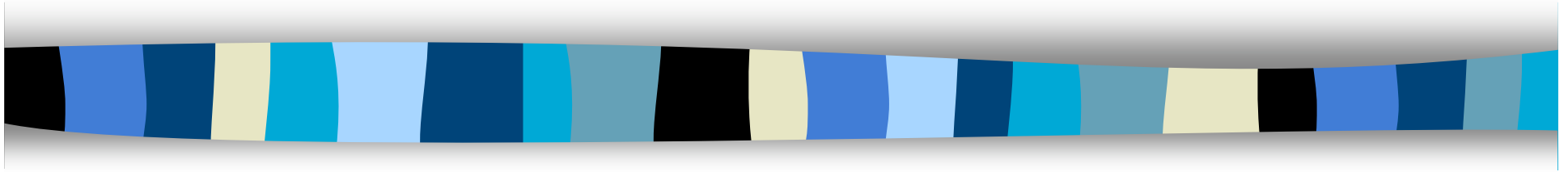


Atom Publishing Protocol



CSCI 7818

Elizabeth Fischer



History of Atom

- Initially created because of [dissatisfaction with RSS](#)
- June 2003, Sam Ruby set up a wiki to discuss [a well-formed log entry](#)
- Fully open-source effort
- APP standard is still in draft format



Atom Collections

- Central concept in protocol – a resource that contains a set of Member Entries
- Analogous to folders or directories in a file system.
- Each collection has a unique URI
- Adding an entry to a collection yields a new URI
- All operations done on URIs with GET, POST, PUT, DELETE



Use of HTTP methods

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



Functions available

- Discovery
- Listing
- Create
- Read
- Update
- Delete



Discovery

To discover the location of the collections exposed by an APP service, the client must locate and request an Introspection Document.

Client	Server
1.) GET Introspection	
----->	
2.) Introspection Doc	
<-----	

```
GET /servicedocument HTTP/1.1
Host: example.org
```



Example Discovery Response

```
<?xml version="1.0" encoding='utf-8'?>
<service xmlns="http://purl.org/atom/app#">
  <workspace title="Main Site" >
    <collection contents="entries" title="My Blog Entries"
      href="http://example.org/reilly/feed" />
    <collection contents="generic" title="Documents"
      href="http://example.org/reilly/pic" />
  </workspace>
  <workspace title="Side Bar Blog">
    <collection contents="entries" title="Entries"
      href="http://example.org/reilly/feed" />
    <collection contents="http://example.net/booklist"
      title="Books" href="http://example.org/reilly/books"
    />
  </workspace>
</service>
```



Listing

```
Client                                     Server
|                                         |
| 1.) GET to Collection URI              |
|----->|
|                                         |
| 2.) 200 OK, Atom Feed Doc             |
|<-----|
|                                         |
```

```
GET /blog/entries HTTP/1.1
Host: example.org
```




Response from listing request

```
<feed xmlns="http://www.w3.org/2005/Atom"
  xml:base="http://example.org/blog/entries">
  <id>http://example.org/blog/entries</id>
  <title>My Blog Entries</title>
  <updated>2006-08-12T13:40:03Z</updated>
  <link rel="self" href="/blog/entries" />
  <link href="http://blog.example.org" />
  <entry>
    <id>tag:example.org,2006:/blog/entries/1</id>
    <title>Atom-Powered Robots Run Amok</title>
    <link href="http://example.org/2003/12/13/atom03"/>
    <link rel="edit"
      href="http://example.org/blog/entries/1" />
    <updated>2006-08-12T13:40:03Z</updated>
    <author><name>James</name></author>
    <summary>Some text.</summary>
  </entry>
  <entry> ...
  </entry> ...
</feed>
```



Example of multi-page response

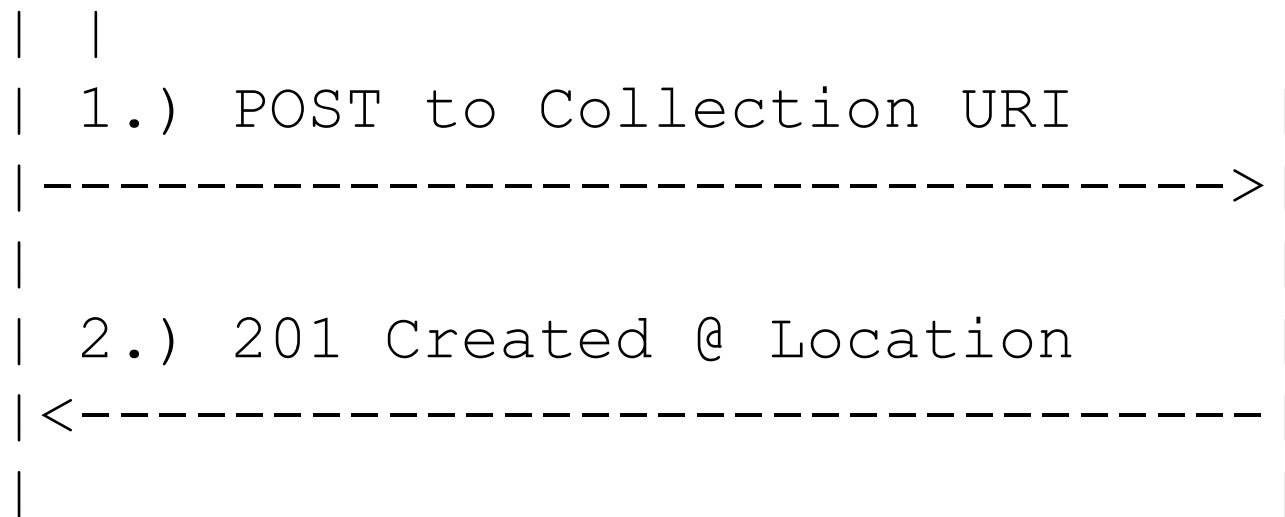
```
<feed xmlns="http://www.w3.org/2005/Atom"
xml:base="http://example.org/blog/entries
?page2">
<link rel="next" href="entries?page3" />
<link rel="previous" href="entries?page1"
/>
...
```

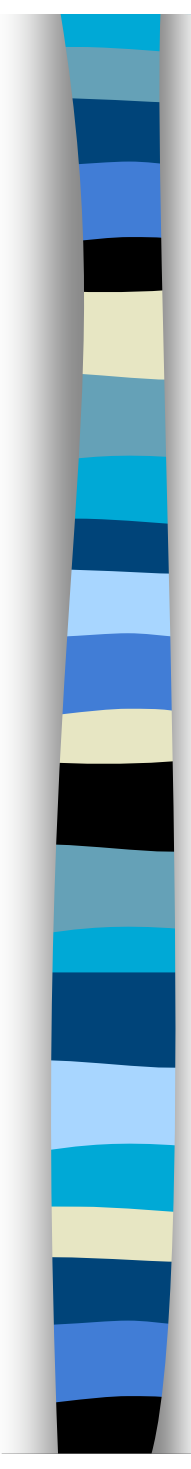


Create

Client

Server





HTTP/1.1 201 Created

Date: nnnn

Content-Type: application/atom+xml; charset=utf-8

Content-Location: /blog/entries/1

...

```
<?xml version="1.0" ?>
```

```
<entry xmlns="http://www.w3.org/2005/Atom">
```

```
<title>Atom-Powered Robots Run Amok</title>
```

```
<link
```

```
  href="http://example.org/2003/12/13/atom03"/>
```

```
<id>urn:uuid:1225c695-cfb8-4ebb-aaaa-
```

```
  80da344efa6a</id>
```

```
<updated>2003-12-13T18:30:02Z</updated>
```

```
<author><name>James</name></author>
```

```
<summary>Some text.</summary>
```

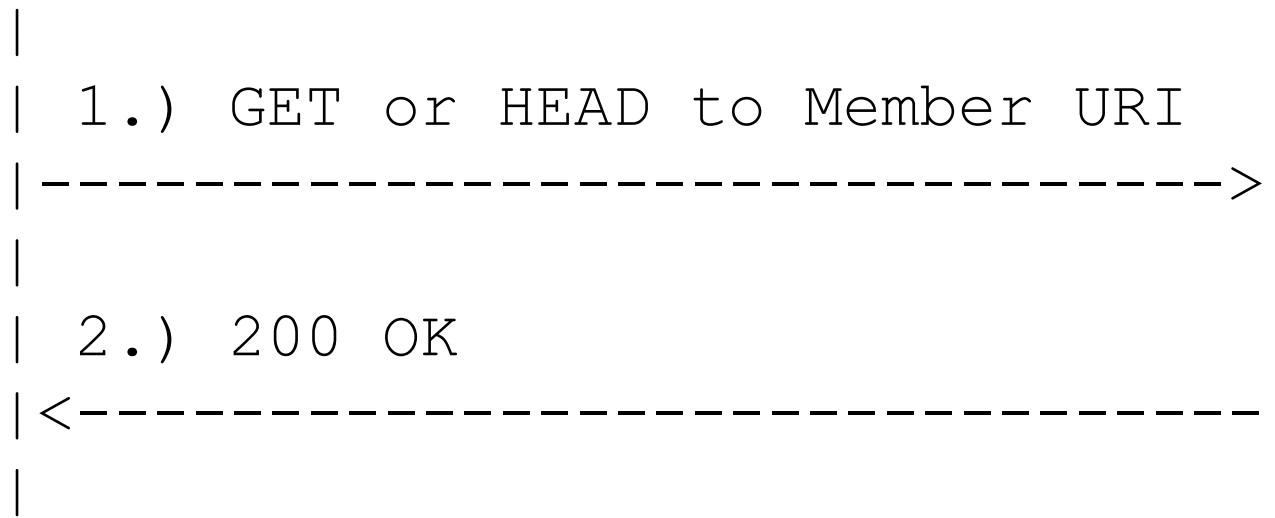
```
</entry>
```



Read

Client

Server





Update

Client

Server

|

|

| 1.) PUT to Member URI

|

|----->

|

|

|

| 2.) 200 OK

|

|<-----

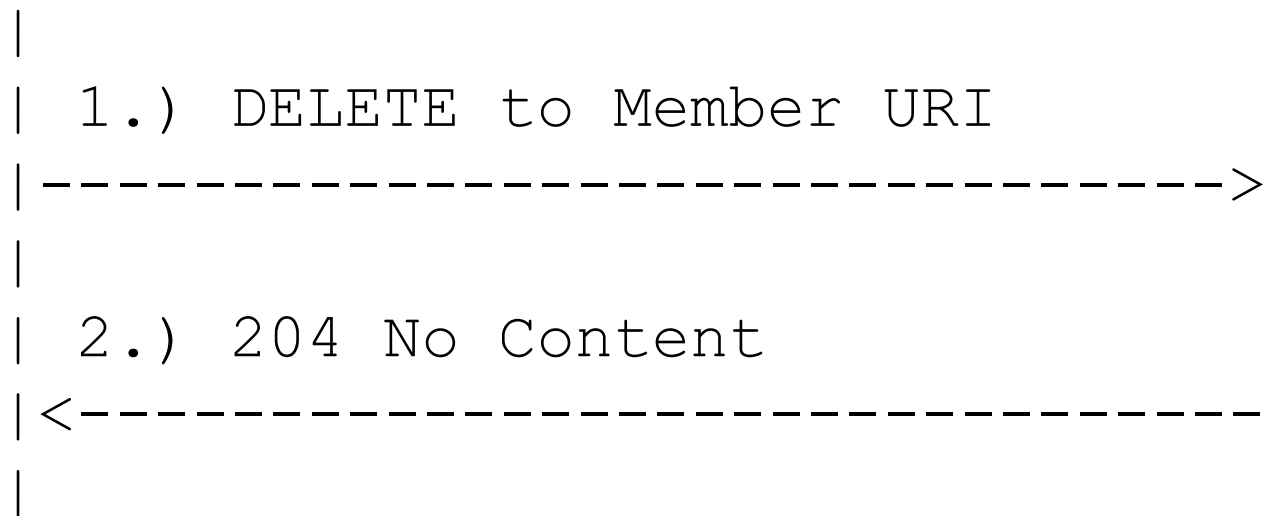
|



Delete

Client

Server





Expected Usage Patterns

- Case 1 – 1st time client encounters a collection
 - retrieve list of collection members
 - possibly retrieve some/all members
 - A non-partial GET on collection may result in document root element of collection document with 'next' attribute pointing to next member of collection
- Case 2 - Client needs a re-sync
 - Client does partial GET on collection document, supplying a Range header that begins from the end of its last GET.



Security

- All Atom publishing entities are expected to be protected by authentication to control posting and editing
- Expected to use either:
 - HTTP Digest Authentication [\[RFC2617\]](#)
 - [@@TBD@@ CGI Authentication ref]



Resources

- [Official Draft Atom Standard](#)
- [A More Readable Version](#)
- [Slightly more detailed tutorial](#)
- [Atom Wiki](#)