

Bug [32016](#)

Summary:	CalDAV synchronisation fails in Thunderbird Lightning		
Product:	OX App Suite	Reporter:	Moritz Schlarb <moschlar@metalabs.de>
Component:	DAV	Assignee:	Friedrich, Tobias <tobias.friedrich@open-xchange.com>
Status:	RESOLVED INVALID	QA Contact:	Quality Assurance <qa@open-xchange.com>
Severity:	3		
Priority:	P5	CC:	bmesa@gmx.de, liob@metalabs.de
Version:	7.4.2		
Target Milestone:	later		
Hardware:	PC/Mac		
OS:	---		
URL:	https://bugzilla.mozilla.org/show_bug.cgi?id=995740		
See Also:	https://bugzilla.mozilla.org/show_bug.cgi?id=995740		
Whiteboard:			
Server OS:	Debian 7	Client OS:	Linux
Client Software:	---	Reproducibility:	---
Ticket number:		Customer Name:	
Patch ID:		Internal verification:	---
Attachments:	Lightning log Apache log		

Moritz Schlarb 2014-04-13 13:38:35 CEST

[Description](#)

Created [attachment 17130](#) [\[details\]](#)
Lightning log

I do not know whether this is more an issue for OX or for lightning, so I create a bug report on each project.
Please tell me what steps I could take to debug the issue.

Steps to Reproduce:

I use the Open-Xchange Appsuite groupware and I have an already filled calendar there. I take the given CalDAV-URI for the calendar and add it as a new network calendar with CalDAV in Lightning.

Actual Results:

The Calendar gets marked with an exclamation sign as "momentarily not available".
The log entries from Thunderbird and Lightning, as well as from the Webserver log are attached. If needed, I can also supply a wireshark packet dump of the respective packets.

Expected Results:

When I create a new, empty calendar in Open-Xchange and add it to Lightning, it works as expected. I can create/change/delete entries on both sides and syncing works. So I assume that the issue has something to do with the specific contents of my primary calendar.

Moritz Schlarb 2014-04-13 13:38:57 CEST

[Comment 1](#)

Created [attachment 17131](#) [\[details\]](#)
Apache log

Friedrich, Tobias 2014-04-14 08:58:55 CEST

[Comment 2](#)

From [attachment #17130](#) [\[details\]](#), it looks like the client applies some URL decoding for the resource URIs, which leads to two URIs being wrong in the following calendar-multiget request:

Server sends:

```

> <D:multistatus>
>   <D:sync-token>1397315195113</D:sync-token>
>   [...]
>   <D:response>
>     <D:href>/caldav/29/AAAAAKjBIOfq2tdEnpbRdQf%2Fx9yEASAA.ics</D:href>
>     <D:propstat>
>       <D:prop>
>         <D:getcontenttype>text/calendar</D:getcontenttype>
>         <D:getetag>http://www.open-xchange.com/etags/3331-1361055684000</D:getetag>
>       </D:prop>
>       <D:status>HTTP/1.1 200 OK</D:status>
>     </D:propstat>
>   </D:response>
> </D:response>

```

```

> <D:href>/caldav/29/AAAAAKjBiofq2tdEnpbRdQf%2Fx9wkASAA.ics</D:href>
> <D:propstat>
> <D:prop>
> <D:getcontenttype>text/calendar</D:getcontenttype>
> <D:getetag>http://www.open-xchange.com/etags/3332-1355693324000</D:getetag>
> </D:prop>
> <D:status>HTTP/1.1 200 OK</D:status>
> </D:propstat>
> </D:response>
> [...]
> </D:multistatus>

```

Client sends:

```

> <C:calendar-multiget xmlns:D="DAV:" xmlns:C="urn:ietf:params:xml:ns:caldav">
> <D:prop>
> <D:getetag/>
> <C:calendar-data/>
> </D:prop>
> [...]
> <D:href>/caldav/29/AAAAAKjBiofq2tdEnpbRdQf/x9wkASAA.ics</D:href>
> <D:href>/caldav/29/AAAAAKjBiofq2tdEnpbRdQf/x9yEASAA.ics</D:href>
> [...]
> </C:calendar-multiget>

```

The client would need to use the URIs as sent from the server instead, i.e.:

```

> <C:calendar-multiget xmlns:D="DAV:" xmlns:C="urn:ietf:params:xml:ns:caldav">
> <D:prop>
> <D:getetag/>
> <C:calendar-data/>
> </D:prop>
> [...]
> <D:href>/caldav/29/AAAAAKjBiofq2tdEnpbRdQf%2Fx9yEASAA.ics</D:href>
> <D:href>/caldav/29/AAAAAKjBiofq2tdEnpbRdQf%2Fx9wkASAA.ics</D:href>
> [...]
> </C:calendar-multiget>

```

As a workaround, you could try to identify the appointments with the UIDs "AAAAAKjBiofq2tdEnpbRdQf%2Fx9yEASAA" and "AAAAAKjBiofq2tdEnpbRdQf%2Fx9wkASAA" and move them to another calendar folder using the web interface, then try to synchronize again.

Moritz Schlarb 2014-04-14 12:09:09 CEST

[Comment 3](#)

Thank you very much for the quick response!

Is there any way to find out how those UUIDs are generated or where they are stored so I have a way of finding which of the appointments are the bad ones? I tried looking through our OX sql database, but couldn't find them there anywhere.

Moritz Schlarb 2014-04-14 12:45:28 CEST

[Comment 4](#)

(In reply to Moritz Schlarb from [comment #3](#))

> Is there any way to find out how those UUIDs are generated or where they are stored so I have a way of finding which
> of the appointments are the bad ones? I tried looking through our OX sql database, but couldn't find them there
> anywhere.

Got it, I used the HTTP API! ;)

Thanks again.

[Legal](#) | [Privacy Policy](#) | [Contact Us](#)