XEP-0129: WebDAV File Transfers

Peter Saint-Andre
mailto:stpeter@stpeter.im
xmpp:stpeter@jabber.org
https://stpeter.im/

Dave Smith
mailto:dizzyd@jabber.org
xmpp:dizzyd@jabber.org

Status	Type	Short Name
Deferred	Standards Track	TO BE ASSIGNED

This document specifies a method for completing file transfers between XMPP entities using WebDAV.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 by the XMPP Standards Foundation (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy> or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1 Introduction

SI File Transfer (XEP-0096) [1] defines mechanisms for transferring files between Jabber users, and defines the preferred approach for file transfers in Jabber applications. Unfortunately, the mechanisms defined therein require that both the sender and recipient be online at the same time. However, sometimes it is desirable for the sender to initiate a file transfer while the recipient is offline. One way to make this possible is for the sender to upload the file to a unique URL, then inform the recipient of the URL. The sender could do this by uploading the file to their own web server, but not everyone has their own web server. Fortunately, there is a well-defined protocol for such file management operations: a set of HTTP extensions known as WebDAV and defined in RFC 2518 [2] (see also the revision-in-progress, rfc2518bis [3]).

The use case in which the recipient is offline is the main motivation for this document. Another WebDAV use case presents itself in environments that use, or even require, WebDAV for file transfers using other protocols (e.g., files attached to email messages). The usual rationale for such deployments is virus-checking: the file is put onto the WebDAV server (either by an end-user or a script that, for example, strips attached files off email messages) and then checked for viruses; only after the virus check successfully completes is the recipient allowed to retrieve the file. A further benefit of such deployments is that it enables the sender to provide the file to multiple recipients. Thus the approach defined herein provides the added benefit of being usable in generic WebDAV environments as well.

2 Requirements

This document addresses the following requirements:

1. Enable file transfers when recipient is offline.

2. Use WebDAV for file puts and gets.

3 Terminology

This document inherits terms from RFC 2518 [4], RFC 2616 [5], and RFC 2617 [6].

---

[1] XEP-0096: SI File Transfer <https://xmpp.org/extensions/xep-0096.html>.

[2] RFC 2518: HTTP Extensions for Distributed Authoring -- WEBDAV <http://tools.ietf.org/html/rfc2518>.

[3] RFC 2518: HTTP Extensions for Distributed Authoring -- WebDAV <http://tools.ietf.org/html/draft-ietf-webdav-rfc2518bis>.

[4] RFC 2518: HTTP Extensions for Distributed Authoring -- WEBDAV <http://tools.ietf.org/html/rfc2518>.

[5] RFC 2616: Hypertext Transport Protocol -- HTTP/1.1 <http://tools.ietf.org/html/rfc2616>.

[6] RFC 2617: HTTP Authentication: Basic and Digest Access Authentication <http://tools.ietf.org/html/rfc2617>.

4 Protocol Flow

The client SHOULD attempt to PUT the file on the server. [7] The PUT request MUST include an "If-None-Match" header as well as an "Authorization" header that specifies appropriate authentication information.

Listing 1: Initial PUT Request

```
PUT /missive.html HTTP/1.1
  Host: files.shakespeare.lit
  Authorization: Basic cm9tZW9AbW9udGFndWUubmV0
  If-None-Match: *
  Content-Length: xxx
  Content-Type: text/html

  [body omitted]
```

Prior to storing the file, the server MUST verify the user's authentication credentials via any supported method. When the file is stored, the server also MUST set the owner "live" property to ensure that only the user that originally posted this file is allowed to modify the file in any way. Other users MAY be allowed to see properties and retrieve the file (upon authentication) but MUST NOT be able to perform operations such as DELETE, MOVE, and PROPPATCH.

Listing 2: HTTP OK

```
HTTP/1.1 200 OK
  Date: Thu, 18 Dec 2003 15:01:20 GMT
```

In the absence of any other authorization method (e.g., RFC 3744 [8] or Security Assertion Markup Language [9]) in use by the deployed WebDAV server, the client SHOULD perform a PROPPATCH request to set the list of Jabber IDs authorized to retrieve this file, and the server MUST NOT allow access until this configuration is completed.

Listing 3: PROPPATCH Request

```
PROPPATCH /missive.html HTTP/1.1
  Host: files.shakespeare.lit
  Authorization: Basic cm9tZW9AbW9udGFndWUubmV0
  Content-Type: text/xml
  Content-Length: 243
```

---

[7]Alternatively, the client MAY first attempt one or more HEAD requests to determine a unique URL.

[8]RFC 3744: Web Distributed Authoring and Versioning (WebDAV) Access Control Protocol <http://tools.ietf.org/html/rfc3744>.

[9]Security Assertion Markup Language <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security>.

```
<propertyupdate xmlns='DAV:'>
  <set>
    <prop>
      <jidlist xmlns='http://www.xmpp.org/extensions/xep-0129.html#
          ns'>
        <jid>juliet@capulet.com</jid>
        <jid>benvolio@montague.net/home</jid>
        <jid>mercutio@capulet.com</jid>
      </jidlist>
    </prop>
  </set>
</propertyupdate>
```

Note: The semantics of the JID list defined above are:

- If a JID is a bare JID (no resource), any fully-qualified form of that JID may access this resource (in the example above, this means that any resource of juliet@capulet.com may access this URL).

- If a JID includes a resource identifier, only that specific JID may access this URL (in the example above, this means that only the JID benvolio@montague.net/home may access this URL; benvolio@montague.net/other may not).

- If both a full JID and a bare JID are specified in a single JID list, the bare JID takes precedence.

The server responds when the properties have been updated. This is typically a 207 (Multi-Part) response, which means that the body can contain multiple status codes, as shown in the following example.

Listing 4: Server Returns HTTP 207

```
HTTP/1.1 207 MultiPart Response
  Date: Thu, 18 Dec 2003 15:03:20 GMT
  Content-Type: text/xml; charset=UTF-8
  Content-Length: 211

  <multistatus xmlns='DAV:'>
    <response>
      <href>http://files.shakespeare.lit/missive.html</href>
      <propstat>
        <prop><jidlist xmlns='http://www.xmpp.org/extensions/xep-0129#
            ns'/></prop>
        <status>HTTP/1.1 200 OK</status>
      </propstat>
    </response>
  </multistatus>
```

Now that the file is available via WebDAV and the client has specified what Jabber IDs may access the URL, the sender sends a message to the target user(s) containing the URL of the file, encapsulated via Out-of-Band Data (XEP-0066) [10]. (The example below shows the file being sent to multiple users using the Extended Stanza Addressing (XEP-0033) [11] protocol.)

Listing 5: Sender Generates XMPP Message with URL

```
<message from='romeo@montague.net/pda' to='multicast.jabber.org'>
    <addresses xmlns='http://jabber.org/protocol/address'>
      <address type='to' jid='juliet@capulet.com'/>
      <address type='to' jid='benvolio@montague.net/home'/>
      <address type='cc' jid='mercutio@capulet.com'/>
    </addresses>
    <x xmlns='jabber:x:oob'>
      <url>http://files.shakespeare.lit/missive.html</url>
    </x>
  </message>
```

When the target recipients have received the message, they may then perform an HTTP GET to download the file (the following request is from juliet@capulet.com).

Listing 6: Recipient GET Request

```
GET /missive.html HTTP/1.1
  Host: files.shakespeare.lit
  Authorization: Digest username="juliet@capulet.com",
                 realm="xmpp",
                 nonce="ec2cc00f21f71acd35ab9be057970609",
                 uri="missive.html",
                 qop=auth,
                 nc=00000001,
                 cnonce="0a4f113b",
                 response="6629fae49393a05397450978507c4ef1",
                 opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

The server then checks to ensure that the provided JID was specified via the jidlist property. If not, the server MUST return an HTTP 403 (Forbidden) error; if so, the server attempts to authorize the user via Verifying HTTP Requests via XMPP (XEP-0070) [12]:

Listing 7: Confirmation Request Sent via Message

```
<message type='normal'
         from='files.shakespeare.lit'
         to='juliet@capulet.com'>
```

---

[10]XEP-0066: Out of Band Data <https://xmpp.org/extensions/xep-0066.html>.
[11]XEP-0033: Extended Stanza Addressing <https://xmpp.org/extensions/xep-0033.html>.
[12]XEP-0070: Verifying HTTP Requests via XMPP <https://xmpp.org/extensions/xep-0070.html>.

```
  <thread>e0ffe42b28561960c6b12b944a092794b9683a38</thread>
  <confirm xmlns='http://jabber.org/protocol/http-auth'
           id='0a4f113b'
           method='GET'
           url='https://files.shakespeare.lit:9345/missive.html'/>
</message>
```

If the Verifying HTTP Requests via XMPP (XEP-0070) [13] verification is successful, the server then allows the file to be retrieved:

Listing 8: Server Returns File

```
HTTP/1.1 200 OK
  Date: Thu, 18 Dec 2003 18:00:00 GMT
  Content-Type: text/html
  Content-Length: xxx

  [body omitted]
```

5 Determining Support

In order to discover a WebDAV server that supports this protocol, a client SHOULD use Service Discovery (XEP-0030) [14]. Support for this protocol MUST be advertised by means of a service discovery feature named "http://www.xmpp.org/extensions/xep-0129.html#ns" (see Protocol Namespaces regarding issuance of one or more permanent namespaces). An example of the discovery flow is shown below.

Listing 9: Client Discovers Services

```
<iq from='romeo@shakespeare.lit/home'
    id='disco1'
    to='shakespeare.lit'
    type='get'>
  <query xmlns='http://jabber.org/protocol/disco#items'/>
</iq>

<iq from='shakespeare.lit'
    id='disco1'
    to='romeo@shakespeare.lit/home'
    type='result'>
  <query xmlns='http://jabber.org/protocol/disco#items'>
    ...
    <item jid='files.shakespeare.lit'/>
    ...
```

---

[13]XEP-0070: Verifying HTTP Requests via XMPP <https://xmpp.org/extensions/xep-0070.html>.
[14]XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.

```
    </query>
</iq>
```

Listing 10: Client Queries Service Regarding Supported Features

```
<iq from='romeo@shakespeare.lit/home'
    id='disco2'
    to='files.shakespeare.lit'
    type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>

<iq from='files.shakespeare.lit'
    id='disco2'
    to='romeo@shakespeare.lit/home'
    type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='http://jabber.org/protocol/http-auth'/>
    <feature var='http://www.xmpp.org/extensions/xep-0129.html#ns'/>
    ...
  </query>
</iq>
```

The user now knows that the "files.shakespeare.lit" service supports this protocol.

6 Security Considerations

See RFC 2518 [15], XMPP Core, and Verifying HTTP Requests via XMPP (XEP-0070) [16] for security considerations related to those protocols, which are used by the profile defined herein. The initiating client MUST ensure that appropriate access controls are specified, normally by performing a PROPPATCH request to set the list of Jabber IDs authorized to retrieve the file. The server MUST NOT allow access to the file until access controls have been specified. In addition, the server MUST NOT allow access to the file by any unauthorized entity.

7 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [17].

---

[15] RFC 2518: HTTP Extensions for Distributed Authoring -- WEBDAV <http://tools.ietf.org/html/rfc2518>.

[16] XEP-0070: Verifying HTTP Requests via XMPP <https://xmpp.org/extensions/xep-0070.html>.

[17] The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.

8 XMPP Registrar Considerations

8.1 Protocol Namespaces

Until this specification advances to a status of Draft, its associated namespace shall be "http://www.xmpp.org/extensions/xep-0129.html#ns"; upon advancement of this specification, the XMPP Registrar [18] shall issue one or more permanent namespaces in accordance with the process defined in Section 4 of XMPP Registrar Function (XEP-0053) [19].

9 XML Schema

```xml
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='http://www.xmpp.org/extensions/xep-0129.html#ns'
    xmlns='http://www.xmpp.org/extensions/xep-0129.html#ns'
    elementFormDefault='qualified'>

  <xs:element name='jidlist'>
    <xs:complexType>
      <xs:sequence minOccurs='0' maxOccurs='unbounded'>
        <xs:element name='jid' type='xs:string'/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

10 Acknowledgements

Thanks to Lisa Dusseault and Julian Reschke for their feedback.

---

[18]The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <https://xmpp.org/registrar/>.

[19]XEP-0053: XMPP Registrar Function <https://xmpp.org/extensions/xep-0053.html>.