

Fulfilling the Hypermedia Constraint Via HTTP OPTIONS, the HTTP Vocabulary In RDF, And Link Headers

Second International Workshop on RESTful Design
WWW 2011, 28 March 2011, Hyderabad, India

Thomas Steiner, [@tomayac](#)
Google Germany GmbH, Universitat Politècnica de Catalunya

Jan Algermissen, [@algermissen](#)
NORD Software Consulting

Sitting on an HTTP API is not RESTing.



Slouching on a hypermedia API is RESTing.



aka. hypermedia

"When I say ~~hypertext~~, I mean the simultaneous presentation of information and controls such that the information becomes the affordance through which the user (or automaton) obtains choices and selects actions."



Simultaneous presentation of information and controls.



Information



State



Controls



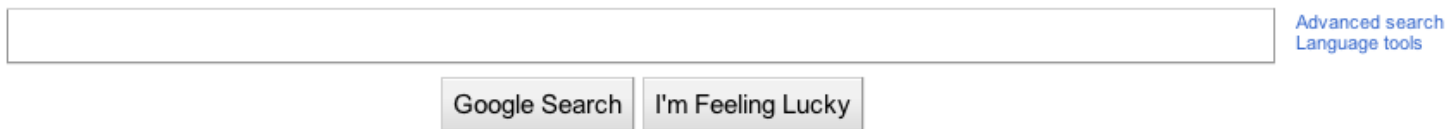
```
<atom:link rel="search"
type="application/opensearchdescription+xml"
href="http://thisweekinrest.wordpress.com/osd.xml"
title="This week in REST" />
<atom:link rel='hub'
href='http://thisweekinrest.wordpress.com/?pushpress=hub' />
```

```
atomLink = element atom:link {
  atomCommonAttributes,
  attribute href { atomUri },
  attribute rel { atomNCName | atomUri }?,
  attribute type { atomMediaType }?,
  attribute hreflang { atomLanguageTag }?,
  attribute title { text }?,
  attribute length { text }?,
  undefinedContent }
```

- Registry of link relations:
<http://tools.ietf.org/html/rfc4287#page-31>.
- Can be an IRI for custom link relations.

```
link: [  
  - {  
    rel: "related",  
    type: "application/atom+xml",  
    href: "http://gdata.youtube.com/feeds/base/users/google"  
  },  
  - {  
    rel: "alternate",  
    type: "text/html",  
    href: "http://www.youtube.com/profile\_videos?user=Google"  
  },  
]
```

- Based on Atom Syndication protocol.
- XML and JSON serialization.



```
<form
action="/search" id=tsf method=GET
name=f onsubmit="return
q.value!=' '" role=search
style="display:block;margin:0
auto;background:none">
```

- Can use the XHTML @role attribute.

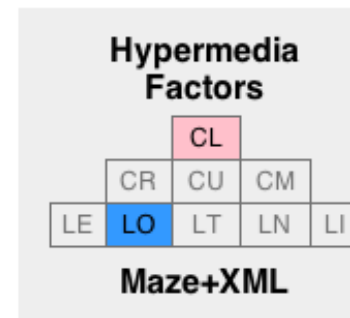
Media Types

[Home](#) » [Maze+XML](#)

Maze+XML - Maze Document

Description

The Maze+XML media type is an XML data format for sharing maze state information between clients and servers. It can be used to implement simple mazes, adventure games, and other related data.



NOTE:

The current version exposes only the basics of navigating a simple 2-D maze. Future additions to the media type will increase functionality w/o breaking support for the current feature set.

MIME Types

- [application/vnd.amundsen.maze+xml](#) approved

- Versioning happens via the media type.
- Can use the vendor (vnd.) or personal (prs.) tree.

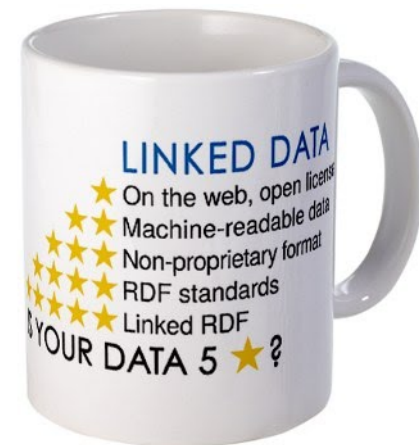
```
$ curl -i http://search.example.org/page8
HTTP/1.1 200 OK
Date: Fri, 25 Mar 2011 08:24:20 GMT
Server: Apache
Last-Modified: Fri, 18 Feb 2011 08:30:51 GMT
ETag: "d1caaa08-8aa-4d5e2e3b"
Link: <http://search.example.org/page7>; rel="previous";
title="previous results"
Link: <http://search.example.org/page9>; rel="next";
title="next results"
Content-Length: 2218
Content-Type: text/html
```

- Semantically equivalent to HTML link, atom:link.
- Web Linking RFC updates Atom link relations:
<http://tools.ietf.org/html/rfc5988#section-6.2.2>.

```
$ curl -i -X OPTIONS http://www.upc.edu/  
HTTP/1.1 200 OK  
Allow: GET, HEAD, POST, PUT, DELETE, OPTIONS, TRACE
```

- On HTTP OPTIONS from RFC2616 (<http://tools.ietf.org/html/rfc2616#section-9.2>):
- The OPTIONS method represents a request for information about the communication options available on the request/response chain.
- A 200 response SHOULD include any **header fields** that indicate optional features implemented by the server and applicable to that resource (e.g., Allow).
- The **response body**, if any, SHOULD include information about the communication options.
- OPTIONS is an optional method. Servers are not required to support it.

- Atom, Google Data, HTML Link, XHTML @role, Link headers can use extension relation types via URIs[†].
- With Linked Data, when you have some of it, you can find other, related, data.
- Tim Berners-Lee's Linked Data principles (<http://www.w3.org/DesignIssues/LinkedData.html>):
 1. Use URIs as names for things.
 2. Use HTTP URIs so that people can look up those names.
 3. When someone looks up a URI, provide useful information using the standards (RDF*, SPARQL).
 4. Include links to other URIs, so that they can discover more things.



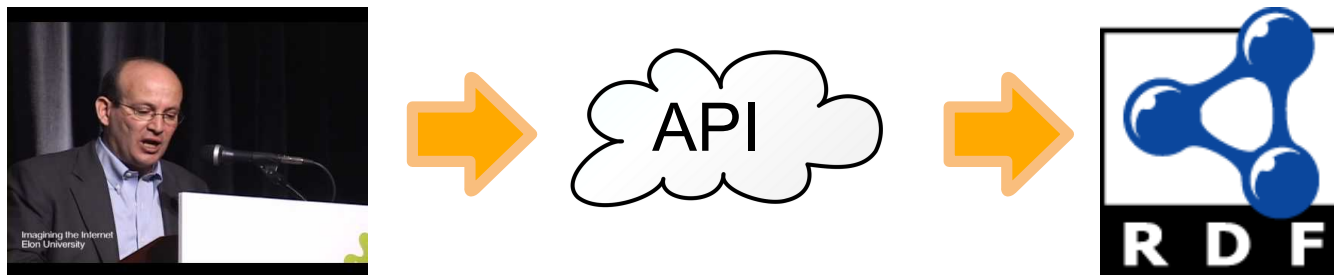
[†]For Atom even IRIs, limiting ourselves to URIs here.

- HTTP Vocabulary in RDF intended to record HTTP(S) request and response messages, including the various headers (<http://www.w3.org/TR/HTTP-in-RDF10/>).

```
$ curl -H "Accept: text/turtle" http://dbpedia.org
```

```
_ :req a http:Request ;  
  http:httpVersion "1.1" ;  
  http:methodName "GET" ;  
  http:mthd <http://www.w3.org/2008/http-methods#GET> ;  
  http:headers (  
    [ http:fieldName "Host" ;  
      http:fieldValue "dbpedia.org" ;  
      http:hdrName <[...]w3.org/2008/http-header#host> ]  
    [ http:fieldName "Accept" ;  
      http:fieldValue "text/turtle" ;  
      http:hdrName <[...]w3.org/2008/http-header#accept> ]  
  ) .
```

1. Link headers to transparently inject data into an HTTP response without touching the body.
 2. HTTP OPTIONS as a means of requesting communication options information.
 3. HTTP Vocabulary in RDF for modeling HTTP communication.
- Real-world example: API for annotating YouTube videos with RDF:

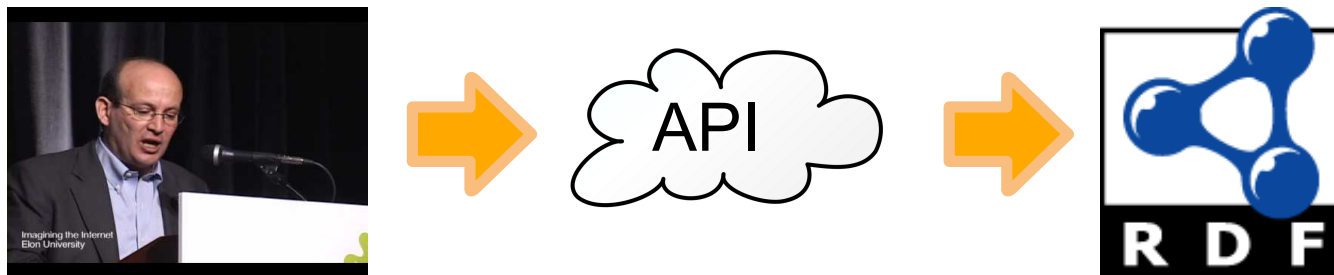


In: YouTube video ID

Out: text/turtle, application/rdf+xml

- User agents can GET an RDF representation of a video.
- Generic string with meta information "this is a YouTube ID" as input by API definition.
- Generic output media types by API definition.
- RDF output document can be manually corrected via PUT, or DELETED.

```
$ curl -i -X GET -H "Accept: text/turtle" http://  
api.example.org/youtube/videos/o4T9_MduMEQ
```



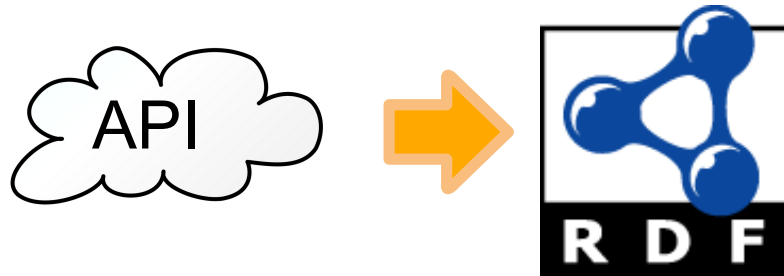
In: YouTube video ID

Out: text/turtle, application/rdf+xml

- Nice to document input constraints (YouTube ID).



- Nice to document next steps based on the output.



- Can use a combination of OPTIONS, Link headers, and HTTP Vocabulary in RDF to do this **transparently**.
 - Send next steps (media controls) as Link headers.
 - Document constraints for each request via OPTIONS.

Bringing it all together



```
$ curl -i -X OPTIONS http://api.example.org/youtube/videos
HTTP/1.1 200 OK
Content-Type: text/turtle; charset=utf-8
Link: <http://api.example.org/youtube/videos/{video_id}>; rel="related"
Allow: GET, PUT, DELETE, HEAD, OPTIONS, PATCH
Content-Length: xx
```

```
_ :req_1 a http:Request ;
  http:httpVersion "1.1" ;
  http:methodName "GET" ;
  ...

_ :req_2 a http:Request ;
  http:httpVersion "1.1" ;
  http:methodName "PUT" ;
  ...

_ :req_3 a http:Request ;
  http:httpVersion "1.1" ;
  http:methodName "DELETE" ;
  ...
```

Bringing it all together



```
$ curl -i -X OPTIONS http://api.example.org/youtube/videos
...
Link: <http://api.example.org/youtube/videos/{video_id}>; rel="related"
Allow: GET, HEAD, PUT, DELETE, OPTIONS, PATCH
...
```

```
_ :req_1 a http:Request ;
_ http:httpVersion "1.1" ;
_ http:methodName "GET" ;
_ http:mthd <http://www.w3.org/2008/http-methods#GET> ;
_ exHttp:prefixPath "/youtube/" ;
_ exHttp:suffixPath [
  a api:uriTemplate ;
  a yt:videoid ;
] ;
_ http:headers (
  [ http:fieldName "Host" ;
    http:fieldValue "api.example.org" ;
    http:hdrName <http://www.w3.org/2008/http-header#host>
  ]
) .
```

```
_ :req_1 a http:Request ;
  http:httpVersion "1.1" ;
  http:methodName "GET" ;
  http:mthd <http://www.w3.org/2008/http-methods#GET> ;
  exHttp:prefixPath "/youtube/" ;
  exHttp:suffixPath [
    a api:uriTemplate ;
    a yt:videoid ;
  ] ;
  http:headers (
    [ http:fieldName "Host" ;
      http:fieldValue "api.example.org" ;
      http:hdrName <http://www.w3.org/2008/http-header#host>
    ]
  ) .
```

exHttp: Suggested extension of the HTTP Vocabulary in RDF for splitting `http:abs_path (/foo/bar)` in `exHttp:prefixPath (/foo)` and `exHttp:suffixPath (/bar)`.

```
_ :req_1 a http:Request ;
  http:httpVersion "1.1" ;
  http:methodName "GET" ;
  http:mthd <http://www.w3.org/2008/http-methods#GET> ;
  exHttp:prefixPath "/youtube/" ;
  exHttp:suffixPath [
    a api:uriTemplate ;
    a yt:videoid ;
  ] ;
  http:headers (
    [ http:fieldName "Host" ;
      http:fieldValue "api.example.org" ;
      http:hdrName <http://www.w3.org/2008/http-header#host>
    ]
  ) .
```

`api:uriTemplate`: Borrowed from <http://purl.org/linked-data/api/vocab#> in order to model URI templates.

Bringing it all together



```
_ :req_1 a http:Request ;
  http:httpVersion "1.1" ;
  http:methodName "GET" ;
  http:mthd <http://www.w3.org/2008/http-methods#GET> ;
  exHttp:prefixPath "/youtube/" ;
  exHttp:suffixPath [
    a api:uriTemplate ;
    a yt:videoid ;
  ] ;
  http:headers (
    [ http:fieldName "Host" ;
      http:fieldValue "api.example.org" ;
      http:hdrName <http://www.w3.org/2008/http-header#host>
    ]
  ) .
```

[yt:videoid](http://gdata.youtube.com/schemas/2007/videoid): Defines the URI template to be a YouTube video ID (<http://gdata.youtube.com/schemas/2007/videoid>).

- The approach might encourage tight coupling. Evil? Depends. Might be OK if user agents evaluate their OPTIONS at runtime.
- HTTP Vocabulary in RDF might be too verbose, but it's a standard, and usually a bad standard is better than no standard.
- Total freedom (=chaos?) with regards to the RDF triples. Fair enough. However, extension relation types via URLs require human interaction as well.



Thanks for your attention



-
- Thomas Steiner
 - tomac@google.com
 - [@tomayac](#)

Image credits



Lawn chair - <http://www.cp-objekt.de/hundemoebel/Klappstuhl-Montana-st-426710-gr.jpg>

Poäng - <http://www.ikea.com/au/en/catalog/products/10157948>

Roy Fielding - http://upload.wikimedia.org/wikipedia/commons/thumb/c/cd/Roy_Fielding.jpg/450px-Roy_Fielding.jpg

Direction sign - <http://www.myparkingsign.com/img/lg/X/One-Direction-Arrow-Sign-X-W1-6L.gif>

Remote control - http://www.faqs.org/photo-dict/photofiles/list/824/1251remote_control.jpg

LCD TV - http://www.slashgear.com/wp-content/uploads/2009/02/philips_netv-480x370.jpg

Fridge - http://modculture.typepad.com/photos/uncategorized/2008/06/13/smeg_mini.jpg

Linked Data mug - http://www.w3.org/DesignIssues/diagrams/lod/480759174v0_350x350_Back.jpg

Cloud - http://www.clipart.clipartist.net/openclipart/clipart/signs_and_symbols/weather/w_cloud_xxl.png

Dr. Evil - http://upload.wikimedia.org/wikipedia/en/1/16/Dreuil_million_dollars.jpg