Network Working Group
Internet-Draft
Expires: August 17, 2005

J. Reschke
greenbytes

February 13, 2005

The HTTP ADDMEMBER Method
draft-reschke-http-addmember-00

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of Section 3 of RFC 3667. By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she 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 17, 2005.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

Frequently, servers may want to allow resource creation through HTTP, but are not able to support HTTP's PUT method for creating new resources, as resource names are completely controlled by the server. This document proposes a new HTTP method called "ADDMEMBER" with semantics similar to those of PUT, except for the fact that the server chooses the URI for the newly created resource.

Editorial Note

Distribution of this document is unlimited. Please send comments to the Hypertext Transfer Protocol (HTTP) mailing list at ietf-http-wg@w3.org [1], which may be joined by sending a message with subject "subscribe" to ietf-http-wg-request@w3.org [2].

Discussions of the HTTP working group are archived at <http://lists.w3.org/Archives/Public/ietf-http-wg/>.

Table of Contents

1. Introduction

Frequently, servers may want to allow resource creation through HTTP, but are not able to support HTTP's PUT method for creating new resources, as resource names are completely controlled by the server (see [RFC2616], Section 9.6). This document proposes a new HTTP method called "ADDMEMBER" with semantics similar to those of PUT, except for the fact that the server chooses the URI for the newly created resource.

Some alternative approaches are summarized in Appendix A for discussion.

2. 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 [RFC2119].

All terminology not defined explicitly in this document is inherited from [RFC2616].

3. ADDMEMBER method

The ADDMEMBER method requests that the enclosed entity be stored as a new resource under a URI selected by the server based on the Request-URI referring to a container resource. [[anchor4: Do we need to require a specific containment model here, such as WebDAV's collections? --reschke]]

If a new resource is created, the origin server MUST inform the user agent via the 201 (Created) response, including a "Location" response header containing the URI of the newly created resource. If the resource could not be created, an appropriate error response SHOULD be given that reflects the nature of the problem. The recipient of the entity MUST NOT ignore any Content-* (e.g. Content-Range) headers that it does not understand or implement and MUST return a 501 (Not Implemented) response in such cases.

Responses to this method are not cacheable.

The fundamental difference between the ADDMEMBER and PUT requests is reflected in the different meaning of the Request-URI. The URI in an ADDMEMBER request identifies the resource that will handle the enclosed entity by storing it as a new resource with a server-selected URI. In contrast, the URI in a PUT request identifies the entity enclosed with the request -- the user agent knows what URI is intended and the server MUST NOT attempt to apply the request to some

other resource.

ADDMEMBER requests MUST obey the message transmission requirements set out in Section 8.2 of [RFC2616].

Entity-headers in the ADDMEMBER request SHOULD be handled the same way as defined for PUT.

This method is neither safe nor idempotent (see [RFC2616], Section 9).

3.1 Example: ADDMEMBER

>> Request:

```
ADDMEMBER /CollY HTTP/1.1
Host: www.example.com
Content-Type: application/xml

<foobar/>
```

>> Response:

```
HTTP/1.1 201 Created
Location: http://www.example.com/CollY/3253623
```

4. Feature Discovery

Clients can detect server support for the ADDMEMBER method by inspecting the "Allow" response header returned for an OPTIONS request on the Request-URI. Note that a server may support ADDMEMBER only on a subset of the URIs it is handling.

5. Security Considerations

The same security considerations as those for HTTP PUT apply.

6. Acknowledgements

[[anchor7: TBD. --reschke]]

7. References

7.1 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.

7.2 Informative References

[RFC2774] Nielsen, H., Leach, P., and S. Lawrence, "An HTTP Extension Framework", RFC 2774, February 2000.

URIs

[1] <mailto:ietf-http-wg@w3.org>

[2] <mailto:ietf-http-wg-request@w3.org?subject=subscribe>

Author's Address

Julian F. Reschke
greenbytes GmbH
Salzmannstrasse 152
Muenster, NW 48159
Germany

Phone: +49 251 2807760
Fax: +49 251 2807761
Email: julian.reschke@greenbytes.de
URI: http://greenbytes.de/tech/webdav/

Appendix A. Dicussion of alternative approaches

This section tries to summarize alternative approaches.

A.1 POST

POST is a very generic method and therefore can be used to achieve the same result. However, clients that rely on the very specific processing defined for ADDMEMBER would need a reliable way to discover how the server is processing POST requests, requiring a new discovery mechanism.

A.2 Implicit PUT extensions

Several communities are discussing to simply use PUT in these situations. The server would allocate a new URI and send a "Location" response header with the new URI, rather than storing the entity at the Request-URI. This seems to be contrary to the stated HTTP semantics for PUT, but would allow existing clients to make use

of this functionality (although it's not clear how well they would handle the "URI change upon creation" scenario.

Example:

```
>> Request:

PUT /CollY/something HTTP/1.1
Host: www.example.com
If-None-Match: *
Content-Type: application/xml

<foobar/>

>> Response:

HTTP/1.1 201 Created
Location: http://www.example.com/CollY/3253623
```

A.3 Explicit extensions based on RFC2774

The extension mechanism defined in [RFC2774] could be used to extend either POST or PUT with the desired semantics.

Example:

```
>> Request:

M-POST /CollY HTTP/1.1
Host: www.example.com
Man: "urn:ietf:id:draft-reschke-http-addmember-00"; ns=00
00-store-enclosed-entity:
Content-Type: application/xml

<foobar/>

>> Response:

HTTP/1.1 201 Created
Location: http://www.example.com/CollY/3253623
```

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.