

Push technology

From Wikipedia, the free encyclopedia

Push, or **server push**, describes a style of Internet-based communication where the request for a given transaction is initiated by the publisher or central server. It is contrasted with pull/get, where the request for the transmission of information is initiated by the receiver or client.

Contents

- 1 General use
- 2 Examples
 - 2.1 HTTP server push
 - 2.2 Pushlet
 - 2.3 Long polling
 - 2.4 Flash XMLSocket relays
- 3 See also
- 4 References
- 5 External links

General use

Push services are often based on information preferences expressed in advance. This is called a publish/subscribe model. A client "subscribes" to various information "channels" provided by a server; whenever new content is available on one of those channels, the server pushes that information out to the client.

Synchronous conferencing and instant messaging are typical examples of push services. Chat messages and sometimes files are pushed to the user as soon as they are received by the messaging service. Both decentralised peer-to-peer programs (such as WASTE) and centralised programs (such as IRC or XMPP) allow pushing files, which means the sender initiates the data transfer rather than the recipient.

Email may also be a push system: The SMTP protocol is a push protocol (see Push e-mail). However, the last step—from mail server to desktop computer—typically uses a pull protocol like POP3 or IMAP. Modern e-mail clients make this step seem instantaneous by repeatedly polling the mail server, frequently checking it for new mail. The IMAP protocol includes the IDLE command, which allows the server to tell the client when new messages arrive. The original BlackBerry was the first popular example of push-email in a wireless context.

Another example is the PointCast Network, which was widely implemented in the 1990s. It delivered news and stock market data. Both Netscape and Microsoft integrated it into their software at the height of the browser wars, but it was never popular and later faded away. Browsers replaced it in the 2000s with RSS (a pull system).

Other uses of push-enabled web applications include market data distribution (stock tickers), online chat/messaging systems (webchat), auctions, online betting and gaming, sport results, monitoring consoles, and sensor network monitoring.

Examples

HTTP server push

HTTP server push (also known as HTTP streaming) is a mechanism for sending unsolicited (asynchronous) data from a web server to a web browser. HTTP server push can be achieved through any of several mechanisms.

As a part of HTML5 the WebSocket API allows a web server and client to communicate over a full-duplex TCP connection.

Generally the web server does not terminate a connection after response data has been served to a client. The web server leaves the connection open so that if an event occurs (for example, a change in internal data which needs to be reported to one or multiple clients), it can be sent out immediately; otherwise, the event would have to be queued until the client's next request would have been received. Most web servers offer this functionality via CGI (e.g., Non-Parsed Headers scripts on Apache). The underlying mechanism for this approach is Chunked transfer encoding.

Another mechanism is related to a special MIME type called `multipart/x-mixed-replace`, which was introduced by Netscape in 1995. Web browsers interpret this as a document changing whenever the server feels like pushing a new version to the client.^[1] It is still supported by Firefox, Opera, and Safari today, but it is ignored by Internet Explorer.^[2] It can be applied to HTML documents, and also for streaming images in webcam applications.

The WHATWG Web Applications 1.0 proposal^[3] includes a mechanism to push content to the client. On September 1, 2006, the Opera web browser implemented this new experimental system in a feature called "Server-Sent Events".^{[4][5]} It is now being standardized as part of HTML5.^[6]

Pushlet

In this technique, the server takes advantage of persistent HTTP connections, leaving the response perpetually "open" (i.e., the server never terminates the response), effectively fooling the browser to remain in "loading" mode after the initial page load could be considered complete. The server then periodically sends snippets of JavaScript to update the content of the page, thereby achieving push capability. By using this technique, the client doesn't need Java applets or other plug-ins in order to keep an open connection to the server; the client is automatically notified about new events, pushed by the server.^{[7][8]} One serious drawback to this method, however, is the lack of control the server has over the browser timing out; a page refresh is always necessary if a timeout occurs on the browser end.

Long polling

Long polling is itself not a true push; long polling is a variation of the traditional polling technique, but it allows emulating a push mechanism under circumstances where a real push is not possible, such as sites with security policies that require rejection of incoming HTTP/S Requests.

With long polling, the client requests information from the server exactly as in normal polling, except it issues its HTTP/S requests (polls) at a much slower frequency. If the server does not have any information available for the client when the poll is received, instead of sending an empty response, the server holds the request open and waits for response information to become available. Once it does, the server immediately sends an HTTP/S response to the client, completing the open HTTP/S Request. In this way the usual response latency (the time between when the information first becomes available and the next client request) otherwise associated with polling clients is eliminated.

For example, BOSH is a popular, long-lived HTTP technique used as a long-polling alternative to TCP when TCP is difficult or impossible to employ directly (e.g., in a web browser);^[9] it is also an underlying technology in the XMPP, which Apple uses for its iCloud push support.

Flash XMLSocket relays

This technique, used by Cbox and other chat applications, makes use of the XMLSocket object in a single-pixel Adobe Flash movie. Under the control of JavaScript, the client establishes a TCP connection to a unidirectional relay on the server. The relay server does not read anything from this socket; instead it immediately sends the client a unique identifier. Next, the client makes an HTTP request to the web server, including with it this identifier. The web application can then push messages addressed to the client to a local interface of the relay server, which relays them over the Flash socket. The advantage of this approach is that it appreciates the natural read-write asymmetry that is typical of many web applications, including chat, and as a consequence it offers high efficiency. Since it does not accept data on outgoing sockets, the relay server does not need to poll outgoing TCP connections *at all*, making it possible to hold open tens of thousands of concurrent connections. In this model, the limit to scale is the TCP stack of the underlying server operating system.

See also

- Apple Push Notification Service
- Android Cloud to Device Messaging Service
- BOSH
- Comet
- Client–server model
- File transfer
- Pull technology
- Push Access Protocol
- Push e-mail
- SPDY
- Streaming media
- SQL Server Notification Services
- WebSocket
- BlazeDS
- Lightstreamer
- GraniteDS

References

1. CGI Programming on the World Wide Web (http://www.oreilly.com/openbook/cgi/ch06_06.html) O'Reilly book explaining how to use Netscape server-push
2. Server-Push Documents (HTML & XHTML: The Definitive Guide) (http://victor.transformadora.com/Oreilly/wdesign/xhtml/ch13_03.htm) O'Reilly book explaining server-push
3. "Web Applications 1.0 specification" (<http://www.whatwg.org/specs/web-apps/current-work/#scs-server-sent>).
4. "Event Streaming to Web Browsers" (<http://my.opera.com/WebApplications/blog/show.dml/438711>). 2006-09-01. Retrieved 2007-03-23.
5. "Opera takes the lead with AJAX support among browsers: More efficient streaming" (<http://operawatch.com/news/2006/09/opera-takes-the-lead-with-ajax-support-among-browsers-more-efficient-streaming.html>). 2006-09-01. Retrieved 2007-03-23.
6. Server-Sent Events (<http://dev.w3.org/html5/eventsource/>)
7. Pushlets introduction (<http://www.pushlets.com/doc/whitepaper-toc.html>)
8. JavaWorld article about pushlets (<http://www.javaworld.com/jw-03-2000/jw-03-pushlet.html>)
9. "XEP-0124: Bidirectional-streams Over Synchronous HTTP (BOSH)" (<http://xmpp.org/extensions/xep-0124.html#intro>). Retrieved 2012-06-26.

External links

- W3C Push Workshop (http://www.w3.org/Architecture/9709_Workshop/). A 1997 workshop that discussed push technology and some early examples thereof
- HTTP Streaming with Ajax (http://ajaxpatterns.org/HTTP_Streaming) A description of HTTP Streaming from the Ajax Patterns website
- The WebSocket API (<http://www.w3.org/TR/websockets/>) candidate recommendation
- HTML5 Server-Sent Events (<http://www.w3.org/TR/eventsource/>) draft specification
- HTTP Server Push at UDAParts (<http://www.udaparts.com/document/Tutorial/httppush.htm>) A commercial grade Comet implementation with source code available
- Google Web Toolkit's Server Push FAQ (<http://code.google.com/p/google-web-toolkit-incubator/wiki/ServerPushFAQ>) Explains Server Push and how you can achieve this with GWT
- Ajax Push Engine (<http://www.ape-project.org>) An open source web server for pushing real-time data to Rich Internet Applications using Comet
- GWT Comet Adapter (<http://code.google.com/p/gwt-comet-streamhub/>) Open source adapter for implementing server push in Google Web Toolkit
- XEP-0124: Bidirectional-streams Over Synchronous HTTP (BOSH) (<http://xmpp.org/extensions/xep-0124.html>) A dual HTTP long-lived connection
- Part 2 HTML5 Server-push Technologies (<http://today.java.net/article/2010/04/26/html5-server-push-technologies-part-2>) Introduction into HTML5 Server-push Technologies. Part 1 (<http://today.java.net/article/2010/03/31/html5-server-push-technologies-part-1>) covers ServerSent Events, Part 2 (<http://today.java.net/article/2010/04/26/html5-server-push-technologies-part-2>) covers WebSockets
- Data from a Flex Publisher Client to a Consumer Client (<http://www.adobe.com/devnet/flex/articles/flex-apache-nms.html/Pushing>) Covers Apache NMS, ActiveMQ and WebORB
- Countly, an open source, self-hosted push notification platform with support for APNS (<http://github.com/countly/countly-server>)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Push_technology&oldid=667105132"

Categories: [Internet terminology](#) | [Mobile technology](#) | [Web development](#) | [Ajax \(programming\)](#)

- This page was last modified on 15 June 2015, at 21:23.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.