their internal network, to signpost authorised users (e.g. company employees) towards internal/private APIs not intended for third-party use. This scenario may incur additional security considerations, as noted in Section 9

6. The API Catalog

The API Catalog is a document listing hyperlinks to a Publisher's APIs. The Publisher may host this API Catalog document at any URI(s) they choose. Hence the API Catalog document URI of example.com/my_api_catalog.json can be requested directly, or via a request to example.com/.well-known/api-catalog, which the Publisher will resolve to example.com/my_api_catalog.
There is no mandated format for the API Catalog document: the Publisher is free to choose any format that supports the automated discovery, and machine (and human) usage of their APIs. However, it is RECOMMENDED to use a linkset [RFC9264] of API endpoints (see Appendix A for an example).

The API Catalog document MUST include hyperlinks to API endpoints, and is RECOMMENDED to include useful metadata, such as usage policies, API version information, links to the OpenAPI Specification [OAS] definitions for each API, etc. If the Publisher does not include these metadata directly in the API Catalog document, they SHOULD make that metadata available at the API endpoint URIs they have listed (see Appendix A.2 for an example).

Some suitable API Catalog document formats include:

*(RECOMMENDED) A linkset [RFC9264] of API endpoints and information to facilitate API usage. The linkset SHOULD include a profile parameter (section 5 of [RFC9264]) with the Profile URI 'THIS-RFC-URL' to indicate the linkset is representing an API Catalog document as defined above.

*An APIs.json document [APIsjson]

*API bookmarks that represent an API entry-point URI, which may be followed to discover purpose and usage

*A RESTDesc semantic description for hypermedia APIs [RESTdesc]

*A Hypertext Application Language document [HAL]

*An extension to the Schema.org WebAPI type [WebAPIext]

Appendix A includes example API Catalog documents based on the linkset format.

7. Conformance to RFC8615

The requirements in section 3 of [RFC8615] for defining Well-Known Uniform Resource Identifiers are met as follows:

7.1. Path prefix

The api-catalog URI SHALL be appended to the /.well-known/ path-prefix for "well-known locations".

7.2. Supported URI schemes

The api-catalog well-known URI may be used with the HTTP and HTTPS URI schemes.
7.3. Registration of the api-catalog well-known URI

See Section 8 considerations below.

8. IANA Considerations

8.1. The api-catalog well-known URI

This specification registers the "api-catalog" well-known URI in the Well-Known URI Registry as defined by [RFC8615].

*URI suffix: api-catalog

*Change Controller: IETF

*Specification document(s): THIS-RFC

*Status: permanent

*Related information: The "api-catalog" documents obtained from the same host using the HTTP and HTTPS protocols (using default ports) MUST be identical.

8.2. The api-catalog link relation

This specification registers the "api-catalog" link relation by following the procedures per section 4.2.2 of [RFC8288].

*Relation Name: api-catalog

*Description: Identifies a catalog of APIs published by the context Publisher.

*Reference: THIS-RFC

8.3. the api-catalog Profile URI

This specification registers "THIS-RFC-URL" in the "Profile URIs" registry according to [RFC7284].

*Profile URI: THIS-RFC-URL

*Common Name: API Catalog

*Description: A profile URI to request or signal a linkset representing an API Catalog.

*Reference: THIS-RFC
9. Security Considerations

For all scenarios: the Publisher SHOULD perform a security and privacy review of the API Catalog prior to deployment, to ensure it does not leak personal, business or other metadata, nor expose any vulnerability related to the APIs listed.

For the internal/private APIs scenario: the Publisher SHOULD take steps to ensure that appropriate access controls are in place to ensure only authorised users access the internal api-catalog well-known URI.

10. References

10.1. Normative References


10.2. Informative References


Appendix A. Example API Catalog document

This section is informative, and provides an example of an API Catalog document using the RECOMMENDED linkset format.

A.1. Using Linkset with RFC8615 relations

This example uses the linkset format [RFC9264], and the following link relations defined in [RFC8631]:

**"service-desc"**, used to link to a description of the API that is primarily intended for machine consumption.

**"service-doc"**, used to link to API documentation that is primarily intended for human consumption.

**"service-meta"**, used to link to additional metadata about the API, and is primarily intended for machine consumption.
**"status", used to link to the API status (e.g. API "health" indication etc.) for machine and/or human consumption.

Client request:

GET .well-know/api-catalog HTTP/1.1
Host: example.com
Accept: application/linkset+json

Server response:

HTTP/1.1 200 OK
Date: Mon, 01 Jun 2023 00:00:01 GMT
Server: Apache-Coyote/1.1
Content-Type: application/linkset+json;
    profile="THIS-RFC-URL"
{
  "linkset": [
    {
      "anchor": "https://developer.example.com/apis/foo_api",
      "service-desc": [
        {
          "href": "https://developer.example.com/apis/foo_api/spec",
          "type": "application/yaml"
        }
      ],
      "status": [
        {
          "href": "https://developer.example.com/apis/foo_api/status",
          "type": "application/json"
        }
      ],
      "service-doc": [
        {
          "href": "https://developer.example.com/apis/foo_api/doc",
          "type": "text/html"
        }
      ],
      "service-meta": [
        {
          "href": "https://developer.example.com/apis/foo_api/policies",
          "type": "text/xml"
        }
      ]
    },
    {
      "anchor": "https://developer.example.com/apis/bar_api",
      "service-desc": [
        {
          "href": "https://developer.example.com/apis/bar_api/spec",
          "type": "application/yaml"
        }
      ],
      "status": [
        {
          "href": "https://developer.example.com/apis/bar_api/status",
          "type": "application/json"
        }
      ],
      "service-doc": [
        {
          "href": "https://developer.example.com/apis/bar_api/doc",
          "type": "text/plain"
        }
      ]
    }
  ]
}
A.2. Using Linkset with bookmarks

This example also uses the linkset format [RFC9264], listing the API endpoints in an array of bookmarks. Each link shares the same context (the API Catalog) and "item" [RFC9264] link relation (to indicate they are an item in the catalog). The intent is that by following a bookmark link, a machine-client can discover the purpose and usage of each API, hence the document targeted by the bookmark link should support this.

Note in the example below, the context anchor is example/com/.well-known/api-catalog, however as explained above the context anchor may be any other URI at which the api-catalog is available.

Client request:

GET .well-know/api-catalog HTTP/1.1
Host: example.com
Accept: application/linkset+json

Server response:

HTTP/1.1 200 OK
Date: Mon, 01 Jun 2023 00:00:01 GMT
Server: Apache-Coyote/1.1
Content-Type: application/linkset+json;
profile="THIS-RFC-URL"
Appendix B. Acknowledgements

Thanks to Phil Archer, Ben Bucksch, Sanjay Dalal, Max Maton, Darrel Miller, Mark Nottingham, Roberto Polli, Rich Salz, Herbert Van De Sompel and Erik Wilde for their suggestions and feedback.

Author's Address

Kevin Smith
Vodafone

Email: kevin.smith@vodafone.com
URI: https://www.vodafone.com