

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: March 15, 2009

J. Gregorio, Ed.  
Google  
September 11, 2008

**AtomPub Multipart Media Resource Creation  
draft-gregorio-atompub-multipart-04**

Status of this Memo

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 becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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 March 15, 2009.

Abstract

This specification defines how an Atom Publishing Protocol collection may process multipart/related requests and also defines how a service announces that it accepts multipart/related entities.

Editorial Note

To provide feedback on this Internet-Draft, join the Atom Protocol mailing list (<http://www.imc.org/atom-protocol/>) [1].

Table of Contents

- [1.](#) Introduction . . . . . [3](#)
- [1.1.](#) Notational Conventions . . . . . [3](#)
- [1.2.](#) Design Considerations . . . . . [3](#)
- [1.3.](#) Applicability . . . . . [3](#)
- [2.](#) Terminology . . . . . [3](#)
- [3.](#) Multipart Representations . . . . . [4](#)
- [4.](#) Server Processing . . . . . [4](#)
- [5.](#) Service Document Extension . . . . . [5](#)
- [6.](#) Examples . . . . . [5](#)
- [7.](#) Security Considerations . . . . . [9](#)
- [8.](#) IANA Considerations . . . . . [9](#)
- [9.](#) Normative References . . . . . [10](#)
- Author's Address . . . . . [10](#)
- Intellectual Property and Copyright Statements . . . . . [11](#)

## **1. Introduction**

The Atom Publishing Protocol [[RFC5023](#)] defines Media Collections and how to create a Media Resource by POSTing the media to the Media Collection. [RFC 5023](#) does not define handling multipart/related [[RFC2387](#)] representations nor does it specify how the acceptance of such representations should be advertised in the Service Document. This specification covers both the processing and the Service Document aspects of handling multipart/related content.

### **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 [[RFC2119](#)].

### **1.2. Design Considerations**

The primary objective of multipart/related POSTs is to reduce round-trips for creating Media Resources. There are three round trips in the typical Media Resource creation scenario; POST of the media, GET of the Media Link Entry, and subsequent PUT of the updated Media Link Entry. This specification reduces that to just a single round-trip by allowing the client to package up the media and the associated Media Link Entry into a single multipart/related representation which is POSTed to the Media Collection.

The design of the handling of multipart/related representations is aimed at backward compatibility, that is for non-multipart/related aware clients to fully function. A second aim is to retain and utilize the expressiveness of the current app:accept element in the Service Document. The last aim is to ease the burden on clients by allowing the multipart representation to be constructed in an order that is convenient for the client.

### **1.3. Applicability**

The applicability of multipart/related representations to AtomPub Collections is restricted to the creation of new entries in Media collections. It does not specify the creation or use of a resource that supports a GET to return the multipart/related representation nor does it specify the creation or use of a resource that supports a PUT of a multipart/related representation.

## **2. Terminology**

The terms Collection, Media Resource, Media Link Entry, Foreign

Markup, and Service Document are used as defined in [[RFC5023](#)]. The term Object Root is used as defined in [[RFC2387](#)].

### **3. Multipart Representations**

This section covers the constraints on a multipart/related representation sent to a Media Collection. [Section 5](#) covers how a client discovers that a Media Collection accepts multipart/related representations.

There may be other specifications that define uses for multipart representations in the AtomPub context; this specification only describes one particular representation and how to use it in the media-resource creation process. Follow-on multipart/related specifications will have to define a method by which a server can differentiate which specification is in force, which is beyond the scope of this document.

A multipart/related POST to a Media Collection MUST be a valid multipart/related representation as defined by [[RFC2387](#)] and MUST contain two body parts. One, referred to as the Entry Part, MUST be an Atom Entry with a media type of 'application/atom+xml' or 'application/atom+xml;type=entry'. The other, referred to as the Media Part, MUST be of a media type acceptable to the collection. The object root MUST be the Entry Part. The Entry Part's atom:content element MUST have a 'src' attribute whose value is the URI of the related media in the Media Part. The 'src' attribute value MUST be a 'cid:' URI as defined by [[RFC2392](#)]. The Content-Type: header of the POST request MUST specify "application/atom+xml;type=entry" or "application/atom+xml".

### **4. Server Processing**

A successful POST of a multipart/related representation to a Media Collection causes several resource-creation processes to occur as described in [[RFC5023](#)]. The Media Part is used to create a Media Resource as if it had been posted as a request body to the collection as described in [section 9.6](#), and the Entry Part is used to create the corresponding Media Link Entry. Assuming these two steps are successful, the server returns a 201 status code and a Location: header pointing to the newly created Media Link Entry. The applicable rules from [[RFC5023](#)] MUST be followed, including Slug: header processing.

While a multipart/related request replaces three round trips in the typical Media Resource creation scenario, AtomPub has no mechanism to

report partial success. Thus, the handling of a multipart/related request by the server MUST be atomic; that is, either succeed with a 201 Created status code, or return an error status code.

## **5. Service Document Extension**

An AtomPub service announces that it will accept multipart/related POSTs by an extension to the app:accept element. The "alternate" attribute's value MUST be one or more tokens, space-separated if more than one. The only token defined by this specification is "multipart-related". The presence of the "multipart-related" token in the 'alternate' attribute's value indicates that the collection accepts multipart/related POSTs whose Media Part has a content-type matching that specified in the app:accept element. The following example indicates a collection that allows the creation of resources with the Ogg Bitstream Format and will also accept them in multipart form.

```
<app:accept alternate="multipart-related">application/ogg</app:accept>
```

The 'alternate' attribute is foreign markup and will be ignored by clients that do not understand multipart/related uploads. In addition it permits the full range of the app:accept element to be used. The following indicates that the collection accepts any image media type and will also accept them in multipart form.

```
<app:accept alternate="multipart-related">image/*</app:accept>
```

The 'alternate' attribute allows clients that are unaware of multipart/related to continue to operate as normal since the alternate attribute is foreign markup. The alternative, which is to put a multipart/related media type in the app:accept element, loses flexibility since the 'type' parameter to the multipart/related media type accepts only media types and not media ranges.

## **6. Examples**

Here is an example service document that contains two media collections. The first collection accepts multipart/related POSTs for video media types only. It does not accept multipart/related POSTs for audio or text. The second collection accepts multipart/related POSTs for image/gif and image/png media types.

```
<?xml version="1.0" encoding='utf-8'?>
<service xmlns="http://www.w3.org/2007/app"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <workspace>
    <atom:title>Media Collections</atom:title>
    <collection
      href="http://example.org/blog/main" >
      <atom:title>Mostly Media</atom:title>
      <accept alternate="multipart-related">video/*</accept>
      <accept alternate="" >text/*</accept>
      <accept
        >audio/*</accept>
    </collection>
    <collection
      href="http://example.org/blog/pic" >
      <atom:title>Pictures Only</atom:title>
      <accept alternate="multipart-related">image/png</accept>
      <accept alternate="multipart-related">image/gif</accept>
    </collection>
  </workspace>
</service>
```

Here is an example interaction of a client creating a new Media Resource in the Pictures Only media collection using a png image in a multipart/related representation.

```

POST /blog/pic HTTP/1.1
Host: example.org
Content-Length: nnnn
Content-Type: multipart/related;
    boundary="====1605871705==";
    type="application/atom+xml"
Slug: The Beach
MIME-Version: 1.0

```

```

Media Post
--====1605871705==
Content-Type: application/atom+xml; charset="utf-8"
MIME-Version: 1.0

```

```

<?xml version="1.0"?>
<entry xmlns="http://www.w3.org/2005/Atom">
  <title>The Beach</title>
  <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
  <updated>2005-10-07T17:17:08Z</updated>
  <author><name>Daffy</name></author>
  <summary type="text">
    A nice sunset picture over the water.
  </summary>
  <content src="cid:99334422@example.com"
    type="image/gif" />
</entry>

```

```

--====1605871705==
Content-Type: image/gif
MIME-Version: 1.0
Content-ID: <99334422@example.com>

```

```

GIF89a...binary image data...
--====1605871705===--

```

If the request was successful the response might look like:

HTTP/1.1 201 Created  
Date: Fri, 7 Oct 2005 17:17:11 GMT  
Content-Length: nnn  
Content-Type: application/atom+xml;type=entry;charset="utf-8"  
Location: http://example.org/media/edit/the\_beach.atom

```
<?xml version="1.0"?>
<entry xmlns="http://www.w3.org/2005/Atom">
  <title>The Beach</title>
  <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
  <updated>2005-10-07T17:17:08Z</updated>
  <author><name>Daffy</name></author>
  <summary type="text">
    A nice sunset picture over the water.
  </summary>
  <content type="image/png"
    src="http://media.example.org/the_beach.png"/>
  <link rel="edit-media" type="image/png"
    href="http://media.example.org/edit/the_beach.png" />
  <link rel="edit"
    href="http://example.org/media/edit/the_beach.atom" />
</entry>
```

The initial request does not have to have the Entry Part first to indicate that it is the object root, it's location can be indicated by the 'start' attribute, as in this example.



```
POST /blog/pic HTTP/1.1
Host: example.org
Content-Length: nnnn
Content-Type: multipart/related;
    boundary="====1605871705==";
    type="application/atom+xml";
    start="<10101033@example.com>"
Slug: The Beach
MIME-Version: 1.0
```

```
Media Post
--====1605871705==
Content-Type: image/gif
MIME-Version: 1.0
Content-ID: <99334422@example.com>
```

```
GIF89a...binary image data...
--====1605871705==
Content-Type: application/atom+xml; charset="utf-8"
MIME-Version: 1.0
Content-ID: <10101033@example.com>
```

```
<?xml version="1.0"?>
<entry xmlns="http://www.w3.org/2005/Atom">
  <title>The Beach</title>
  <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
  <updated>2005-10-07T17:17:08Z</updated>
  <author><name>Daffy</name></author>
  <summary type="text">
    A nice sunset picture over the water.
  </summary>
  <content src="cid:99334422@example.com"
    type="image/gif" />
</entry>
--====1605871705=== - -
```

## **7. Security Considerations**

The security considerations are the same as delineated in [\[RFC5023\]](#).

## **8. IANA Considerations**

No IANA actions are required by this document.

## **9. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2387] Levinson, E., "The MIME Multipart/Related Content-type", [RFC 2387](#), August 1998.
- [RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", [RFC 2392](#), August 1998.
- [RFC5023] Gregorio, J. and B. de h0ra, "The Atom Publishing Protocol", [RFC 5023](#), October 2007.
- [1] <<http://www.imc.org/atom-protocol/>>

### Author's Address

Joe Gregorio (editor)  
Google

Email: [joe@bitworking.org](mailto:joe@bitworking.org)  
URI: <http://bitworking.org/>

## Full Copyright Statement

Copyright (C) The IETF Trust (2008).

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.

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, THE IETF TRUST 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.

## Intellectual Property

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](mailto:ietf-ipr@ietf.org).