Network Working Group	M. Douglass
Internet-Draft	Spherical Cow Group
Updates: 5988 (if approved)	September 5, 2017
Intended status: Standards Track	
Expires: March 9, 2018	

Calendar subscription upgrades

draft-douglass-subscription-upgrade-02

Abstract

This specification introduces an approach to allow subscribers to calendar feeds to upgrade to a more performant protocol.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Currently clients subscribe to calendar feeds as an ics file which is often published as a resource accessible using the unofficial 'webcal' scheme.

The only available option for updating that resource is the usual HTTP polling of cached resources using Etags.

There is the usual tension between clients wishing to see a timely response to changes and servers not wishing to be overloaded by frequent requests for possibly large amounts of data.

This specification introduces an approach whereby clients can discover a more performant access method. Given the location of the resource as an ics file, the client can perfom an OPTIONS request on the resource and inspect the returned headers which will offer a number of alternative access methods.

Given that many clients already support CalDAV this provides an easy upgrade path for those clients. CalDAV and DAV subsets are specified here to allow lighter weight implementations.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Enhanced GET

2.1. Introduction

This is a lightweight protocol which allows simple clients to efficiently discover and download changes in the targeted resource.

It has many similarities to WebDAV sync and for a server could be implemented as an extension of the

specification.

In this protocol the Etag is used as the sync change token. By adding the If-None-Match header field to the vary header field we can ensure intermediate caching proxies will be able to cache different versions of the data.

The resource is treated as a set of individual events each of which may be updated or deleted separately. The client will first fetch the entire ics file. On subsequent requests it uses the Prefer header with a value of "return=minimal" to indicate that it wants a set of changes since the last fetch.

2.2. Deletions

When an entity (VEVENT, VTODO or other valid top-level component) is deleted from the source data the server needs to be able to inform a client of the deletion. This specification introduces a new value for the STATUS property of DELETED.

On the first conditional GET after the entity has been deleted a skeleton, but valid, entity will be returned with STATUS: DELETED. The receiving client is free to remove the entity or update it's STATUS property.

On subsequent conditional fetches the entity will not be returned.

2.3. Examples

```
>> Request <<

GET /events.ics HTTP/1.1
Host: example.com
Accept: text/calendar

>> Response <<

HTTP/1.1 200 OK
Content-Length: xxxx
ETag: "1234"                    current ETag (for conditional GET)
Vary: Prefer, If-None-Match             so caching proxy can key off of client?s ETag (sync token) and preference

 BEGIN:VCALENDAR:
 ?  /* full feed */
 END:VCALENDAR
```

This is an example of the initial request and response from a server that supports the extended GET protocol.

```
>> Request <<

GET /events.ics HTTP/1.1
Host: example.com
Accept: text/calendar; q=0.5, component=VPATCH, text/calendar;
If-None-Match: ?1234?             conditional request
Prefer: return=minimal

>> Response <<

HTTP/1.1 304 Not Modified
Content-Length: 0
ETag: ?1234?
```

```
Vary: Prefer, If-None-Match
```

This is an example of the subsequent request and response when no changes have occurred. The Accept header field indicates that a VPATCH format is most desirable but simple text/calendar is acceptable.

```
>> Request <<

GET /events.ics HTTP/1.1
Host: example.com
Accept: text/calendar; q=0.5, component=VPATCH, text/calendar;
If-None-Match: "1234"            conditional request
Prefer: return=minimal

>> Response <<

HTTP/1.1 304 Not Modified
Content-Length: 0
ETag: "1234"
Vary: Prefer, If-None-Match
```

This is an example of the subsequent request and response when no changes have occurred. The Accept header field indicates that a VPATCH format is most desirable but simple text/calendar is acceptable.

```
>> Request <<

GET /events.ics HTTP/1.1
Host: example.com
Accept: text/calendar; q=0.5, component=VPATCH, text/calendar;
If-None-Match: "1234"            conditional request
Prefer: return=minimal

>> Response <<

HTTP/1.1 200 OK
Content-Type: text/calendar
Content-Length: xxxx
ETag: "5678"                  current ETag (for conditional GET)
Preference-Applied: return=minimal    signals to client that stream is changes  only
Vary: Prefer, If-None-Match           so caching proxy can key off of client?s  ETag (sync token) and preference

BEGIN:VCALENDAR:
...  only new/changed events
...  when not returning VPATCH, deleted events have STATUS:DELETED
END:VCALENDAR
```

This is an example of the subsequent request and response when changes have occurred and the server can create the minimal format.

```
>> Request <<

GET /events.ics HTTP/1.1
Host: example.com
```

```
Accept: text/calendar; q=0.5, component=VPATCH, text/calendar;
If-None-Match: "1234"            conditional request
Prefer: return=minimal

>> Response <<

HTTP/1.1 200 OK
Content-Type: text/calendar
Content-Length: xxxx
ETag: "5678"                  current ETag (for conditional GET)
Vary: Prefer, If-None-Match          so caching proxy can key off of client?s  ETag (sync token) and preference

BEGIN:VCALENDAR:
...  full set of data
END:VCALENDAR
```

This is an example of the subsequent request and response when changes have occurred and the server cannot create the minimal format - perhaps because of an old or invalid token. Note there is no Preference-Applied header field.

3. Changes to the iCalendar specifications

This specification does not require any changes to [RFC5545] or its extensions. However it does introduce the use of some properties to provide more information about the resource, for example the time range it covers.

4. Discovering alternative access methods

The advertising of other access points is achieved through the use of the LINK header as defined in [RFC5988]. New link relation types are defined in this specification - each being associated with a protocol or protocol subset.

These LINK headers will be delivered when a client carries out an OPTIONS request targeting the URL of the resource.

5. Link relation subscribe-caldav

This specifies an access point which is a full implementation of caldav but requires no authentication. The end point allows the full range of reports as defined by the CalDAV specification.

The client MUST follow the specification to determine exactly what operations are allowed on the access point - for example to determine if sync-report is supported.

The URL MAY include some form of token to allow write access to the targeted collection. The client must check it's permissions to determine whether or not it has been granted write access.

6. Link relation subscribe-caldav-auth

This specifies an access point which is a full implementation of caldav and requires authentication. This may allow read-write access to the resource.

The client MUST follow the specification to determine exactly what operations are allowed on the access point - for example to determine if sync-report is supported.

7. Link relation subscribe-webdav-sync

This specifies an access point which supports only webdav sync.

This allows the client to issue a sync-report on the resource to obtain updates.

NOTE: say something about initial startup - use ics to populate? Initial token?

The client MUST follow that specification.

8. Link relation subscribe-enhanced-get

This specifies an access point which supports something new.

The client MUST follow that specification.

9. Security Considerations

Applications using these properties need to be aware of the risks entailed in using the URIs provided as values. See [RFC3986] for a discussion of the security considerations relating to URIs.

10. Privacy Considerations

Properties with a "URI" value type can expose their users to privacy leaks as any network access of the URI data can be tracked. Clients SHOULD NOT automatically download data referenced by the URI without explicit instruction from users. This specification does not introduce any additional privacy concerns beyond those described in [RFC5545].

11. IANA Considerations

11.1. Link Relation Registrations

This document defines the following new iCalendar properties to be added to the registry defined in Section 8.2.3 of [RFC5545]:

Relation Name	Description	Reference
subscribe-caldav	Current	RFCXXXX, Section 5
subscribe-caldav_auth	Current	RFCXXXX, Section 6
subscribe-webdav-sync	Current	RFCXXXX, Section 7
subscribe-enhanced_get	Current	RFCXXXX, Section 8

12. Acknowledgements

The author would also like to thank the members of the Calendaring and Scheduling Consortium Calendar Sharing technical committee and the following individuals for contributing their ideas and support:

...

The authors would also like to thank the Calendaring and Scheduling Consortium for advice with this specification.

13. Normative References

[I-D.ietf-calext-extensions] Daboo, C., "New Properties for iCalendar", Internet-Draft draft-ietf-calext-extensions-05, August 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, DOI 10.17487/RFC2434, October 1998.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004.

[RFC3986] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005.

[RFC4589] Schulzrinne, H. and H. Tschofenig, "Location Types Registry", RFC 4589, DOI 10.17487/RFC4589, July 2006.

[RFC5545] Desruisseaux, B., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 5545, DOI 10.17487/RFC5545, September 2009.

[RFC5546] Daboo, C., "iCalendar Transport-Independent Interoperability Protocol (iTIP)", RFC 5546, DOI 10.17487/RFC5546, December 2009.

[RFC5988] Nottingham, M., "Web Linking", RFC 5988, DOI 10.17487/RFC5988, October 2010.

[W3C.REC-xml-20060816] Bray, T., Paoli, J., Sperberg-McQueen, M., Maler, E. and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", World Wide Web Consortium Recommendation REC-xml-20060816, August 2006.

Appendix A. Open issues

restype values:

Need to determine what if any registry of resource types already exists and use that.

Appendix B. Change log

v01 2017-07-28 MD

- Examples
- More text for extended get. Talk about deletions.

v01 2017-02-17 MD

- Add text about OPTIONS
- Add text abut read/write CalDAV

v00 2017-02-15 MD

- First pass

Author's Address

Michael Douglass
Spherical Cow Group
226 3rd Street
Troy, NY 12180
USA
EMail: mdouglass@sphericalcowgroup.com
URI: http://sphericalcowgroup.com