Internet Engineering Task Force (IETF)
Request for Comments: 5995
Category: Standards Track
ISSN: 2070-1721

J. Reschke
greenbytes

September 2010

Using POST to Add Members to Web Distributed Authoring and Versioning (WebDAV) Collections

Abstract

The Hypertext Transfer Protocol (HTTP) Extensions for the Web Distributed Authoring and Versioning (WebDAV) do not define the behavior for the "POST" method when applied to collections, as the base specification (HTTP) leaves implementers lots of freedom for the semantics of "POST".

This has led to a situation where many WebDAV servers do not implement POST for collections at all, although it is well suited to be used for the purpose of adding new members to a collection, where the server remains in control of the newly assigned URL. In fact, the Atom Publishing Protocol (AtomPub) uses POST exactly for that purpose. On the other hand, WebDAV-based protocols, such as the Calendaring Extensions to WebDAV (CalDAV), frequently require clients to pick a unique URL, although the server could easily perform that task.

This specification defines a discovery mechanism through which servers can advertise support for POST requests with the aforementioned "add collection member" semantics.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5995.

Table of Contents

1. Introduction

The Hypertext Transfer Protocol (HTTP) Extensions for the Web Distributed Authoring and Versioning (WebDAV) ([RFC4918], Section 9.5) do not define the behavior for the "POST" method when applied to collections, as the base specification (HTTP) leaves implementers lots of freedom for the semantics of "POST":

> 9.5 POST for Collections
>
> Since by definition the actual function performed by POST is determined by the server and often depends on the particular resource, the behavior of POST when applied to collections cannot be meaningfully modified because it is largely undefined. Thus, the semantics of POST are unmodified when applied to a collection.

This has led to a situation where many WebDAV servers do not implement POST for collections at all, although it is well suited to be used for the purpose of adding new members to a collection, where the server remains in control of the newly assigned URL. In fact, the Atom Publishing Protocol (AtomPub) uses POST exactly for that purpose ([RFC5023], Section 9.2):

> 9.2 Creating Resources with POST
>
> To add members to a Collection, clients send POST requests to the URI of the Collection.

On the other hand, WebDAV-based protocols, such as Calendaring Extensions to WebDAV (CalDAV), frequently require clients to pick a unique URL, although the server could easily perform that task ([RFC4791], Section 5.3.2):

> 5.3.2 Creating Calendar Object Resources
>
> ...
>
> When servers create new resources, it's not hard for the server to choose an unmapped URI. It's slightly tougher for clients, because a client might not want to examine all resources in the collection and might not want to lock the entire collection to ensure that a new resource isn't created with a name collision. (...)

Letting the server choose the member URI not only is a simplification for certain types of clients, but can also reduce the complexity of the server (in that it doesn't need to persist an additional client-supplied identifier where it already has an internal one like a Universally Unique Identifier (UUID) or a primary key).

> Note: The vCard Extensions to WebDAV (CardDAV) ([CARDDAV]) suffer from the same issue, and may be able to take advantage of this specification.

This specification defines a discovery mechanism through which servers can advertise support for POST requests with the aforementioned "add collection member" semantics.

This specification deliberately only addresses the use case of creating new non-collection resources. It was not a goal for this specification to supply the same functionality for creating collection resources (MKCOL) or for other operations that require the client to specify a new URL (LOCK, MOVE, or COPY).

> Note: The author previously proposed a new HTTP method for exactly this purpose ([ADDMEMBER]), but quite a few reviewers pointed out that this would duplicate the original semantics of POST. Thus, this proposal, which avoids adding a new HTTP method, is made.

2. Terminology

The terminology used here follows that in the WebDAV specification ([RFC4918]).

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 [RFC2119].

This document uses XML DTD fragments ([XML]) as a purely notational convention. In particular:

- Element ordering is irrelevant.
- Extension elements/attributes (elements/attributes not already defined as valid child elements) may be added anywhere, except when explicitly stated otherwise.

  > Note: This specification defines new properties and precondition names in the "DAV:" namespace, which the WebDAV specification reserves for use by the IETF ([RFC4918], Section 21.1). However, there was rough consensus in the WebDAV community that the specification is of general applicability to other WebDAV-related standards efforts, and thus deserves inclusion into the base namespace.

3. Protocol Extension

Due to the reasons stated in Section 1, clients cannot rely on a specific server behavior when POST is applied to a collection. This problem is addressed by this specification by allowing servers to advertise a URI that has the desired "add member" semantics.

Servers that already use POST for a different purpose can just expose a separate URI. Other servers can just advertise the collection's own URI, thus avoiding minting another URI for a limited purpose.

3.1. Definition of "Add-Member" URI

The "Add-Member" URI of a WebDAV collection is a URI that will accept HTTP POST requests, and will interpret these as requests to store the enclosed entity as a new internal member of the collection (see Section 3 of [RFC4918] for the definition of "internal member"). It MUST identify a resource on the same server as the WebDAV collection (the host and port components ([RFC2616], Section 3.2.2) of the URIs must match).

If there are preconditions related to creating a resource in the collection using a PUT request, then those same preconditions apply to the new POST request behavior, and the same HTTP response body will be returned on failure.

The URI of the newly created resource is returned in the HTTP Location response header field ([RFC2616], Section 14.30).

> Note: The fact that a server advertises an "Add-Member" URI does not imply any special semantics of the collection itself. For instance, member URIs assigned by the server are not necessarily unique over time (a member URI may be assigned again to a new resource when it previously was removed).
>
> Note: The "Add-Member" URI can be identical to the collection's URI (in which case the server just advertises the fact that POST to the WebDAV collection's URI is supported as defined within this specification). But it can also be different from it, in which case it doesn't need to have any relation to the collection's URI.
>
> Given a collection URI of
>
> /docs/collection/
>
> any of the URIs below might occur as "Add-Member" URIs:
>
> /docs/collection/
> /docs/collection/;post
> /docs/collection;post/
> /docs/collection/&post
> /post-service?path=/collection/

The remainder of the document uses the same format just for reasons of consistency; any other HTTP URI on the same server would do as well.

3.2. Discovery

3.2.1. DAV:add-member Property (Protected)

DAV:add-member is a protected property (see [RFC4918], Section 15) defined on WebDAV collections, and contains the "Add-Member" URI for that collection (embedded inside a DAV:href element).

```
<!ELEMENT add-member (href)>
<!-- href: defined in [RFC4918], Section 14.7 -->
```

A PROPFIND/allprop request SHOULD NOT return this property (see [RFC4918], Section 9.1). Servers MUST implement the DAV:supported-live-property-set property defined in Section 3.1.4 of [RFC3253], and report the property DAV:add-member as a supported live property.

3.2.2. Example

```
>>Request

PROPFIND /collection/ HTTP/1.1
Host: example.com
Content-Type: application/xml; charset="utf-8"
Content-Length: 118

<?xml version="1.0" encoding="utf-8" ?>
<propfind xmlns="DAV:">
  <prop>
    <add-member/>
  </prop>
</propfind>
```

```
>>Response

HTTP/1.1 207 Multi-Status
Content-Type: application/xml; charset="utf-8"
Content-Length: 340

<?xml version="1.0" encoding="utf-8" ?>
<multistatus xmlns="DAV:">
  <response>
    <href>/collection/</href>
    <propstat>
      <prop>
        <add-member>
          <href>/collection;add-member/</href>
        </add-member>
      </prop>
      <status>HTTP/1.1 200 OK</status>
    </propstat>
  </response>
</multistatus>
```

In this case, the server has minted a separate URI for the purpose of adding new content.

3.3. Relation to AtomPub's "Slug" Header Field

In the AtomPub protocol, clients can use the entity header field "Slug" to suggest parts of the URI to be created (see [RFC5023], Section 9.7). Note that servers are free to ignore this suggestion, or to use whatever algorithm makes sense to generate the new URI.

The same applies to the extension defined here: clients can use the "Slug" header field, as by definition it is a generic HTTP header field. Servers should process it exactly in the way defined by AtomPub.

3.4. Example Operation

```
>>Request

POST /collection;add-member/ HTTP/1.1
Host: example.com
Content-Type: text/plain
Slug: Sample Title
Content-Length: 12

Sample text.
```

```
>>Response

HTTP/1.1 201 Created
Location: http://example.com/collection/sample%20title
```

4. Additional Semantics for Existing Methods

One important use case for this specification is collections that act as WebDAV collections for the purpose of read access (PROPFIND Depth 1/Infinity), but which only support internal member URIs assigned by the server. These collections will not allow a client to create a new member using methods like PUT, MKCOL, LOCK, COPY, or MOVE. Therefore, this specification defines a new precondition name ([RFC4918], Section 16) that can be used to provide the client with additional information regarding exactly why the request failed.

> Note: Although the precondition defined below can be used for methods other than PUT, the "Add-Member" mechanism defined by this specification deliberately is restricted to PUT.

4.1. Additional Preconditions

(DAV:allow-client-defined-URI): the server allows clients to specify the last path segment for newly created resources.

The precondition element MAY contain an add-member-uri XML element specifying the "Add-Member" URI associated with the collection, on which the creation of a new child resource was attempted:

```
<!ELEMENT allow-client-defined-uri (add-member?)>
```

4.2. Example: Failed PUT Request

In this example, the client tries to use PUT to create a new internal member of /collection/.

```
>>Request

PUT /collection/new.txt HTTP/1.1
Host: example.com
Content-Type: text/plain
Content-Length: 12

Sample text.
```

```
>>Response

HTTP/1.1 405 Method Not Allowed
Allow: GET, HEAD, TRACE, PROPFIND, COPY, MOVE
Content-Type: application/xml; charset=UTF-8
Content-Length: 172

<error xmlns="DAV:">
  <allow-client-defined-uri>
    <add-member>
      <href>/collection;add-member/</href>
    </add-member>
  </allow-client-defined-uri>
</error>
```

The request fails with a 405 (Method Not Allowed) status, but also provides the reason, and a pointer to the "Add-Member" URI in the response body.

5. Relationship to WebDAV Access Control Protocol

The WebDAV Access Control Protocol specification requires the DAV:bind privilege to be granted on a collection for the client to be able to add new collection members ([RFC3744], Section 3.9). Consistent with that, a server MUST reject a POST request to the Add-Member URI of a collection, unless the principal executing the request is granted DAV:bind privilege on the associated WebDAV collection resource.

6. Internationalization Considerations

This document does not introduce any new internationalization considerations beyond those discussed in Section 19 of [RFC4918].

7. Security Considerations

Security considerations applicable to HTTP [RFC2616], WebDAV [RFC4918], and XML [XML] apply for this specification as well, namely, Section 20 of [RFC4918] and Section 7 of [RFC3470].

Furthermore, servers should be aware that deriving the member path from the data being stored in the resource could potentially expose confidential information. This could even be the case when only a hash code of the content is used.

In addition, on servers that do not support this specification, a malevolent user could set the DAV:add-member URI as a custom property, tricking other users to post content to an entirely different URI. Clients can protect themselves against this scenario by

- not following DAV:add-member URIs to different servers, and/or
- verifying that the DAV:add-member property is indeed a live property (this can be achieved by testing the DAV:supported-live-property-set property, or by checking whether DAV:add-member is returned upon PROPFIND/allprop).

8. Acknowledgements

This document has benefited from thoughtful discussion by Cyrus Daboo and Bernard Desruisseaux.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3253] Clemm, G., Amsden, J., Ellison, T., Kaler, C., and J. Whitehead, "Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning)", RFC 3253, March 2002.

[RFC3744] Clemm, G., Reschke, J., Sedlar, E., and J. Whitehead, "Web Distributed Authoring and Versioning (WebDAV) Access Control Protocol", RFC 3744, May 2004.

[RFC4918] Dusseault, L., Ed., "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)", RFC 4918, June 2007.

[XML] Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C REC-xml-20081126, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126/>.

9.2. Informative References

[ADDMEMBER] Reschke, J., "The HTTP ADDMEMBER Method", Work in Progress, February 2005.

[CARDDAV] Daboo, C., "vCard Extensions to WebDAV (CardDAV)", Work in Progress, November 2009.

[RFC3470] Hollenbeck, S., Rose, M., and L. Masinter, "Guidelines for the Use of Extensible Markup Language (XML) within IETF Protocols", BCP 70, RFC 3470, January 2003.

[RFC4791] Daboo, C., Desruisseaux, B., and L. Dusseault, "Calendaring Extensions to WebDAV (CalDAV)", RFC 4791, March 2007.

[RFC5023] Gregorio, J., Ed. and B. de hOra, Ed., "The Atom Publishing Protocol", RFC 5023, October 2007.

Index

Author's Address

Julian F. Reschke
greenbytes GmbH
Hafenweg 16
Muenster, NW 48155
Germany

Phone: +49 251 2807760
EMail: julian.reschke@greenbytes.de
URI: http://greenbytes.de/tech/webdav/