Lemonade HTTP Binding

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

Lemonade (see [LEMONADEPROFILE]) describes extensions to the IMAPv4 Rev1 protocol [RFC3501] for optimization in a mobile setting, aimed at delivering extended functionality for mobile devices with limited resources. This draft describes bindings to HTTP.

Conventions used in this document

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

An implementation is not compliant if it fails to satisfy one or more of the MUST or REQUIRED level requirements for the protocol(s) it implements. An implementation that satisfies all the MUST or REQUIRED level and all the SHOULD level requirements for a protocol is said to be "unconditionally compliant" to that protocol; one that satisfies all the MUST level requirements but not all the SHOULD level requirements is said to be "conditionally compliant." When describing the general syntax, some definitions are omitted as they are defined in [RFC3501].

Table of Contents

Status of this Memo...............................................1
Abstract............................................................1
Conventions used in this document................................1
Table of Contents...................................................2
1. Introduction......................................................2
2. HTTP/HTTPS Binding............................................2
   2.1. HTTP/HTTPS Request/Response Format........................2
   2.2. HTTP/HTTPS Request/Response Format........................4
Security Considerations.............................................4
References.........................................................................4
Authors Addresses......................................................5
Intellectual Property Statement......................................6
Full Copyright Statement..............................................7

1. Introduction

This document provides a binding of Lemonade to HTTP. A regular HTTP connection can be used to support in-response connectivity mode for a Lemonade session, whereas a persistent HTTP connection can be used to support inband connectivity.

2. HTTP/HTTPS Binding

2.1. HTTP/HTTPS Request/Response Format

It is possible to use HTTP/HTTPS as transport protocol for commands between the client and server. In this case, the client device embeds Lemonade commands in the body of a request and POSTs it to the Lemonade server. Multiple Lemonade commands may be included in the same POST request. The Lemonade server sends HTTP responses back to
the device client with the result of the execution of the Lemonade commands and pending events.

If the client indicates that it understands gzip-compressed response by setting "Accept-Encoding: gzip" in the request header, server will compress the response, regardless of the current IMAP commands or session state.

The content-type is defined as "application/vnd.lemonade". The general format for a client device to send commands to a Lemonade server is:

    POST /lemonadeServer HTTP/1.1 <CRLF>
    Content-Type: application/vnd.lemonade <CRLF>
    Content-Length: <size of command string(s)> <CRLF>
    Accept-Encoding: gzip <CRLF>
    <CRLF>
    <tag> <Lemonade command> <CRLF>
    [<tag> <Lemonade command> <CRLF>]

- The Lemonade command should be plain text (7bit) and should follow what is specified in [NOTIFICATIONS].
- Multiple Lemonade commands may be sent on the same request. Thus Lemonade commands must be tagged.
- These are the only HTTP headers required to be sent to the Lemonade servers.

When the Lemonade server sends back a response it must be in the following format:

    HTTP/1.1 <HTTP Status Code> <CRLF>
    Content-Type: application/vnd.lemonade <CRLF>
    Content-Length: <size of response string> <CRLF>
    Content-Encoding: gzip <CRLF>
    <CRLF>
    <tag> <Lemonade Server response> <CRLF>
    [<tag> <Lemonade Server response> <CRLF>]

Notes:
The first line is the HTTP status code of the command execution. This could be one of the following:

- 200
  - One of the following 4 cases: all commands succeeded, or at least one command syntax is not correct, or at least command syntax is correct but semantics is not correct, or the current state is not correct. The Lemonade client needs to further parse response body to see what is the case. It should not depend on HTTP status code.

- 500
- at least one command caused internal server error, meaning the Lemonade Server failed to execute the command.

2.2. HTTP/HTTPS Request/Response Format

It is possible to use persistent HTTP or persistent HTTPS so that the server can instantly send notifications to the client while a Lemonade session is open. The client needs to open a persistent connection and keep it active. In this case, the HTTP headers must be sent the first time the client device opens the connection to the Lemonade Server. These headers define a huge content-length and set the transfer coding to be chunked [RFC2616, Sec. 3.6.1]. All subsequent client-server requests are written to the open connection. Thus, the server can use this open channel to push events to the client device at any time.

Security Considerations

The protocol calls for the same security requirements for an in-response and inband connectivity mode as IMAP.

For the outband connectivity mode, servers should use encryption methods for notifications if sensitive information is included in the payload of that notification.

HTTPS protocol can be used to provide end-to-end security

Proxy-based implementations may still require payload encryption for end-to-end security.

References


http://www.ietf.org/rfc/rfc2234

http://www.ietf.org/rfc/rfc2420

http://www.ietf.org/rfc/rfc2616

http://www.ietf.org/rfc/rfc2617

http://www.ietf.org/rfc/rfc2683

http://www.ietf.org/rfc/rfc2177

http://www.ietf.org/rfc/rfc2818

http://www.ietf.org/rfc/rfc2822

http://www.ietf.org/rfc/rfc3501


Authors Addresses
Stephane H. Maes
Oracle Corporation
500 Oracle Parkway
M/S 4op634
Redwood Shores, CA 94065
USA
Phone: +1-650-607-6296
Email: stephane.maes@oracle.com

Rodrigo Lima
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
USA

Chang Kuang
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
USA

Ray Cromwell
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
USA

Vida Ha
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
USA

Eugene Chiu
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
USA

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.
The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Full Copyright Statement

Copyright (C) The Internet Society 2003. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.