Abstract

This specification extends the Web Distributed Authoring and Versioning (WebDAV) MKCOL method to allow collections of arbitrary
resource type to be created and to allow properties to be set at the same time.

Table of Contents

1. Introduction ........................................... 3
2. Conventions Used in This Document ..................... 3
3. WebDAV extended MKCOL .................................. 3
   3.1. Extended MKCOL Support .............................. 4
       3.1.1. Example: Using OPTIONS for the Discovery of
               Support for Extended MKCOL ..................... 4
   3.2. Status Codes ....................................... 5
   3.3. Additional Precondition for Extended MKCOL .......... 5
   3.4. Example: Successful Extended MKCOL Request .......... 5
3. Replacing Existing MKxxx Methods ........................ 6
   4.1. MKCALENDAR Replacement ............................. 7
       4.1.1. Example: Replacing MKCALENDAR with MKCOL ....... 7
   5. XML Element Definitions ................................ 9
      5.1. mkcol XML Element .................................. 9
      5.2. mkcol-response XML Element ........................ 9
6. Security Considerations ............................... 10
7. IANA Considerations .................................... 10
8. Acknowledgments ....................................... 10
9. Normative References ................................... 10
Appendix A. Change History (to be removed prior to publication as an RFC) ............... 11
1. Introduction

WebDAV [RFC4918] defines the HTTP [RFC2616] method MKCOL. This method is used to create WebDAV collections on the server. However, several WebDAV-based specifications (e.g., CalDAV [RFC4791]) define "special" collections - ones which are identified by additional values in the DAV:resourcetype property assigned to the collection resource, or through other means. These "special" collections are created by new methods (e.g., MKCALENDAR). The addition of a new MKxxxx method for each new "special" collection adds to server complexity and is detrimental to overall reliability due to the need to make sure intermediaries are aware of these methods.

This specification proposes an extension to the WebDAV MKCOL method that adds a request body allowing a client to specify WebDAV properties to be set on the newly created collection or resource. In particular, the DAV:resourcetype property can be used to create a "special" collection, or other properties used to create a "special" resource. This avoids the need to invent new MKxxxx methods.

2. Conventions Used in This Document

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].

Definitions of XML elements in this document use XML element type declarations (as found in XML Document Type Declarations), described in Section 3.2 of [W3C.REC-xml-20060816].

When XML element types in the namespace "DAV:" are referenced in this document outside of the context of an XML fragment, the string "DAV:" will be prefixed to the element type names.

Processing of XML by clients and servers MUST follow the rules described in Appendix A of [RFC4918].

3. WebDAV extended MKCOL

The WebDAV MKCOL request is extended to allow the inclusion of a request body. The request body is an XML document containing a single DAV:mkcol XML element as the root element. The Content-Type request header MUST be set appropriately for an XML body (e.g., set to "text/xml" or "application/xml"). XML-typed bodies for an MKCOL request that do not have DAV:mkcol as the root element are reserved for future usage.

One or more DAV:set XML elements MAY be included in the DAV:mkcol XML body.
element to allow setting properties on the collection as it is created. In particular, to create a collection of a particular type, the DAV:resourcetype XML element MUST be included in a DAV:set XML element and MUST specify the expected resource type elements for the new resource, that MUST include the DAV:collection element that needs to be present for any WebDAV collection.

As per the PROPPATCH method ([RFC4918], Section 9.2), servers MUST process any DAV:set instructions in document order (an exception to the normal rule that ordering is irrelevant). Instructions MUST either all be executed or none executed. Thus, if any error occurs during processing, all executed instructions MUST be undone and a proper error result returned. Failure to set a property value on the collection MUST result in a failure of the overall MKCOL request.

If a server attempts to make any of the property changes in an extended MKCOL request (i.e., the request is not rejected for high-level errors before processing the body), the response MUST be an XML document containing a single DAV:mkcol-response XML element, which MUST contain DAV:propstat XML elements with the status of each property.

In all other respects the behavior of the extended MKCOL request follows that of the standard MKCOL request.

3.1. Extended MKCOL Support

A server supporting the features described in this document, MUST include "extended-mkcol" as a field in the DAV response header from an OPTIONS request on any URI that supports use of the extended MKCOL method.

3.1.1. Example: Using OPTIONS for the Discovery of Support for Extended MKCOL

>> Request <<

OPTIONS /addressbooks/users/ HTTP/1.1
Host: addressbook.example.com
3.2. Status Codes

As per Section 9.3.1 of [RFC4918].

3.3. Additional Precondition for Extended MKCOL

WebDAV ([RFC4918], Section 16) defines preconditions and postconditions for request behavior. This specification adds the following precondition for the extended MKCOL request.

Name: valid-resource-type

Namespace: DAV:

Use with: Typically 403 (Forbidden)

Purpose: (precondition) -- The server MUST support the specified resource-type value for the specified collection.

3.4. Example: Successful Extended MKCOL Request

This example shows how the extended MKCOL request is used to create a collection of a fictitious type "special-resource".
>> Request <<

MKCOL /home/special/ HTTP/1.1
Host: special.example.com
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:mkcol xmlns:D="DAV:"
xmlns:E="http://example.com/ns/">
  <D:set>
    <D:prop>
      <D:resourcetype>
        <D:collection/>
        <E:special-resource/>
      </D:resourcetype>
      <D:displayname>Special Resource</D:displayname>
    </D:prop>
  </D:set>
</D:mkcol>

>> Response <<

HTTP/1.1 201 Created
Cache-Control: no-cache
Date: Sat, 11 Nov 2006 09:32:12 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:mkcol-response xmlns:D="DAV:">
  <D:propstat>
    <D:prop>
      <D:resourcetype/>
      <D:displayname/>
    </D:prop>
  </D:propstat>
  <D:status>HTTP/1.1 200 OK</D:status>
</D:mkcol-response>

4. Replacing Existing MKxxx Methods

One of the goals of this extension is to eliminate the need for other extensions to define their own variant of MKCOL to create the special collections they need. This extension can be used to replace existing MKxxx methods in other extensions as detailed below. If a server supports this extension and the other extension listed, then the server MUST support use of the extended MKCOL method to achieve
the same result as the MKxxx method of the other extension.

4.1. MKCALENDAR Replacement

CalDAV defines the MKCALENDAR method to create a calendar collection as well as set properties during creation ([RFC4791], Section 5.3.1).

The extended MKCOL method can be used instead by specifying both DAV: collection and CALDAV:calendar-collection XML elements in the DAV: resourcetype property, set during the extended MKCOL request.

4.1.1. Example: Replacing MKCALENDAR with MKCOL

The first example below shows an MKCALENDAR request containing a CALDAV:mkcalendar XML element in the request body, and returning a CALDAV:mkcalendar-response XML element in the response body. The second example shows the equivalent extended MKCOL request with the same request and response XML elements.

>> MKCALENDAR Request <<

MKCALENDAR /home/lisa/calendars/events/ HTTP/1.1
Host: calendar.example.com
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:mkcalendar xmlns:D="DAV:"
 xmlns:C="urn:ietf:params:xml:ns:caldav">
 <D:set>
  <D:prop>
   <D:displayname>Lisa's Events</D:displayname>
  </D:prop>
 </D:set>
</C:mkcalendar>
>> MKCALENDAR Response <<

HTTP/1.1 201 Created
Cache-Control: no-cache
Date: Sat, 11 Nov 2006 09:32:12 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:mkcalendar-response xmlns:D="DAV:"
    xmlns:C="urn:ietf:params:xml:ns:caldav">
    <D:propstat>
        <D:prop>
            <D:displayname/>
        </D:prop>
        <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
</C:mkcalendar-response>

>> MKCOL Request <<

MKCOL /home/cyrus/calendars/events/ HTTP/1.1
Host: calendar.example.com
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:mkcol xmlns:D="DAV:"
    xmlns:C="urn:ietf:params:xml:ns:caldav">
    <D:set>
        <D:prop>
            <D:resourcetype>
                <D:collection/>
                <C:calendar/>
            </D:resourcetype>
            <D:displayname>Cyrus’ Events</D:displayname>
        </D:prop>
    </D:set>
</D:mkcol>
>> MKCOL Response <<

HTTP/1.1 201 Created
Cache-Control: no-cache
Date: Sat, 11 Nov 2006 09:32:12 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:mkcol-response xmlns:D="DAV:"
xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:propstat>
    <D:prop>
      <D:resourcetype/>
      <D:displayname/>
    </D:prop>
    <D:status>HTTP/1.1 200 OK</D:status>
  </D:propstat>
</D:mkcol-response>

5. XML Element Definitions

5.1. mkcol XML Element

Name:  mkcol

Namespace:  DAV:

Purpose:  Used in a request to specify properties to be set in an extended MKCOL request, as well as any additional information needed when creating the resource.

Description:  This XML element is a container for the information required to modify the properties on a collection resource as it is created in an extended MKCOL request.

Definition:

  <!ELEMENT mkcol (set+)>
Purpose: Used in a response to indicate the status of properties that were set or failed to be set during an extended MKCOL request.

Description: This XML element is a container for the information returned about a resource that has been created in an extended MKCOL request.

Definition:

<!ELEMENT mkcol-response (propstat+, ANY)>

6. Security Considerations

This extension does not introduce any new security concerns beyond those already described in HTTP and WebDAV.

7. IANA Considerations

This document does not require any actions on the part of IANA.

8. Acknowledgments

Several people suggested this approach, including Julian Reschke and Bernard Desruisseaux. Thanks also to Mike Douglass.

9. Normative References


[W3C.REC-xml-20060816] Yergeau, F., Paoli, J., Bray, T., Sperberg-McQueen, C., and E. Maler, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", World
Appendix A.  Change History (to be removed prior to publication as an RFC)

Changes from -02:

1. Boiler plate update.

Changes from -01:

1. Minor formatting/wording changes proposed by Julian Reschke were applied.
2. Removed reference to DeltaV entirely as the spec no longer replaces the MKxxx DeltaV defines.
3. Added Namespace definition to precondition.
5. Added statement that DAV:collection must be present in DAV:resourcetype in the request.
6. Added statement on use of DTD fragments.
7. Added statement about setting proper Content-Type for the MKCOL body.
8. Added statement that MKCOL bodies using a different root element are reserved for future extensions.

Changes from -00:

1. Fixed an example.

Changes from draft-daboo-webdav-mkcol-00:

1. Removed MKACTIVITY and MKWORKSPACE replacement behavior.
2. Added valid-resourcetype precondition.
Author’s Address

Cyrus Daboo
Apple Inc.
1 Infinite Loop
Cupertino, CA  95014
USA

EMail: cyrus@daboo.name
URI:  http://www.apple.com/