

The Atom Notification Protocol
draft-snell-atompub-notification-01

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 1, 2005.

Copyright Notice

Copyright (C) The Internet Society (2005). All Rights Reserved.

Abstract

This memo presents a protocol for posting notifications of new or updated content using a combination of the Atom Syndication Format and HTTP POSTs.

Table of Contents

- [1.](#) Introduction [3](#)
- [1.1](#) Notational Conventions [3](#)
- [1.2](#) Terminology [3](#)
- [2.](#) The Atom Notification Protocol Model [3](#)
- [3.](#) Functional Specification [3](#)
- [3.1](#) NotificationURI [3](#)
- [3.1.1](#) Locating the NotificationURI [4](#)
- [3.1.2](#) Request [4](#)
- [3.1.3](#) Response [5](#)
- [4.](#) Security Considerations [5](#)
- [5.](#) IANA Considerations [6](#)
- [6.](#) References [6](#)
- Author's Address [6](#)
- Intellectual Property and Copyright Statements [7](#)

1. Introduction

The Atom Notification Protocol is an application-level protocol for posting notification of new or updated content using HTTP and the Atom Syndication Format.

1.1 Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [1].

1.2 Terminology

Atom Entry: An Atom Entry is a fragment of a full Atom feed. In this case, the fragment is a single 'entry' element and all its child elements. Each Atom Entry describes a single Web resource, providing metadata and optionally a textual representation of that resource.

Atom Head: atom:head elements are used within the Atom Syndication Format as children of both the atom:feed and atom:entry elements to provide information descriptive of the feed.

NotificationURI: A HTTP URI that is used to receive notifications about new or updated Atom entries.

2. The Atom Notification Protocol Model

The Atom Notification Protocol has been designed to complement the Atom Publishing Protocol by providing the means of sending notifications when Atom-based content is modified in some way.

The Atom Notification Protocol works by POSTing atom:entry or atom:head elements to a NotificationURI using HTTP POST.

As is the case with the Atom Publishing Protocol, this document does not seek to specify the form of the URIs that are used. This document does, however, specify the formats of the entities posted to those URIs.

3. Functional Specification

3.1 NotificationURI

The NotificationURI is used to POST notifications. A notification consists of a single atom:entry or atom:head element. The notification is essentially a one-way operation that implies no operational semantics or action on the part of the receiver.

3.1.1 Locating the NotificationURI

Because of the broad variety of cases in which the Atom Notification Protocol may be used and the lack of any single operational semantic for notifications beyond basic delivery, no single location mechanism for NotificationURI's will be defined. This means that it is up to specific application profiles to determine the best way to make the NotificationURI known within the context of that application.

3.1.2 Request

The request contains a filled-in atom:entry or atom:head element.

A notification request containing an atom:entry is intended to notify the receiving endpoint that a specific entry has been created or updated.

A notification request containing an atom:head is intended to notify the receiving endpoint that a specific feed described by the atom:head has been created or updated.

atom:entry's POSTed to a NotificationURI SHOULD contain a atom:head element that identifies the feed to which the entry belongs.

atom:head elements POSTed to a NotificationURI MUST have a version attribute that identifies the Atom Syndication Format version used. The version attribute is identical to the version attribute defined for the atom:feed element in the Atom Syndication Format.

```
<?xml version="1.0" encoding="utf-8"?>
<head version="draft-ietf-atompub-format-05: do not deploy"
  xmlns="http://purl.org/atom/ns#draft-ietf-atompub-format-05">
  <title>Example Feed</title>
  <link href="http://example.org/" />
  <updated>2003-12-13T18:30:02Z</updated>
  <author>
    <name>John Doe</name>
  </author>
</head>
```

POST is the only method that SHOULD be supported by the NotificationURI. Clients MUST NOT submit requests using any other method to the NotificationURI. If a client submits a request using any other method than POST, The NotificationURI SHOULD respond with a 405 Method Not Allowed response.

3.1.3 Response

The response to a notification POST MUST be an empty message. That is, the message MUST NOT contain any content beyond the HTTP headers. Clients MUST ignore any content that a NotificationURI implementation happens to include.

3.1.3.1 2xx Response Codes

A response code of 202 indicates that the notification was successfully received and accepted. The body of the message SHOULD be empty.

All other 2xx HTTP status codes SHOULD be treated as if they were 202 responses.

3.1.3.2 3xx Response Codes

A response code of 301 indicates that the NotificationURI has permanently changed locations and that the client MUST NOT make any further attempts to send notifications to this location. A new location SHOULD be provided using the Location HTTP header field.

A response code of 302 indicates that the NotificationURI has temporarily changed locations and that the client SHOULD reissue their notification to the new location specified in the Location HTTP header field but that future notifications should continue to be sent to this location.

NotificationURI's SHOULD NOT return 300, 303, 304, 306 or 307 response codes. If a NotificationURI's does return any of these codes, they MUST be ignored.

3.1.3.3 4xx Response Codes

All 4xx response codes are to be interpreted as they are defined in the HTTP specification. 416 and 417 SHOULD NOT be issued by the NotificationURI and MUST be ignored if they are.

3.1.3.4 5xx Response Codes

All 5xx response codes are to be interpreted as they are defined in the HTTP specification.

4. Security Considerations

The decision of whether or not to secure the Atom Notification Protocol will be made on a case-by-case decision. Some notification

endpoints may be restricted to known authenticated users while others will be open for anyone who wishes to post notifications. If a given NotificationURI is restricted, the same authentication mechanism(s) used by the Atom Publishing Protocol SHOULD be used.

One particular challenge that implementors of NotificationURI endpoints will need to be aware of is the potential for denial of service attacks and notification spamming. This document shall not deal with potential solutions to such attacks.

5. IANA Considerations

This document has no actions for IANA.

6 References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [2] Gregorio, J., Ed. and R. Sayre, Ed., "The Atom Publishing Protocol", [draft-ietf-atompub-protocol-02](#) (work in progress), September 2004.
- [3] Nottingham, M., Ed. and R. Sayre, Ed., "The Atom Syndication Format", [draft-ietf-atompub-format-05](#) (work in progress), January 2005.

Author's Address

James M Snell

EMail: james@snellspace.com

URI: <http://www.snellspace.com>

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

The Atom Notification Protocol
[draft-snell-atompub-notification-01](#)

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 1, 2005.

Copyright Notice

Copyright (C) The Internet Society (2005). All Rights Reserved.

Abstract

This memo presents a protocol for posting notifications of new or updated content using a combination of the Atom Syndication Format and HTTP POSTs.

Table of Contents

- [1.](#) Introduction [3](#)
- [1.1](#) Notational Conventions [3](#)
- [1.2](#) Terminology [3](#)
- [2.](#) The Atom Notification Protocol Model [3](#)
- [3.](#) Functional Specification [3](#)
- [3.1](#) NotificationURI [3](#)
- [3.1.1](#) Locating the NotificationURI [4](#)
- [3.1.2](#) Request [4](#)
- [3.1.3](#) Response [5](#)
- [4.](#) Security Considerations [5](#)
- [5.](#) IANA Considerations [6](#)
- [6.](#) References [6](#)
- Author's Address [6](#)
- Intellectual Property and Copyright Statements [7](#)

1. Introduction

The Atom Notification Protocol is an application-level protocol for posting notification of new or updated content using HTTP and the Atom Syndication Format.

1.1 Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [1].

1.2 Terminology

Atom Entry: An Atom Entry is a fragment of a full Atom feed. In this case, the fragment is a single 'entry' element and all its child elements. Each Atom Entry describes a single Web resource, providing metadata and optionally a textual representation of that resource.

Atom Head: atom:head elements are used within the Atom Syndication Format as children of both the atom:feed and atom:entry elements to provide information descriptive of the feed.

NotificationURI: A HTTP URI that is used to receive notifications about new or updated Atom entries.

2. The Atom Notification Protocol Model

The Atom Notification Protocol has been designed to complement the Atom Publishing Protocol by providing the means of sending notifications when Atom-based content is modified in some way.

The Atom Notification Protocol works by POSTing atom:entry or atom:head elements to a NotificationURI using HTTP POST.

As is the case with the Atom Publishing Protocol, this document does not seek to specify the form of the URIs that are used. This document does, however, specify the formats of the entities posted to those URIs.

3. Functional Specification

3.1 NotificationURI

The NotificationURI is used to POST notifications. A notification consists of a single atom:entry or atom:head element. The notification is essentially a one-way operation that implies no operational semantics or action on the part of the receiver.

3.1.1 Locating the NotificationURI

Because of the broad variety of cases in which the Atom Notification Protocol may be used and the lack of any single operational semantic for notifications beyond basic delivery, no single location mechanism for NotificationURI's will be defined. This means that it is up to specific application profiles to determine the best way to make the NotificationURI known within the context of that application.

3.1.2 Request

The request contains a filled-in atom:entry or atom:head element.

A notification request containing an atom:entry is intended to notify the receiving endpoint that a specific entry has been created or updated.

A notification request containing an atom:head is intended to notify the receiving endpoint that a specific feed described by the atom:head has been created or updated.

atom:entry's POSTed to a NotificationURI SHOULD contain a atom:head element that identifies the feed to which the entry belongs.

atom:head elements POSTed to a NotificationURI MUST have a version attribute that identifies the Atom Syndication Format version used. The version attribute is identical to the version attribute defined for the atom:feed element in the Atom Syndication Format.

```
<?xml version="1.0" encoding="utf-8"?>
<head version="draft-ietf-atompub-format-05: do not deploy"
  xmlns="http://purl.org/atom/ns#draft-ietf-atompub-format-05">
  <title>Example Feed</title>
  <link href="http://example.org/" />
  <updated>2003-12-13T18:30:02Z</updated>
  <author>
    <name>John Doe</name>
  </author>
</head>
```

POST is the only method that SHOULD be supported by the NotificationURI. Clients MUST NOT submit requests using any other method to the NotificationURI. If a client submits a request using any other method than POST, The NotificationURI SHOULD respond with a 405 Method Not Allowed response.

3.1.3 Response

The response to a notification POST MUST be an empty message. That is, the message MUST NOT contain any content beyond the HTTP headers. Clients MUST ignore any content that a NotificationURI implementation happens to include.

3.1.3.1 2xx Response Codes

A response code of 202 indicates that the notification was successfully received and accepted. The body of the message SHOULD be empty.

All other 2xx HTTP status codes SHOULD be treated as if they were 202 responses.

3.1.3.2 3xx Response Codes

A response code of 301 indicates that the NotificationURI has permanently changed locations and that the client MUST NOT make any further attempts to send notifications to this location. A new location SHOULD be provided using the Location HTTP header field.

A response code of 302 indicates that the NotificationURI has temporarily changed locations and that the client SHOULD reissue their notification to the new location specified in the Location HTTP header field but that future notifications should continue to be sent to this location.

NotificationURI's SHOULD NOT return 300, 303, 304, 306 or 307 response codes. If a NotificationURI's does return any of these codes, they MUST be ignored.

3.1.3.3 4xx Response Codes

All 4xx response codes are to be interpreted as they are defined in the HTTP specification. 416 and 417 SHOULD NOT be issued by the NotificationURI and MUST be ignored if they are.

3.1.3.4 5xx Response Codes

All 5xx response codes are to be interpreted as they are defined in the HTTP specification.

4. Security Considerations

The decision of whether or not to secure the Atom Notification Protocol will be made on a case-by-case decision. Some notification

endpoints may be restricted to known authenticated users while others will be open for anyone who wishes to post notifications. If a given NotificationURI is restricted, the same authentication mechanism(s) used by the Atom Publishing Protocol SHOULD be used.

One particular challenge that implementors of NotificationURI endpoints will need to be aware of is the potential for denial of service attacks and notification spamming. This document shall not deal with potential solutions to such attacks.

5. IANA Considerations

This document has no actions for IANA.

6 References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [2] Gregorio, J., Ed. and R. Sayre, Ed., "The Atom Publishing Protocol", [draft-ietf-atompub-protocol-02](#) (work in progress), September 2004.
- [3] Nottingham, M., Ed. and R. Sayre, Ed., "The Atom Syndication Format", [draft-ietf-atompub-format-05](#) (work in progress), January 2005.

Author's Address

James M Snell

EMail: james@snellspace.com

URI: <http://www.snellspace.com>

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.