Network Working Group RS. Sankar, Ed.
Internet-Draft BrioTribes Technologies (India) Pvt Ltd.
Expires: September 14, 2013 March 13, 2013

Upgrading the 100-CONTINUE status for Broadcast in HTTP/1.1 Protocol
draft-bt-http-broadcast-00

Abstract

The Hypertext Transfer Protocol -- HTTP/1.1 in RFC 2616 uses the 100-Continue status message with Expect: Continue header to send request body at a later time to the origin server. This memo explains how to extend this status message to send multiple responses from the origin server to the client for an extended period of time. This achieves the essential function of broadcast from the origin server to the client, without the client waiting for a response to a request.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on September 14, 2013.

1. Introduction

The Hypertext Transfer Protocol -- HTTP/1.1 in RFC 2616 is essentially a request response protocol. A request has to be sent from the client to the origin server, for the server to send across any information to the client.

In recent times HTTP/1.1 protocol is widely used to provide presentation logic for business applications. A number of business applications such as dashboard application requires a push functionality that pushes data from the origin server to the client without intervention from the client.

In the standard approach called long-polling, the client starts a connection and waits for a response. The server responds when the data is available. Once a response is received, the client has to request again to receive the next message. Here, the server cannot send more than one response for the same request. This is better than polling, but it still requires the client to keep resending a request to receive a response.

The best approach to broadcasting is for the client to initiate a request indicating that the connection is available to receive and process broadcast messages. The origin server and client maintain this connection. The server sends any broadcast message on this connection using 100-Continue status, when data is available. Once the timeout for the connection is reached, or the server decides no more broadcasts are possible on the connection, the server disconnects using the 200-OK message. The client can reestablish the connection if it requires to read more broadcast messages.

In a broadcast, the ORIGIN SERVER is aware of when data is available to be sent and when no more broadcasts can be sent or when a broadcast connection has to be closed. For eg., when a client logs out of a session, the server is aware that the client is no more authorised to receive broadcasts or when a session times out, no more broadcast can be sent. Thus while the client initiates a connection, the server decides when to disconnect the connection.

When the client initiates a broadcast request, if the server determines that the client cannot receive broadcast messages based on application logic, the server can send back an appropriate error message immediately such as 404-Not Present to prevent the client from receiving broadcast messages.

2. Extension to Extend 100:Continue status

Broadcasting is a mechanism where an origin server can push data to a client without the client knowing when to expect data from the origin server. This upgrade proposes the following sequence to allow broadcast using HTTP/1.1 protocol:

The client initiates a connection to the server with the below headers

Expect: Broadcast

Connection: Keep-Alive

KeepAliveTimeout: time in ms

The origin server recognizes the broadcast header. It interprets the header to indicate that the client can process broadcast data.

The keepAliveTimeout is the time for which the connection will be open to process broadcast data.

Once this initial connection is established by the client, the origin server will push data to the client using the 100-Continue messages. The data to be sent to the client is sent in the body of 100-Continue messages.

The message is sent across with the following headers:

Content-Length -- the length of content

Content-Type -- the type of content

The client uses the content-length to read the content of the data broadcast. More than one 100-Continue message can be sent before the 200-OK message is sent.

For the current upgrade to work, both the client and the server MUST work with the upgraded 100-Continue status message. The client must process the 100-Continue status messages instead of ignoring them.

Once the KeepAliveTimeout duration is reached, the server sends out the 200-OK message to close the connection. The KeepAliveTimeout can be an optional header. If the header is not present, then the connection is closed when the server determines it to be logically the end of broadcast.

3. Uses of HTTP Broadcast

This upgrade can be used to send data in standard formats such as JSON or XML to be displayed via an AJAX call in an already loaded HTML page.

Typical application includes dashboard kind of application, where data displayed needs to be updated only when they change or a multi-player game played online where one player action needs to be displayed in the screen of the other players.

The intention of this upgrade is not to send multiple HTML pages on the same connection, but to send incremental data that needs to update only portion of the HTML page displayed.

4. References

[httpproto]
, "HTTP/1.1 Protocol RFC ", ,
<http://www.rfc-editor.org/rfc/rfc2616.txt>.

Author's Address

Raji Sankar (editor)
BrioTribes Technologies (India) Pvt Ltd.

Phone: +91-9902530998
Email: raji.sankar@briotribes.com