



IBM Software Group

IBM WebSphere® Application Server Feature Pack for Web 2.0

Web messaging service



@business on demand.

© 2007 IBM Corporation
Updated December 17, 2007

This Presentation will focus on the IBM WebSphere Application Server Feature Pack for Web 2.0 Web messaging service.

Agenda

- Overview
- Installation
- Summary



The presentation will cover the overview, installation and limitation of Web messaging service. At the end of the presentation, there will be a summary of the Web messaging service.

Overview

- The IBM Web messaging service is a publish/subscribe implementation connecting the browser to the WebSphere Application Server service integration bus for server-side event push to the browser.
- Client/server communication is achieved through the Bayeux protocol.
- Client side support for the Bayeux protocol is provided by the Dojo Toolkit.

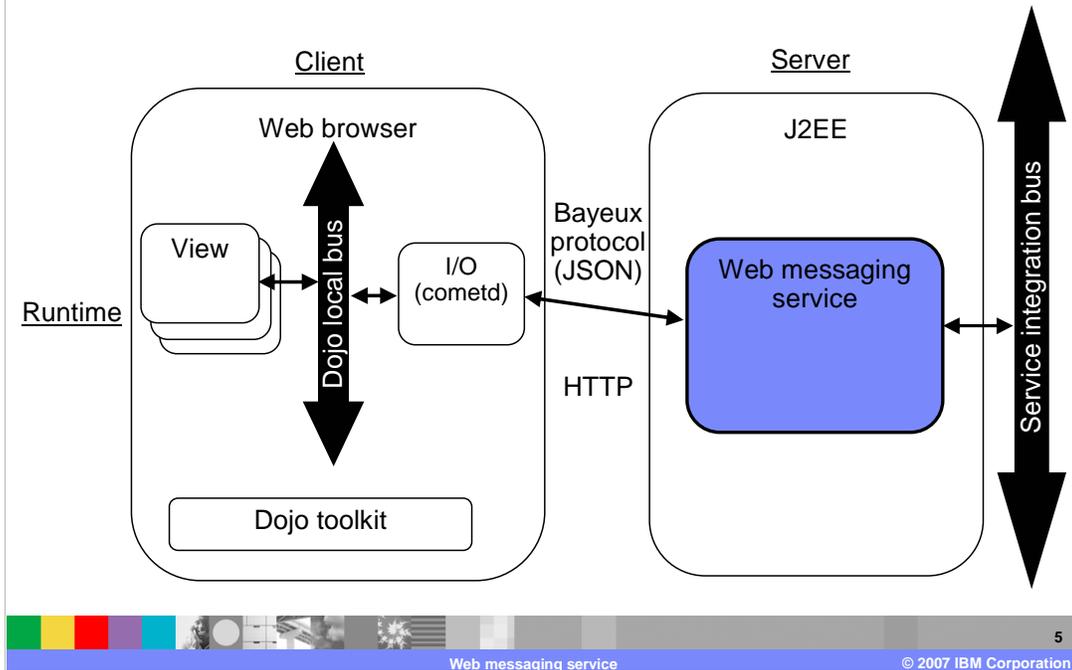
4

Web messaging service

© 2007 IBM Corporation

The IBM Web messaging service is a publish/subscribe implementation connecting the browser to the WebSphere Application Server service integration bus for server-side event push to the browser. Client/server communication is achieved through the Bayeux protocol. The Bayeux protocol is an HTTP based messaging routing protocol. Client side support for the Bayeux protocol is provided by the Dojo Toolkit. Currently the Dojo Toolkit is the only JavaScript™ library to support the Bayeux protocol, although any JavaScript library or HTTP client that implements the Bayeux protocol support can communicate with Web messaging service. The Web Message Service implementation bridges incoming Bayeux requests to the service integration bus allowing Web services, JMS clients, or any item connected to the service integration bus to publish events to Web-based clients. You can use the Web messaging service in a new or existing application by placing a runtime JAR file into WebSphere Application Server. Place a utility file library in an application Web module then set up a simple configuration file, and configure servlet mappings.

Web messaging service – Overall architecture



This overall architecture shows the different pieces of the Web messaging service..

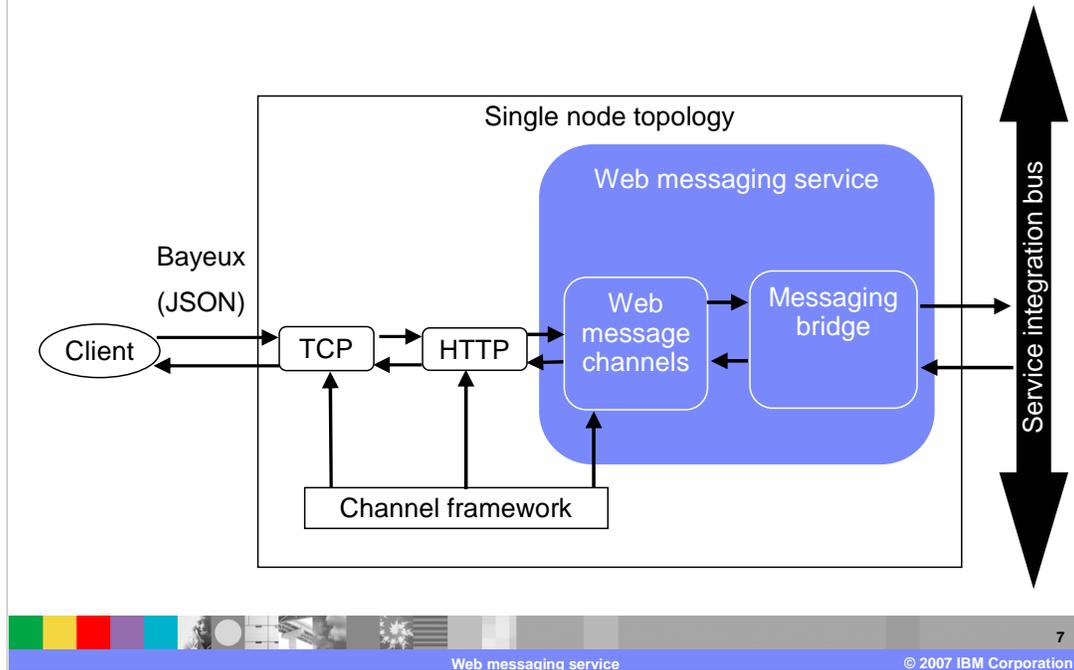
The Web messaging service server bridges browser clients to the service integration bus allowing a Web service or any other item connected to the service integration bus to publish events to Web-based clients. Note the Bayeux protocol mentioned earlier.

Usage

- Web messaging service can be used in a new or existing application by:
 - ▶ Placing a runtime Java archive (JAR) file into WebSphere Application Server.
 - ▶ Placing a utility file library in an application Web module.
 - ▶ Setting up a simple configuration file, and configuring servlet mappings.

Web messaging service can be used in a new or existing application by placing a runtime Java archive (JAR) file into WebSphere Application Server or by placing a utility file library in an application Web module. Web messaging service can also be used in an application by setting up a simple configuration file, and configuring servlet mappings.

Server architecture



Web messaging service is designed to work with existing versions of WebSphere Application Server. The current Web container does not scale well with Bayeux protocol long-lived HTTP connection in which a server typically holds a connection open for a certain time to wait for the server to push an event to the browser. To allow for scalability, channel framework architecture (WebMsg channels) that extends the HTTP channel to bridge incoming Bayeux requests to the service integration bus is used.

Scalability

- Bayeux protocol communicates through a long-lived HTTP connection
 - ▶ server typically holds a connection open for a certain time to wait for the server to push an event to the browser.
 - ▶ Web container cannot scale as each waiting client consumes a thread waiting for an event.
- WebMsg channels used to meet the scalability needs of Web messaging service
 - ▶ Channel framework extends the HTTP channel to bridge incoming Bayeux requests to the service integration bus.

8

Web messaging service

© 2007 IBM Corporation

In a typical application a browser periodically requests updates from a server at a specified interval (polling). The Bayeux protocol communication types differ from the traditional polling model and communicates through a long-lived HTTP connection. In this type of a connection, a server typically holds a connection open for a certain time to wait for the server to push an event to the browser. With this style of communication, the Web container cannot scale as each waiting client consumes a thread waiting for an event. Other servers that handle this style of request have different methods for scaling. The Web messaging service is designed to work with existing versions of WebSphere Application Server. The current Web container does not have the ability to scale well with this method of communication. Therefore a new mechanism is introduced in the Web messaging service to achieve scalability. This new mechanism takes advantage of channel framework architecture and creates new channel framework channel that extends the HTTP channel to bridge incoming Bayeux requests to the service integration bus. These new channels are called WebMsg channels.

Dojo toolkit integration

- Dojo toolkit cometd client module allows for browser support for Web messaging service.
- Server-driven events tie in to the dojo event and topic system through cometd client.
- Dojo cometd usage example includes the QuoteStreamer Sample.



Current browser support for Web messaging service is made possible with the Dojo Toolkit cometd client module. Initialization, subscription and unsubscription, and publishing operations are easy to accomplish with the cometd client. Through the cometd client, server-driven events tie in to the dojo event and topic system. Dojo cometd usage examples are provided in the AJAX developers guide and QuoteStreamer Sample application. Quote Streamer Sample uses the Web messaging service to simulate stock quotes to a Dojo-enabled client application

Service integration bus connectivity

- Web messaging service connects browser clients to the built in messaging engine, the service integration bus, for event subscription, publishing, and message receipt.
- Bridge to the service integration bus occurs through direct API calls to a service integration bus topic space.
 - ▶ Thus allowing for many options to publish messages to Web clients, examples:
 - EJB publishing to topic
 - JMS clients publishing to a topic
 - Web service can publish to other Web clients

The Web messaging service connects browser clients to the built in messaging engine, the service integration bus, for event subscription, publishing, and message receipt. The underlying bridge to the service integration bus occurs through direct API calls to a service integration bus topic space. No JMS artifact setup is needed for the Web messaging service. There are many different ways to publish messages to Web clients since the web clients are connected to the service integration bus. Some of these options include standard EJB publishing to topic and a JMS client publishing to a topic. A Web service such as WS-notification is also an option. The Web client itself can publish to other web clients as an option.

In the Web messaging service, incoming web messaging requests are bridged to the service integration bus topic space for publish/subscribe operations. Before installing a Web messaging enabled application, a service integration bus must be set up and configured. At the very minimum this process consists of; creating the bus, adding a server as a bus member and creating a new topic space.

Section

Setup and installation



This next section will show you how to setup and install Web messaging service.

Installation

- The Web messaging service consists of a run time Java archive (JAR) file.
- A library designed to be inserted into a new or existing Web module as a utility JAR file.
- For complete install instructions refer to the documentation on Web messaging that comes with the IBM WebSphere Application Server Feature Pack for Web 2.0

The Web messaging service consists of a runtime jar that gets placed in the WebSphere Application Server install directory. It also has a library designed to insert into a new or existing Web module as a utility Java Archive (JAR) file.

For complete install instructions refer to the documentation on Web messaging that comes with the IBM WebSphere Application Server Feature Pack for Web 2.0

Web messaging enabled applications

- With the Web messaging library one can enable the Web messaging service and develop a Web messaging enabled application.
- The QuoteStreamer Sample application can be examined and used as a starting point for developing Web messaging enabled applications.



With the Web messaging Library one can enable the Web messaging service and develop a Web messaging enabled application.

The QuoteStreamer Sample application can be examined and used as a starting point for developing Web messaging enabled applications.

Section

Summary



The next section will summarize what you have learnt so far on Web messaging service.

Summary

- The IBM Web messaging service is a publish/subscribe implementation connecting the browser to the WebSphere Application Server service integration bus for server-side event push to the browser.
- The Web messaging service connects Ajax clients to real-time updated data like stock quotes, instant messages, or any scenario where dynamic data needs to be sent from server to browser.
- Limitation when using the Web messaging service is that there is support for Bayeux long-polling transport only.

15

Web messaging service

© 2007 IBM Corporation

The IBM Web messaging service is a publish/subscribe implementation connecting the browser to the WebSphere Application Server service integration bus for server-side event push to the browser. The Web messaging service connects Ajax clients to real-time updated data like stock quotes, instant messages, or any scenario where dynamic data needs to be sent from server to browser.

Examples of Web messaging service applications include stock quotes, front office applications, SIP applications and inbound call consoles. Other examples include collaborative document authoring, chat & instant messaging, sporting event scores, browser based e-mail, management consoles and monitoring & alerts.

Limitations of the Web messaging service include the support for Bayeux long-polling transport only.

References

- Bayeux/cometd - <http://www.cometd.com/>

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_was_web20fp_Web-Messaging-Service.ppt

This module is also available in PDF format at: [../was_web20fp_Web-Messaging-Service.pdf](http://was_web20fp_Web-Messaging-Service.pdf)



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM WebSphere

EJB, J2EE, Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.