

Network Working Group  
Internet-Draft  
Expires: August 5, 2006

J. Gregorio, Ed.  
BitWorking, Inc  
B. de h0ra, Ed.  
Propylon Ltd.  
February 01, 2006

**The Atom Publishing Protocol**  
**draft-ietf-atompub-protocol-08.txt**

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 August 5, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

The Atom Publishing Protocol (APP) is an application-level protocol for publishing and editing Web resources. The protocol is based on HTTP transport of Atom-formatted representations. The Atom format is documented in the Atom Syndication Format ([RFC4287](#)).

Editorial Note

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

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Notational Conventions . . . . .</a>	<a href="#">5</a>
<a href="#">3.</a>	<a href="#">Terminology . . . . .</a>	<a href="#">6</a>
<a href="#">4.</a>	<a href="#">Protocol Model . . . . .</a>	<a href="#">7</a>
<a href="#">5.</a>	<a href="#">Protocol Operations . . . . .</a>	<a href="#">8</a>
<a href="#">5.1</a>	<a href="#">Retrieving an Introspection Document . . . . .</a>	<a href="#">8</a>
<a href="#">5.2</a>	<a href="#">Creating a Resource . . . . .</a>	<a href="#">8</a>
<a href="#">5.3</a>	<a href="#">Editing a Resource . . . . .</a>	<a href="#">8</a>
<a href="#">5.3.1</a>	<a href="#">Retrieving a Resource . . . . .</a>	<a href="#">9</a>
<a href="#">5.3.2</a>	<a href="#">Updating a Resource . . . . .</a>	<a href="#">9</a>
<a href="#">5.3.3</a>	<a href="#">Deleting a Resource . . . . .</a>	<a href="#">9</a>
<a href="#">5.4</a>	<a href="#">Listing Collection Members . . . . .</a>	<a href="#">10</a>
<a href="#">5.5</a>	<a href="#">Use of HTTP Response codes . . . . .</a>	<a href="#">10</a>
<a href="#">6.</a>	<a href="#">XML-related Conventions . . . . .</a>	<a href="#">11</a>
<a href="#">6.1</a>	<a href="#">Referring to Information Items . . . . .</a>	<a href="#">11</a>
<a href="#">6.2</a>	<a href="#">XML Namespace Usage . . . . .</a>	<a href="#">11</a>
<a href="#">6.3</a>	<a href="#">Use of xml:base and xml:lang . . . . .</a>	<a href="#">11</a>
<a href="#">6.4</a>	<a href="#">RELAX NG Schema . . . . .</a>	<a href="#">12</a>
<a href="#">7.</a>	<a href="#">Introspection Documents . . . . .</a>	<a href="#">13</a>
<a href="#">7.1</a>	<a href="#">Example . . . . .</a>	<a href="#">13</a>
<a href="#">7.2</a>	<a href="#">Element Definitions . . . . .</a>	<a href="#">14</a>
<a href="#">7.2.1</a>	<a href="#">The "app:service" Element . . . . .</a>	<a href="#">14</a>
<a href="#">7.2.2</a>	<a href="#">The "app:workspace" Element . . . . .</a>	<a href="#">14</a>
<a href="#">7.2.3</a>	<a href="#">The "app:collection" Element . . . . .</a>	<a href="#">15</a>
<a href="#">7.2.4</a>	<a href="#">The "app:member-type" Element . . . . .</a>	<a href="#">16</a>
<a href="#">8.</a>	<a href="#">Collections . . . . .</a>	<a href="#">17</a>
<a href="#">8.1</a>	<a href="#">Creating resources with POST . . . . .</a>	<a href="#">17</a>
<a href="#">8.1.1</a>	<a href="#">Example . . . . .</a>	<a href="#">17</a>
<a href="#">8.1.2</a>	<a href="#">Title: Header . . . . .</a>	<a href="#">17</a>
<a href="#">8.2</a>	<a href="#">Entry Collections . . . . .</a>	<a href="#">18</a>
<a href="#">8.2.1</a>	<a href="#">Editing entries with foreign markup . . . . .</a>	<a href="#">18</a>
<a href="#">8.3</a>	<a href="#">Media Collections . . . . .</a>	<a href="#">18</a>
<a href="#">8.3.1</a>	<a href="#">Editing Media Resources . . . . .</a>	<a href="#">18</a>
<a href="#">8.3.2</a>	<a href="#">Editing Media Metadata . . . . .</a>	<a href="#">19</a>
<a href="#">9.</a>	<a href="#">Listing Collections . . . . .</a>	<a href="#">20</a>
<a href="#">9.1</a>	<a href="#">Collection Paging . . . . .</a>	<a href="#">20</a>
<a href="#">10.</a>	<a href="#">Atom Format Link Relation Extensions . . . . .</a>	<a href="#">22</a>
<a href="#">10.1</a>	<a href="#">The "edit" Link Relation . . . . .</a>	<a href="#">22</a>
<a href="#">11.</a>	<a href="#">Atom Publishing Control Extensions . . . . .</a>	<a href="#">23</a>
<a href="#">11.1</a>	<a href="#">The Atom Publishing Control Namespace . . . . .</a>	<a href="#">23</a>
<a href="#">11.2</a>	<a href="#">The "pub:control" Element . . . . .</a>	<a href="#">23</a>
<a href="#">11.2.1</a>	<a href="#">The "pub:draft" Element . . . . .</a>	<a href="#">23</a>
<a href="#">12.</a>	<a href="#">Atom Publishing Protocol Example . . . . .</a>	<a href="#">24</a>



<a href="#">13.</a>	Securing the Atom Protocol . . . . .	<a href="#">26</a>
<a href="#">14.</a>	Security Considerations . . . . .	<a href="#">27</a>
<a href="#">15.</a>	IANA Considerations . . . . .	<a href="#">28</a>
<a href="#">16.</a>	References . . . . .	<a href="#">30</a>
<a href="#">16.1</a>	Normative References . . . . .	<a href="#">30</a>
<a href="#">16.2</a>	Informative References . . . . .	<a href="#">31</a>
	Authors' Addresses . . . . .	<a href="#">32</a>
<a href="#">A.</a>	Contributors . . . . .	<a href="#">33</a>
<a href="#">B.</a>	RELAX NG Compact Schema . . . . .	<a href="#">34</a>
<a href="#">C.</a>	Revision History . . . . .	<a href="#">37</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">40</a>



## **1. Introduction**

The Atom Publishing Protocol is an application-level protocol for publishing and editing Web resources using HTTP [[RFC2616](#)] and XML 1.0 [[W3C.REC-xml-20040204](#)]. The protocol supports the creation of arbitrary web resources and provides facilities for:

- o Collections: Sets of resources, which may be retrieved in whole or in part.
- o Introspection: Discovering and describing collections.
- o Editing: Creating, updating and deleting resources.

## **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](#)].

Note: The Introspection Document allows the use of IRIs [[RFC3987](#)], as well as URIs [[RFC3986](#)]. Every URI is an IRI, so any URI can be used where an IRI is needed. How to map an IRI to a URI is specified in [Section 3.1](#) of Internationalized Resource Identifiers (IRIs) [[RFC3987](#)].

### **3. Terminology**

For convenience, this protocol may be referred to as the "Atom Protocol" or "APP".

URI/IRI - A Uniform Resource Identifier and Internationalized Resource Identifier. These terms and the distinction between them are defined in [[RFC3986](#)] and [[RFC3987](#)]. Note that IRIs are mapped to URIs before dereferencing takes place.

Resource - A network-accessible data object or service identified by an IRI, as defined in [[RFC2616](#)]. See [[W3C.REC-webarch-20041215](#)] for further discussion on resources.

The phrase "the URI of a document" in this specification is shorthand for "an URI which, when dereferenced, is expected to produce that document as a representation".

Representation - An entity included with a request or response as defined in [[RFC2616](#)].

Collection - A resource that contains a set of member IRIs. See [Section 8](#).

Member - A resource whose IRI is listed in a Collection.

Introspection Document - A document that describes the location and capabilities of one or more Collections. See [Section 7](#).





#### **4. Protocol Model**

The Atom Publishing Protocol uses HTTP to edit and author web resources. The Atom Protocol uses the following HTTP methods:

- o GET is used to retrieve a representation of a resource or perform a query.
- o POST is used to create a new, dynamically-named resource.
- o PUT is used to update a known resource.
- o DELETE is used to remove a resource.

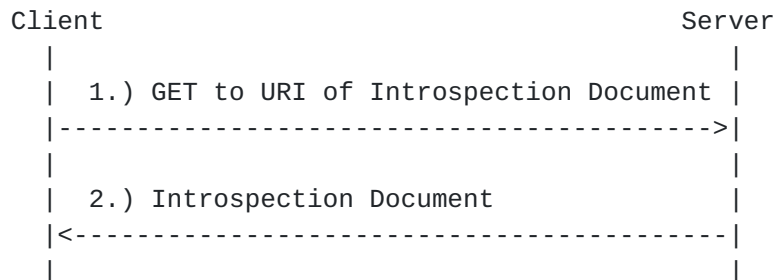
Along with operations on resources, the Atom Protocol provides list-based structures, called Collections, for managing and organising resources, called Members. Collections contain the IRIs of, and metadata about, their Member resources. For authoring and editing of resources to commence, an Atom Protocol client can examine Introspection Documents which represent server-defined groups of Collections.

Note that when an IRI is used for resource retrieval over HTTP, the IRI is first converted to a URI according the procedure defined in [\[RFC3987\] section 3.1](#). The resource that the IRI locates is the same as the one located by the URI obtained after converting the IRI.



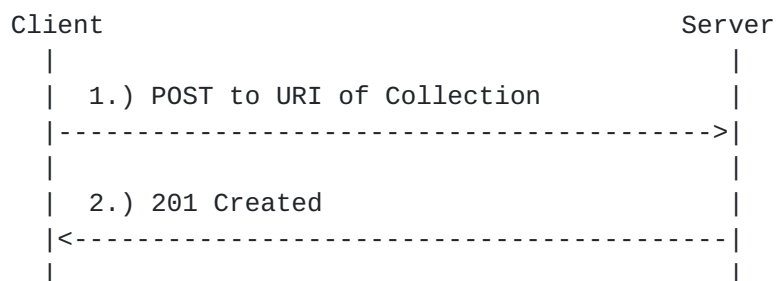
## 5. Protocol Operations

### 5.1 Retrieving an Introspection Document



1. The client sends a GET request to the URI of the Introspection Document.
2. The server responds with the document enumerating the IRIs of a set of Collections and the capabilities of those Collections supported by the server. The content of this document can vary based on aspects of the client request, including, but not limited to, authentication credentials.

### 5.2 Creating a Resource



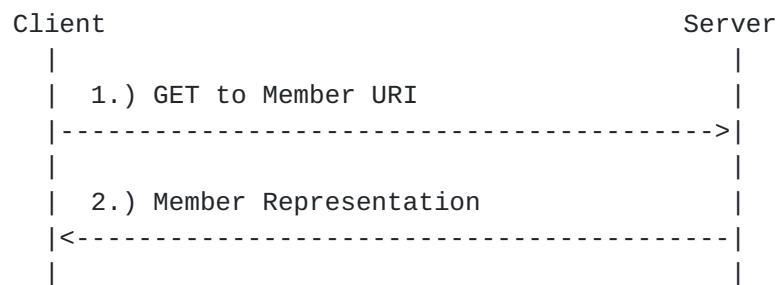
1. The client POSTs a representation of the Member to the URI of the collection.
2. If the Member resource was created successfully, the server responds with a status code of 201 and a Location: header that contains the URI of the newly created resource.

### 5.3 Editing a Resource

Once a resource has been created and its URI is known, that URI may be used to retrieve, update, and delete the resource.

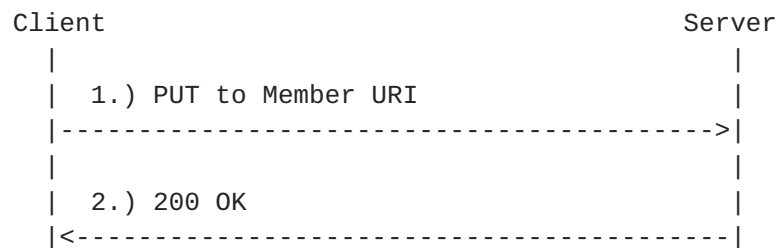


### [5.3.1](#) Retrieving a Resource



1. The client sends a GET request to the Member's URI to retrieve its representation.
2. The server responds with the representation of the resource.

### [5.3.2](#) Updating a Resource



1. The client PUTs an updated representation to the Member's URI.
2. Upon a successful update of the resource the server responds with a status code of 200.

### [5.3.3](#) Deleting a Resource

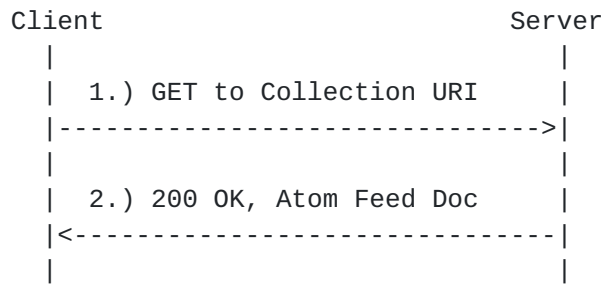


1. The client sends a DELETE request to the Member's URI.
2. Upon the successful deletion of the resource the server responds with a status code of 200.



#### 5.4 Listing Collection Members

To list the members of a Collection the client sends a GET request to the Collection's URI. An Atom Feed Document is returned containing one Atom Entry for each member resource. See [Section 9](#) and [Section 10](#) for a description of the feed contents.



1. The client sends a GET request to the Collection's URI.
2. The server responds with an Atom Feed Document containing the IRIs of the collection members.

#### 5.5 Use of HTTP Response codes

The Atom Protocol uses the response status codes defined in HTTP to indicate the success or failure of an operation. Consult the HTTP specification [[RFC2616](#)] for detailed definitions of each status code. It is RECOMMENDED that entities contained within HTTP 4xx and 5xx responses include an explanation of the error using natural language.





## **6. XML-related Conventions**

The Atom Protocol Introspection format is specified in terms of the XML Information Set [[W3C.REC-xml-infoset-20040204](#)], serialised as XML 1.0 [[W3C.REC-xml-20040204](#)]. Atom Publishing Protocol Documents MUST be well-formed XML. This specification does not define any DTDs for Atom Protocol, and hence does not require them to be "valid" in the sense used by XML.

### **6.1 Referring to Information Items**

This specification uses a shorthand for two common terms: the phrase "Information Item" is omitted when discussing Element Information Items and Attribute Information Items. Therefore, when this specification uses the term "element," it is referring to an Element Information Item in InfoSet terms. Likewise, when it uses the term "attribute," it is referring to an Attribute Information Item.

### **6.2 XML Namespace Usage**

The namespace name [[W3C.REC-xml-names-19990114](#)] for the XML format described in this specification is:

<http://purl.org/atom/app#>

This specification uses the prefix "app:" for the namespace name. The choice of namespace prefix is not semantically significant.

This specification also uses the prefix "atom:" for "http://www.w3.org/2005/Atom", the namespace name of the Atom Publishing Format [[RFC4287](#)].

### **6.3 Use of xml:base and xml:lang**

XML elements defined by this specification MAY have an xml:base attribute [[W3C.REC-xmlbase-20010627](#)]. When xml:base is used, it serves the function described in [section 5.1.1](#) of URI Generic Syntax [[RFC3986](#)], establishing the base URI (or IRI) for resolving any relative references found within the effective scope of the xml:base attribute.

Any element defined by this specification MAY have an xml:lang attribute, whose content indicates the natural language for the element and its descendents. The language context is only significant for elements and attributes declared to be "Language-Sensitive" by this specification. Requirements regarding the content and interpretation of xml:lang are specified in [Section 2.12](#) of XML 1.0 [[W3C.REC-xml-20040204](#)].



```
appCommonAttributes =  
  attribute xml:base { atomUri }?,  
  attribute xml:lang { atomLanguageTag }?,  
  undefinedAttribute*
```

#### [6.4](#) RELAX NG Schema

Some sections of this specification are illustrated with fragments of a non-normative RELAX NG Compact schema [[RNC](#)]. A complete schema appears in [Appendix B](#). However, the text of this specification provides the definition of conformance.

## 7. Introspection Documents

For authoring to commence, a client needs to first discover the capabilities and locations of collections offered. This is done using Introspection Documents. An Introspection Document describes workspaces, which are server-defined groupings of collections.

Introspection documents are identified with the "application/atomserv+xml" media type (see [Section 15](#)).

While an introspection document allows multiple workspaces, there is no requirement that a service support multiple workspaces. In addition, a collection MAY appear in more than one workspace.

### 7.1 Example

```
<?xml version="1.0" encoding='utf-8'?>
<service xmlns="http://purl.org/atom/app#">
  <workspace title="Main Site" >
    <collection
      title="My Blog Entries"
      href="http://example.org/reilly/main" >
        <member-type>entry</member-type>
      </collection>
    <collection
      title="Pictures"
      href="http://example.org/reilly/pic" >
        <member-type>media</member-type>
      </collection>
    </workspace>
  <workspace title="Side Bar Blog">
    <collection title="Remaindered Links"
      href="http://example.org/reilly/list" >
        <member-type>entry</member-type>
      </collection>
    </workspace>
  </service>
```

This Introspection Document describes two workspaces. The first, called "Main Site", has two collections called "My Blog Entries" and "Pictures" whose URIs are "http://example.org/reilly/main" and "http://example.org/reilly/pic" respectively. "My Blog Entries" is an Entry collection and "Pictures" is a Media collection. Entry and Media collections are discussed in [Section 7.2.4](#).

The second workspace is called "Side Bar Blog" and has a single collection called "Remaindered Links" whose collection URI is "http://example.org/reilly/list". "Remaindered Links" is an Entry



collection.

## [7.2](#) Element Definitions

### [7.2.1](#) The "app:service" Element

The root of an introspection document is the "app:service" element.

The "app:service" element is the container for introspection information associated with one or more workspaces. An app:service element MUST contain one or more app:workspace elements.

```
namespace app = "http://purl.org/atom/app#"
start = appService
```

```
appService =
  element app:service {
    appCommonAttributes,
    ( appWorkspace+
      & extensionElement* )
  }
```

### [7.2.2](#) The "app:workspace" Element

The "app:workspace" element contains information elements about the collections of resources available for editing. The app:workspace element MUST contain one or more app:collection elements.

```
appWorkspace =
  element app:workspace {
    appCommonAttributes,
    attribute title { text },
    ( appCollection+
      & extensionElement* )
  }
```

In an app:workspace element, the first app:collection element of each type MUST refer to the preferred or primary collection. In the following example, the "Entries" collection would be considered the preferred (or primary) entries collection of the workspace and the "Photos" collection would be considered the primary media collection:





```
<service>
  <workspace title="My Blog">
    <collection title="Entries"
      href="http://example.org/myblog/entries">
      <member-type>entry</member-type>
    </collection>
    <collection title="Photos"
      href="http://example.org/myblog/fotes">
      <member-type>media</member-type>
    </collection>
  </workspace>
</service>
```

#### [7.2.2.1](#) The "title" Attribute

The `app:workspace` element MUST contain a "title" attribute, which gives a human-readable name for the workspace. This attribute is Language-Sensitive.

#### [7.2.3](#) The "app:collection" Element

The "app:collection" describes an Atom Protocol collection. One child element is defined here for app:collection: "app:member-type".

```
appCollection =
  element app:collection {
    appCommonAttributes,
    attribute title { text },
    attribute href { text },
    ( appMemberType
      & appListTemplate
      & extensionElement* )
  }
```

##### [7.2.3.1](#) The "title" Attribute

The `app:collection` element MUST contain a "title" attribute, whose value gives a human-readable name for the collection. This attribute is Language-Sensitive.

##### [7.2.3.2](#) The "href" Attribute

The `app:collection` element MUST contain a "href" attribute, whose value gives the IRI of the collection.



#### [7.2.4](#) The "app:member-type" Element

The app:collection element MUST contain one "app:member-type" element. The app:member-type element value specifies the types of members that can appear in the collection.

```
appMemberType =  
  element app:member-type {  
    appCommonAttributes,  
    ( appTypeValue )  
  }
```

```
appTypeValue = "entry" | "media"
```

This specification defines two values for the app:member-type element:

- o "entry" - Indicates the collection contains only member resources whose representation MUST be an Atom Entry. Further constraints on the representations of members in a collection of type "entry" are listed in [Section 8.2](#).
- o "media" - Indicates the collection contains member resources whose representation can be of any media type. Additional constraints are listed in [Section 8.3](#).



## **8. Collections**

### **8.1 Creating resources with POST**

To add members to a collection, clients send POST requests to the collection's URI. Collections MAY impose constraints on the media-types that are created in a collection and MAY generate a response with a status code of 415 ("Unsupported Media Type"). On successful creation, the response to the POST request MUST return a Location: header with the URI of the newly created resource.

#### **8.1.1 Example**

Below, the client sends a POST request containing an Atom Entry representation to the URI of the Collection:

```
POST /myblog/entries HTTP/1.1
Host: example.org
User-Agent: Thingio/1.0
Content-Type: application/atom+xml
Content-Length: nnn

<entry xmlns="http://www.w3.org/2005/Atom">
  <title>Atom-Powered Robots Run Amok</title>
  <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
  <updated>2003-12-13T18:30:02Z</updated>
  <content>Some text.</content>
</entry>
```

The server signals a successful creation with a status code of 201 and the response includes a 'Location' header indicating the URI of the Atom Entry.

```
HTTP/1.1 201 Created
Date: Fri, 7 Oct 2005 17:17:11 GMT
Content-Length: 0
Location: http://example.org/edit/first-post.atom
```

#### **8.1.2 Title: Header**

A POST to a Media Collection creating a resource SHOULD contain a Title: header that indicates the client's suggested title for the resource:



```
POST /myblog/fotes HTTP/1.1
Host: example.org
User-Agent: Thingio/1.0
Content-Type: image/png
Content-Length: nnnn
Title: An Atom-Powered Robot
```

...binary data...

The server MAY ignore the content of the Title: header or modify the suggested title.

Title = "Title" ":" [TEXT]

The syntax of this header MUST conform to the augmented BNF grammar in [section 2.1](#) of the HTTP/1.1 specification [[RFC2616](#)]. The [TEXT] rule is described in [section 2.2](#) of the same document. Words of \*TEXT MAY contain characters from character sets other than [[ISO88591](#)] only when encoded according to the rules of [[RFC2047](#)].

## [8.2](#) Entry Collections

Entry Collections are collections that restrict their membership to Atom Entries. They are identified by having an app:member-type of "entry". Every member representation MAY contain an atom:link element with a link relation of "edit" that contains the IRI of the member resource. Member representations MAY contain a pub:control element ([Section 11](#)).

### [8.2.1](#) Editing entries with foreign markup

To avoid unintentional loss of data when editing entry collection members, Atom Protocol clients SHOULD preserve all metadata, including unknown foreign markup, that has not been intentionally modified.

## [8.3](#) Media Collections

Media Collections are collections whose member representations are not constrained. They are identified by having an app:member-type of "media".

### [8.3.1](#) Editing Media Resources

When listing the contents of a Media Collection, every Entry in the Atom Feed Document MUST have an atom:content element with a "src" attribute containing the IRI of the media resource itself. This value may be used to update and delete resources as described in





[Section 5.3](#). When creating a public, read-only reference to the member resource, a client SHOULD use this value.

### **[8.3.2](#) Editing Media Metadata**

Entries in a Media Collection MAY contain an atom:link element with a link relation of "edit" that contains the IRI of an Atom Entry document representing the metadata of the member resource. A client MAY use this to edit the metadata associated with the resource.

## **9. Listing Collections**

Collection resources MUST provide representations in the form of Atom Feed documents. Each entry in the Feed Document MUST have an atom:link element with a relation of "edit" (See [Section 10.1](#)).

The entries in the returned Atom Feed MUST be ordered by their "atom:updated" property, with the most recently updated entries coming first in the document order. Clients SHOULD be constructed in consideration that changes which do not alter the entry's atom:updated value will not affect the position of the entry in a collection.

Clients MUST NOT assume that an Atom Entry returned in the Feed is a full representation of a member resource and SHOULD perform a GET on the member resource before editing.

Collections can contain large numbers of resources. A naive client such as a web spider or web browser could be overwhelmed if the response to a GET contained every entry in the collection, and the server would waste large amounts of bandwidth and processing time on clients unable to handle the response. For this reason, servers MAY return a partial listing containing the most recently updated member resources. Such partial feed documents MUST have an atom:link with a "next" relation whose "href" value is the URI of the next partial listing of the collection (the least recently updated member resources) where it exists. This is called "collection paging".

### **9.1 Collection Paging**

Atom Protocol servers MUST provide representations of collections as Atom feed documents whose entries represent the collection's members. The returned Atom feed MAY NOT contain entries for all the collection's members. Instead, the Atom feed document MAY contain link elements with "rel" attribute values of "next", "previous", "first" and "last" that can be used to navigate through the complete set of matching entries.

For instance, suppose a client is supplied the URI "http://example.org/entries/go" of a collection of member entries, where the server as a matter of policy avoids generating feed documents containing more than 10 entries. The Atom feed document for the collection will then represent the first 'page' in a set of 10 linked feed documents. The "first" relation will reference the initial feed document in the set and the "last" relation references the final feed document in the set. Within each document, the "next" and "previous" link relations reference the preceding and subsequent documents.



```
<feed xmlns="http://www.w3.org/2005/Atom">
  <link rel="first"
        href="http://example.org/entries/go" />
  <link rel="next"
        href="http://example.org/entries/2" />
  <link rel="last"
        href="http://example.org/entries/10" />
  ...
</feed>
```

The "next" and "previous" link elements for the feed 'page' located at "http://example.org/entries/2" would look like this:

```
<feed xmlns="http://www.w3.org/2005/Atom">
  <link rel="first"
        href="http://example.org/entries/go" />
  <link rel="previous"
        href="http://example.org/entries/go" />
  <link rel="next"
        href="http://example.org/entries/3" />
  <link rel="last"
        href="http://example.org/entries/10" />
  ...
</feed>
```



## **10. Atom Format Link Relation Extensions**

### **10.1 The "edit" Link Relation**

The Atom Protocol adds the value "edit" to the Atom Registry of Link Relations (see [section 7.1 of \[RFC4287\]](#)). The value of "edit" specifies that the IRI in the value of the href attribute is the IRI of an editable Atom Entry Document associated with a resource. In a Media Collection this IRI may be used to update the metadata associated with a Media Resource. In an Entry Collection this IRI may be used to update and delete the member resource itself. The link relation MAY appear in Atom Entry representations as well as Entry and Media Collections.



## **11. Atom Publishing Control Extensions**

### **11.1 The Atom Publishing Control Namespace**

This specification defines an Atom Format extension for publishing control called Atom Publishing Control. The namespace name for the Atom Publishing Control's XML vocabulary is "http://example.net/appns/". This specification uses "pub:" for the namespace prefix. The choice of namespace prefix is not semantically significant.

### **11.2 The "pub:control" Element**

```
namespace pub = "http://example.net/appns/"
```

```
pubControl =  
  element pub:control {  
    atomCommonAttributes,  
    pubDraft?  
    & extensionElement  
  }
```

```
pubDraft =  
  element pub:draft { "yes" | "no" }
```

The "pub:control" element MAY appear as a child of an "atom:entry" which is being created or updated via the Atom Publishing Protocol. The "pub:control" element, if it does appear in an entry, MUST only appear at most one time. The "pub:control" element is considered foreign markup as defined in [Section 6 of \[RFC4287\]](#).

The "pub:control" element and its child elements MAY be included in Atom Feed or Entry Documents.

The "pub:control" element MAY contain exactly one "pub:draft" element as defined here, and MAY contain zero or more extension elements as outlined in [Section 6 of \[RFC4287\]](#). Both clients and servers MUST ignore foreign markup present in the pub:control element.

#### **11.2.1 The "pub:draft" Element**

The number of "pub:draft" elements in "pub:control" MUST be zero or one. Its value MUST be one of "yes" or "no". A value of "no" means that the entry MAY be made publicly visible. If the "pub:draft" element is missing then the value MUST be understood to be "no". The pub:draft element MAY be ignored.





## **12. Atom Publishing Protocol Example**

This is an example of a client creating a new entry with an image. The client has an image to publish and an entry that includes an HTML "img" element that uses that image. In this scenario we consider a client that has IRIs of two collections, an entry collection and a media collection, both of which were discovered through an introspection document. The IRI of the entry collection is:

`http://example.net/blog/edit/`

The IRI of the media collection is:

`http://example.net/binary/edit`

First the client creates a new image resource by POSTing the image to the IRI of the media collection.

```
POST /binary/edit/ HTTP/1.1
Host: example.net
User-Agent: Thingio/1.0
Content-Type: image/png
Content-Length: nnnn
Title: A picture of the beach
```

...binary data...

The member resource is created and an HTTP status code of 201 is returned.

```
HTTP/1.1 201 Created
Date: Fri, 25 Mar 2005 17:17:11 GMT
Content-Length: nnnn
Content-Type: application/atom+xml
Location: http://example.net/binary/edit/b/129.png
```

```
<?xml version="1.0" encoding="utf-8"?>
<entry xmlns="http://www.w3.org/2005/Atom">
  <title>A picture of the beach</title>
  <link rel="edit"
        href="http://example.net/binary/edit/b/129.png"/>
  <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-568596895695</id>
  <updated>2005-09-02T10:30:00Z</updated>
  <summary>Waves</summary>
  <content type="image/png"
        src="http://example.net/binary/readonly/129.png"/>
</entry>
```



The client then POSTs the Atom Entry that refers to the newly created image resource. Note that the client takes the URI `http://example.net/binary/readonly/129.png` and uses it in the 'img' element in the Entry content:

```
POST /blog/edit/ HTTP/1.1
```

```
Host: example.net
```

```
User-Agent: Thingio/1.0
```

```
Content-Type: application/atom+xml
```

```
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="utf-8"?>
<entry xmlns="http://www.w3.org/2005/Atom">
  <title>What I did on my summer vacation</title>
  <link href="http://example.org/atom05"/>
  <id>urn:uuid:1225c695-ffb8-4ebb-aaaa-80da354efa6a</id>
  <updated>2005-09-02T10:30:00Z</updated>
  <summary>Beach!</summary>
  <content type="xhtml" xml:lang="en">
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p>We went to the beach for summer vacation.
        Here is a picture of the waves rolling in:
        
      </p>
    </div>
  </content>
</entry>
```



### **13. Securing the Atom Protocol**

All instances of publishing Atom Format entries SHOULD be protected by authentication to prevent posting or editing by unknown sources. Atom Protocol servers and clients MUST support one of the following authentication mechanisms, and SHOULD support both.

- o HTTP Digest Authentication [[RFC2617](#)]
- o CGI Authentication

Atom Protocol servers and clients MAY support encryption of the session using TLS (see [[RFC2246](#)]).

There are cases where an authentication mechanism might not be required, such as a publicly editable Wiki, or when using POST to send comments to a site that does not require authentication from a commenter.

#### **13.1 CGI Authentication**

[[anchor27: note: this section is incomplete; cgi-authentication is described but is unspecified.]] This authentication method is included as part of the protocol to allow Atom Protocol servers and clients that cannot use HTTP Digest Authentication but where the user can both insert its own HTTP headers and create a CGI program to authenticate entries to the server. This scenario is common in environments where the user cannot control what services the server employs, but the user can write their own HTTP services.



## **14. Security Considerations**

The security of the Atom Protocol is based on HTTP Digest Authentication and/or CGI Authentication `[[anchor29: note: refers to incomplete section]]`. Any weaknesses in either of these authentication schemes will affect the security of the Atom Publishing Protocol.

Both HTTP Digest Authentication and CGI Authentication `[[anchor30: note: refers to incomplete section]]` are susceptible to dictionary-based attacks on the shared secret. If the shared secret is a password (instead of a random string with sufficient entropy), an attacker can determine the secret by exhaustively comparing the authenticating string with hashed results of the public string and dictionary entries.

See [\[RFC2617\]](#) for the description of the security properties of HTTP Digest Authentication.

`[[anchor31: expand on HTTP basic and digest authentication, or refer.]]`

`[[anchor32: note: talk here about denial of service attacks using large XML files, or the billion laughs DTD attack.]]`





## 15. IANA Considerations

An Atom Publishing Protocol Introspection Document, when serialized as XML 1.0, can be identified with the following media type:

MIME media type name: application

MIME subtype name: atomserv+xml

Mandatory parameters: None.

Optional parameters:

"charset": This parameter has identical semantics to the charset parameter of the "application/xml" media type as specified in [\[RFC3023\]](#).

Encoding considerations: Identical to those of "application/xml" as described in [\[RFC3023\]](#), [section 3.2](#).

Security considerations: As defined in this specification.  
[[anchor33: update upon publication]]

In addition, as this media type uses the "+xml" convention, it shares the same security considerations as described in [\[RFC3023\]](#), [section 10](#).

Interoperability considerations: There are no known interoperability issues.

Published specification: This specification. [[anchor34: update upon publication]]

Applications that use this media type: No known applications currently use this media type.

Additional information:

Magic number(s): As specified for "application/xml" in [\[RFC3023\]](#), [section 3.2](#).

File extension: .atomsrv

Fragment identifiers: As specified for "application/xml" in [\[RFC3023\]](#), [section 5](#).



Base URI: As specified in [\[RFC3023\], section 6](#).

Macintosh File Type code: TEXT

Person and email address to contact for further information: Joe  
Gregorio <joe@bitworking.org>

Intended usage: COMMON

Author/Change controller: This specification's author(s). [[anchor35:  
update upon publication]]

## **16. References**

### **16.1 Normative References**

- [ISO88591] ISO, "International Standard -- Information Processing -- 8-bit Single-Byte Coded Graphic Character Sets -- Part 1: Latin alphabet No. 1," January 1987.
- [RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", [RFC 2047](#), November 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", [RFC 2246](#), January 1999.
- [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.
- [RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", [RFC 2617](#), June 1999.
- [RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", [RFC 3023](#), January 2001.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.
- [RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", [RFC 3987](#), January 2005.
- [RFC4287] Nottingham, M. and R. Sayre, "The Atom Syndication Format", [RFC 4287](#), December 2005.
- [W3C.REC-xml-20040204] Yergeau, F., Paoli, J., Sperberg-McQueen, C., Bray, T., and E. Maler, "Extensible Markup Language (XML) 1.0 (Third Edition)", W3C REC REC-xml-20040204, February 2004.
- [W3C.REC-xml-infoset-20040204] Cowan, J., Tobin, R., and A. Layman, "XML Information Set



(Second Edition)", W3C REC W3C.REC-xml-infoset-20040204, February 2004.

[W3C.REC-xml-names-19990114]

Hollander, D., Bray, T., and A. Layman, "Namespaces in XML", W3C REC REC-xml-names-19990114, January 1999.

[W3C.REC-xmlbase-20010627]

Marsh, J., "XML Base", W3C REC W3C.REC-xmlbase-20010627, June 2001.

## **16.2 Informative References**

[RNC] Clark, J., "RELAX NG Compact Syntax", December 2001.

[W3C.REC-webarch-20041215]

Walsh, N. and I. Jacobs, "Architecture of the World Wide Web, Volume One", W3C REC REC-webarch-20041215, December 2004.





## URIs

[1] <<http://www.imc.org/atom-protocol/index.html>>

## Authors' Addresses

Joe Gregorio (editor)  
BitWorking, Inc  
1002 Heathwood Dairy Rd.  
Apex, NC 27502  
US

Phone: +1 919 272 3764  
Email: [joe@bitworking.com](mailto:joe@bitworking.com)  
URI: <http://bitworking.com/>

Bill de h0ra (editor)  
Propylon Ltd.  
45 Blackbourne Square, Rathfarnham Gate  
Dublin, Dublin D14  
IE

Phone: +353-1-4927444  
Email: [bill.dehora@propylon.com](mailto:bill.dehora@propylon.com)  
URI: <http://www.propylon.com/>



**[Appendix A.](#) Contributors**

The content and concepts within are a product of the Atom community and the Atompub Working Group.

## [Appendix B](#). RELAX NG Compact Schema

This appendix is informative.

The Relax NG schema explicitly excludes elements in the Atom Protocol namespace which are not defined in this revision of the specification. Requirements for Atom Protocol processors encountering such markup are given in [Section 6.2](#) and [Section 6.3 of \[RFC4287\]](#).

```
# -*- rnc -*-
# RELAX NG Compact Syntax Grammar for the Atom Protocol

namespace app = "http://purl.org/atom/app#"
namespace local = ""

start = appService

# common:attrs

appCommonAttributes =
  attribute xml:base { atomUri }?,
  attribute xml:lang { atomLanguageTag }?,
  undefinedAttribute*

undefinedAttribute =
  attribute * - (xml:base | xml:lang | local:*) { text }

atomUri = text

atomLanguageTag = xsd:string {
  pattern = "[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*"
}

# app:service

appService =
  element app:service {
    appCommonAttributes,
    ( appWorkspace+
      & extensionElement* )
  }

# app:workspace

appWorkspace =
  element app:workspace {
    appCommonAttributes,
```



```
        attribute title { text },
        ( appCollection+
          & extensionElement* )
    }

# app:collection

appCollection =
    element app:collection {
        appCommonAttributes,
        attribute title { text },
        attribute href { atomUri },
        ( appMemberType
          & extensionElement* )
    }

# app:member

appMemberType =
    element app:member-type {
        appCommonAttributes,
        ( appTypeValue )
    }

appTypeValue = "entry" | "media"

# Simple Extension

simpleExtensionElement =
    element * - app:* {
        text
    }

# Structured Extension

structuredExtensionElement =
    element * - app:* {
        (attribute * { text }+,
         (text|anyElement)*)
        | (attribute * { text }*,
         (text?, anyElement+, (text|anyElement)*))
    }

# Other Extensibility

extensionElement =
```



```
simpleExtensionElement | structuredExtensionElement
```

```
# Extensions
```

```
anyElement =  
  element * {  
    (attribute * { text }  
    | text  
    | anyElement)*  
  }
```

```
# EOF
```



## **Appendix C. Revision History**

[draft-ietf-atompub-protocol-08](#): added info:etag; added wording re IRI/URI; fixed URI/IRI ; next/previous fixed as per Atom LinkRelations Attribute (<http://www.imc.org/atom-protocol/mail-archive/msg04095.html>); incorporated: PaceEditLinkMustToMay; PaceMissingDraftHasNoMeaning, PaceRemoveMemberTypeMust, PaceRemoveMemberTypePostMust, PaceTitleHeaderOnlyInMediaCollections, PacePreserveForeignMarkup, PaceClarifyTitleHeader, PaceClarifyMediaResourceLinks, PaceTwoPrimaryCollections;

[draft-ietf-atompub-protocol-07](#): updated Atom refs to [RFC4287](#); incorporated PaceBetterHttpStatusCode; PaceClarifyCollectionAndDeleteMethodByWritingLessInsteadOfMore; PaceRemoveAcceptPostText; PaceRemoveListTemplate2; PaceRemoveRegistry; PaceRemoveWhoWritesWhat; PaceSimplifyClarifyBetterfyRemoveBogusValidityText; PaceCollectionOrderSignificance; PaceFixLostIntrospectionText; PaceListPaging; PaceCollectionControl; element typo in Listing collections para3 (was app:member-type, not app:list-template); changed post atom entry example to be valid. Dropped inline use of 'APP'. Removed nested diagram from [section 4](#). Added ed notes in the security section.

[draft-ietf-atompub-protocol-06](#) - Removed: Robert Sayre from the contributors section per his request. Added in PaceCollectionControl. Fixed all the {daterange} verbage and examples so they all use a dash. Added full rnc schema. Collapsed Introspection and Collection documents into a single document. Removed {dateRange} queries. Renamed search to list. Moved discussion of media and entry collection until later in the document and tied the discussion to the Introspection element app:member-type.

[draft-ietf-atompub-protocol-05](#) - Added: Contributors section. Added: de h0ra to editors. Fixed: typos. Added diagrams and description to model section. Incorporates PaceAppDocuments, PaceAppDocuments2, PaceSimplifyCollections2 (large-sized chunks of it anyhow: the notions of Entry and Generic resources, the [section 4](#) language on the Protocol Model, 4.1 through 4.5.2, the notion of a Collection document, as in [Section 5](#) through 5.3, [Section 7](#) "Collection resources", Selection resources (modified from pace which talked about search); results in major mods to Collection Documents, [Section 9.2](#) "Title: Header" and brokeout para to [section 9.1](#) Editing Generic Resources). Added XML namespace and language section. Some cleanup of front matter. Added Language Sensitivity to some attributes. Removed resource descriptions from terminology. Some juggling of sections. See:



<http://www.imc.org/atom-protocol/mail-archive/msg01812.html>.

[draft-ietf-atompub-protocol-04](#) - Add ladder diagrams, reorganize, add SOAP interactions

[draft-ietf-atompub-protocol-03](#) - Incorporates PaceSliceAndDice3 and PaceIntrospection.

[draft-ietf-atompub-protocol-02](#) - Incorporates Pace409Response, PacePostLocationMust, and PaceSimpleResourcePosting.

[draft-ietf-atompub-protocol-01](#) - Added in sections on Responses for the EditURI. Allow 2xx for response to EditURI PUTs. Elided all mentions of WSSE. Started adding in some normative references. Added the section "Securing the Atom Protocol". Clarified that it is possible that the PostURI and FeedURI could be the same URI. Cleaned up descriptions for Response codes 400 and 500.

Rev [draft-ietf-atompub-protocol-00](#) - 5Jul2004 - Renamed the file and re-titled the document to conform to IETF submission guidelines. Changed MIME type to match the one selected for the Atom format. Numerous typographical fixes. We used to have two 'Introduction' sections. One of them was moved into the Abstract the other absorbed the Scope section. IPR and copyright notifications were added.

Rev 09 - 10Dec2003 - Added the section on SOAP enabled clients and servers.

Rev 08 - 01Dec2003 - Refactored the specification, merging the Introspection file into the feed format. Also dropped the distinction between the type of URI used to create new entries and the kind used to create comments. Dropped user preferences.

Rev 07 - 06Aug2003 - Removed the use of the RSD file for auto-discovery. Changed copyright until a final standards body is chosen. Changed query parameters for the search facet to all begin with atom- to avoid name collisions. Updated all the Entries to follow the 0.2 version. Changed the format of the search results and template file to a pure element based syntax.

Rev 06 - 24Jul2003 - Moved to PUT for updating Entries. Changed all the mime-types to application/x.atom+xml. Added template editing. Changed 'edit-entry' to 'create-entry' in the Introspection file to more accurately reflect its purpose.

Rev 05 - 17Jul2003 - Renamed everything Echo into Atom. Added version numbers in the Revision history. Changed all the mime-types to application/atom+xml.



Rev 04 - 15Jul2003 - Updated the RSD version used from 0.7 to 1.0. Change the method of deleting an Entry from POSTing <delete/> to using the HTTP DELETE verb. Also changed the query interface to GET instead of POST. Moved Introspection Discovery to be up under Introspection. Introduced the term 'facet' for the services listed in the Introspection file.

Rev 03 - 10Jul2003 - Added a link to the Wiki near the front of the document. Added a section on finding an Entry. Retrieving an Entry now broken out into its own section. Changed the HTTP status code for a successful editing of an Entry to 205.

Rev 02 - 7Jul2003 - Entries are no longer returned from POSTs, instead they are retrieved via GET. Cleaned up figure titles, as they are rendered poorly in HTML. All content-types have been changed to application/atom+xml.

Rev 01 - 5Jul2003 - Renamed from EchoAPI.html to follow the more commonly used format: [draft-gregorio-NN.html](#). Renamed all references to URL to URI. Broke out introspection into its own section. Added the Revision History section. Added more to the warning that the example URIs are not normative.



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

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

## 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 (2006). 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.