Introduction to Atom

The Atom Syndication Format and Atom Publishing Protocol

Nirav (Nero) Desai

Atom

- A pair of related standards
  - Atom Syndication Format – an XML language for web feeds
  - Atom Publishing Protocol (AtomPub or APP) – an HTTP-based protocol for creating and updating web resources

Atom and RSS

- Atom developed as an alternative to RSS
- Major differences
  - Standards
    - RSS has Blogger and MetaWebLog publishing protocols; Atom simply has AtomPub
  - Required content
    - Atom requires “author”, “uid” and “last update time”; RSS less restrictive
  - Content model

Content Model – Atom vs. RSS

- RSS payload may be plain text or escaped HTML with no way of specifying
  - `<title>This is bold.</title>`
  - `<title>This is &lt;b&gt;bold&lt;/b&gt;.</title>`

Content Model – Atom vs. RSS

- Atom provides means of clearly labeling the content type as plain text, escaped HTML, XHTML, XML, Base64-encoded binary
  - `<title type="text">This is bold.</title>`
  - `<title type="html">This is <b>bold</b>.</title>`

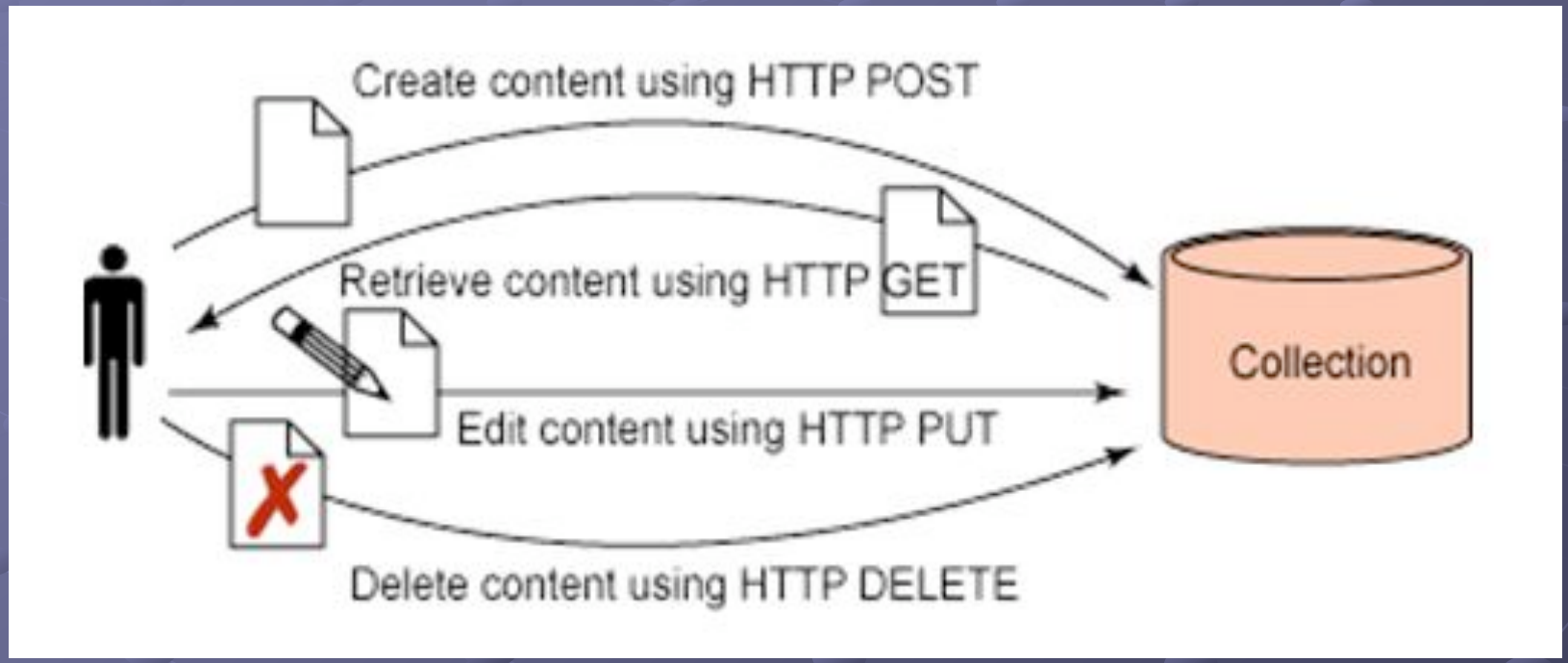
Example Atom feed

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
 <title>Example Feed</title>
 <subtitle>A subtitle.</subtitle>
 <link href="http://example.org/feed/" rel="self"/>
 <link href="http://example.org/"/>
 <updated>2003-12-13T18:30:02Z</updated>
 <author>
   <name>John Doe</name>
   <email>johndoe@example.com</email>
 </author>
 <id>urn:uuid:60a76c80-d399-11d9-b91C-0003939e0af6</id>
 <entry>
   <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>
   <summary>Some text.</summary>
 </entry>
</feed>
```

Atom Publishing Protocol

- Approach for creating and editing Web resources using basic HTTP operations (such as GET, PUT, POST)
- Operations are on Atom Feed and Entry documents that represent blog entries, podcasts, wiki pages, calendar entries, etc.

How it works – high level



Finding collections

- Service document
  - XML format that tells the client what collections are available and types of resources they can contain
  - GET /servicedocument HTTP/1.1
    Host: example.org

Sample Service Document

```
HTTP/1.1 200 OK
Date: ...
Content-Type: application/atomserv+xml; charset=utf-8
Content-Length: nnn
<service xmlns="..." xmlns:atom="http://www.w3.org/2005/Atom">
  <workspace>
    <atom:title>My Weblog</atom:title>
    <collection href="http://www.example.org/blog/entries">
      <atom:title>Entries</atom:title>
      <accept>entry</accept>
    </collection>
    <collection href="http://www.example.org/blog/photos">
      <atom:title>Photos</atom:title>
      <accept>image/*</accept>
    </collection>
  </workspace>
</service>
```

Listing entries of collection

- Once you have URI of collection of interest from service document, you can list its contents
  - GET /blog/entries HTTP/1.1
    Host: example.org
- Returns an Atom Feed Document with entries for each resource in the collection

Posting an entry

- You can also POST to the URI of a collection
- Must contain all required elements even though server may override some
- HTTP response provides
  - Status of the request
    - Ex. HTTP/1.1 201 Created
  - URI of created resource
    - Ex. /blog/entries/23

Edit/Deleting entries

- Use GET followed by PUT to edit an entry
- Can use “If-Match” and/or “If-Unmodified-Since” HTTP header fields to avoid overwriting changes
- Delete entries by issuing a DELETE on the entry’s URI
  - DELETE /blog/entries/23 HTTP/1.1
    Host: example.org

Media resources

- Can add media resources such as photos, documents, audio, etc. to an AtomPub collection
- Server will create Atom entry document linked to the resource called a *media-link entry*

Create media-link entry

- To create a media-link entry, issue POST to collection URI with representation of the media resource
  - POST /blog/photos HTTP/1.1
    Host: example.org
    Content-Type: image/png
    Content-Length: nnn
    Slug: Niagara Falls sunset
    {binary image data}

Media-Link Response

```
…
Content-Location: /blog/photos/Niagara_Falls_sunset
…
Last-Modified: Wed, Oct 29 2008 14:11:04 GMT
<?xml version="1.0"?>
<entry xmlns="http://www.w3.org/2005/Atom">
  <id>…</id>
  <title>Niagara Falls sunset</title>
  <link rel="edit-media" type="image.png"
   href="http://example.org/blog/photos/Niagara_Falls_sunse
   t?media" />
  <updated>2008-10-29T14:11:04Z</updated>
  <author><name>Nirav</name></author>
  <content type="image/png"
   src="http://blog.example.org/photos/Niagara_Falls_sunset
   " />
</entry>
```

Edit media resources

- Editing media resources uses same paradigm as editing Atom entries
  - GET editable version of resource
  - Make modifications
  - PUT it back