

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 25 October 2026

M. Nottingham
23 April 2026

Feed Menus
draft-nottingham-feed-menu-00

Abstract

This specification defines Feed Menus, a simplified means of discovering the feeds (e.g., RSS or Atom) offered by a Web site.

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-nottingham-feed-menu/>.

information can be found at <https://mnot.github.io/I-D/>.

Source for this draft and an issue tracker can be found at
<https://github.com/mnot/I-D/labels/feed-menu>.

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 25 October 2026.

Copyright Notice

Copyright (c) 2026 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. Notational Conventions	3
2. Feed Menu Documents	3
2.1. Menu Objects	4
2.2. Feed Objects	5
3. Publishing Feed Menus	5
4. Processing Feed Menus	6
5. IANA Considerations	7
6. Security Considerations	7
7. References	7
7.1. Normative References	7
7.2. Informative References	8
Appendix A. JSON Schema	8
Appendix B. Implementation Status	9
Author's Address	10

1. Introduction

Autodiscovery of Web feeds (in formats such as RSS [RSS] and Atom [ATOM]) using link relations in HTML [DISCOVERY] is widely supported both by consuming software (e.g., feed readers) and Web sites (including in many content management systems).

Deployment of autodiscovery has uncovered significant issues with this approach.

Because feed links are specific to a page, they are only apparent to software that can take advantage of them (for example, to show a 'subscribe' button) when that page is being viewed. This means that sites need to repeat the relevant feeds in the headmatter of all pages on a site to assure that they are viewable, creating overhead both in effort (to organise the pages) and bandwidth (to serve them).

Practically, this means that it is often necessary to hunt around the pages of a site to find all of the relevant feeds, since sites are unwilling to repeat them on every page.

Furthermore, because many content management systems automatically create this metadata, in practice many feeds that are advertised using autodiscovery are not actually functional -- site owners do not realise that the feeds are being advertised and do not check the feeds for quality and relevance.

Finally, there are no existant best practices for naming or assigning metadata to feeds (for example, their format), leading to confusing user experiences. For example, because many sites make both RSS and Atom feeds available, there are often duplicate entries in the autodiscovery metadata.

This specification documents a different mechanism for discovering the feeds relevant to a Web site. By defining a single, well-known location [WELL-KNOWN] that describes a site's feeds, it encourages a more curated, usable directory of the site's feeds for easy discovery and consumption by clients.

Section 2 describes this format; Section 3 recommends practices for publishers that wish to use it, and Section 4 provides guidance to clients for processing it.

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.

2. Feed Menu Documents

A feed menu document is a JSON [JSON] format whose root is an menu object (Section 2.1).

For example, a minimal feed menu document:

```
{
  "feed-menu": "The Astor Theatre",
  "items": [
    {
      "feed-title": "Upcoming Shows",
      "rss": "/shows/upcoming.xml"
    }
  ]
}
```

2.1. Menu Objects

A menu object is a JSON object with a "feed-menu" member. Its potential members are:

- * "feed-menu": A short human readable title for the menu (REQUIRED)
- * "items": An array of menu objects and feed objects.

Menu objects MAY contain other members, to be defined by updates to this specification. Likewise, the items array MAY contain other types of objects, to be defined by updates to this specification.

Menu objects can contain other menu objects to represent a nested structure. For example:

```
{
  "feed-menu": "Analog:Shift",
  "items": [
    {
      "feed-title": "Analog:Shift News",
      "rss": "/news.xml"
    },
    {
      "feed-menu": "Watches for Sale",
      "items": [
        {
          "feed-title": "All Watches",
          "rss": "/collections/watches/all.xml"
        },
        {
          "feed-title": "Vintage Watches",
          "rss": "/collections/watches/vintage.xml"
        },
        {
          "feed-title": "Neo-Vintage Watches",
          "rss": "/collections/watches/neovintage.xml"
        },
        {
          "feed-title": "Contemporary Watches",
          "rss": "/collections/watches/contemporary.xml"
        }
      ]
    }
  ]
}
```

2.2. Feed Objects

A feed object is a JSON object with a "feed-title" member. Its potential members are:

- * "feed-title": A short human readable title for the feed (REQUIRED)
- * "description": A potentially longer human readable description of the feed's purpose or content (OPTIONAL)
- * "rss": A URL to retrieve a RSS representation of the feed (see below)
- * "atom": A URL to retrieve an Atom representation of the feed (see below)

One of "rss" or "atom" is REQUIRED; both MAY be present.

URLs in "rss" and "atom" values MAY be relative, and MUST be resolved relative to the feed document's URL without a path. For example, if a feed document is located at "https://example.net/.well-known/feed-menu.json" and the "rss" member contains "feed.xml", the resulting RSS feed URL would be "https://example.net/feed.xml".

Feed objects MAY contain other members, to be defined by updates to this specification.

For example, a feed menu document with a more elaborate feed object:

```
{
  "feed-menu": "The Astor Theatre",
  "items": [
    {
      "feed-title": "Upcoming Shows",
      "rss": "/shows/upcoming.xml"
    }
  ]
}
```

3. Publishing Feed Menus

Feed menu documents are published at the "feed-menu.json" well-known URI [WELL-KNOWN] for a given Web site. For example:

<https://www.example.com/.well-known/feed-menu.json>

If the Web site supports multiple languages, publishers can use HTTP proactive negotiation to make different languages available to processors. Publishers generally SHOULD NOT mix languages in the same feed menu document, although there may be cases where this is done intentionally.

It is RECOMMENDED that feed-titles be no longer than 50 characters, and that descriptions be no longer than 140 characters. However, processors SHOULD NOT strictly enforce these limits, although they may take measures to deal with long values.

Publishing software (for example, content management systems) SHOULD NOT automatically create feed menu documents, unless they have a high degree of certainty that the publisher is aware and that doing so is intentional.

4. Processing Feed Menus

Processors MUST only request feed menu documents from the "feed-menu.json" well-known URI.

When requesting feed menu documents, processors SHOULD negotiate for content language using proactive negotiation; see Section 12.5.4 of [HTTP].

Processors SHOULD follow redirects when requesting feed menu documents, subject to limits on loop, abuse, and similar error handling.

Processors MUST ignore unrecognised members of menu and feed objects. Likewise, they MUST ignore unrecognised objects in the menu object's items array.

Processors SHOULD make menu object items available to users in the order they appear in the array. For example, a visual interface might make them available as a list. In some cases, however, they may be made available in another ordering (for example, a search interface).

Processors SHOULD NOT display feed objects that do not have a link that they are able to follow -- for example, those using unsupported URI schemes. In some cases, this may be difficult for implementations to conform to; for example, if feed links are dispatched to separate software.

5. IANA Considerations

IANA should register the following Well-Known URI:

- * URI Suffix: feed-menu.json
- * Change Controller: IETF
- * Specification document(s): [this document]

6. Security Considerations

A malicious actor who gains write access to the well-known location can effectively hijack the subscription list for an entire site's audience.

Clients that automatically fetch the feed menu document leak the fact that the user is interested in subscribing to feeds. In some views this might be a feature; it creates evidence that providing feeds would be beneficial to a publisher's audience. However, it is also a privacy risk.

7. References

7.1. Normative References

- [ATOM] Nottingham, M., Ed. and R. Sayre, Ed., "The Atom Syndication Format", RFC 4287, DOI 10.17487/RFC4287, December 2005, <<https://www.rfc-editor.org/rfc/rfc4287>>.
- [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>>.
- [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>>.

[RSS] "RSS 2.0 Specification", March 2009,
<<https://www.rssboard.org/rss-specification>>.

[WELL-KNOWN]
Nottingham, M., "Well-Known Uniform Resource Identifiers
(URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019,
<<https://www.rfc-editor.org/rfc/rfc8615>>.

7.2. Informative References

[DISCOVERY]
WHATWG, "Feed Autodiscovery", December 2006,
<<https://blog.whatwg.org/feed-autodiscovery>>.

Appendix A. JSON Schema

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "Feed Menu Document",
  "description": "Strict validation schema for draft-nottingham-feed-menu.md",
  "type": "object",
  "$ref": "#/$defs/menuObject",
  "$defs": {
    "menuObject": {
      "type": "object",
      "required": [
        "feed-menu",
        "items"
      ],
      "properties": {
        "feed-menu": {
          "type": "string"
        },
        "items": {
          "type": "array",
          "items": {
            "anyOf": [
              {
                "$ref": "#/$defs/menuObject"
              },
              {
                "$ref": "#/$defs/feedObject"
              }
            ]
          }
        }
      }
    }
  },
  "additionalProperties": false
}
```



```
{,
"feedObject": {
  "type": "object",
  "required": [
    "feed-title"
  ],
  "anyOf": [
    {
      "required": [
        "rss"
      ]
    },
    {
      "required": [
        "atom"
      ]
    }
  ],
  "properties": {
    "feed-title": {
      "type": "string"
    },
    "description": {
      "type": "string"
    },
    "rss": {
      "type": "string",
      "format": "uri-reference"
    },
    "atom": {
      "type": "string",
      "format": "uri-reference"
    }
  },
  "additionalProperties": false
},
}
```

Appendix B. Implementation Status

A Web browser extension for presenting feed menus is available for Safari, Firefox, and Chrome here: <https://github.com/mnot/feedmenu-extension>

The following Web site(s) have a demonstration feed menu:

- * <https://www.mnot.net/>

(other implementations can be listed as they appear; please contact me)

Author's Address

Mark Nottingham
Melbourne
Australia
Email: mnot@mnot.net
URI: <https://www.mnot.net/>