IBM Software Group

IBM WebSphere Application Server Feature Pack for Communications Enabled Applications

Communication services overview

@business on demand.

This presentation will discuss an overview of the IBM WebSphere® Application Server Feature Pack for Communications Enabled Applications communication services.

Agenda

- Overview

- Communication services interfaces

  ▸ Rest API

  ▸ Web services interfaces

- IP-PBX sample application

- Summary

This presentation will begin with a brief introduction to the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications communication services. You will then be introduced to the different ways to access the communication services using REST requests, or Web service requests. A description of the sample IP-PBX application is provided with the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications before summarizing the presentation.

Section

Overview

3

This section will discuss an overview of the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications communication services.

Communication services overview

- The communication services provided with the feature pack are server side services, available to drive communication in enterprise applications

- REST APIs
  - ▸ WebSphere system application invoked using REST APIs
  - ▸ Listening for requests
  - ▸ Drives IBM WebSphere Application Server Feature Pack for CEA Web widgets

- Web service interface
  - ▸ Business application use

The IBM WebSphere Application Server Feature Pack for Communications Enabled Applications has a server side component that is referred to as the communication service. The communication service provides the back end functionality to drive the features of the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications. These services perform activities to make calls between two parties, monitor call status, provide call notification of incoming calls, and register and un-register for call notification to name a few. The communication services are accessed by the Web widgets using REST APIs, and can also be accessed using Web services by business applications.
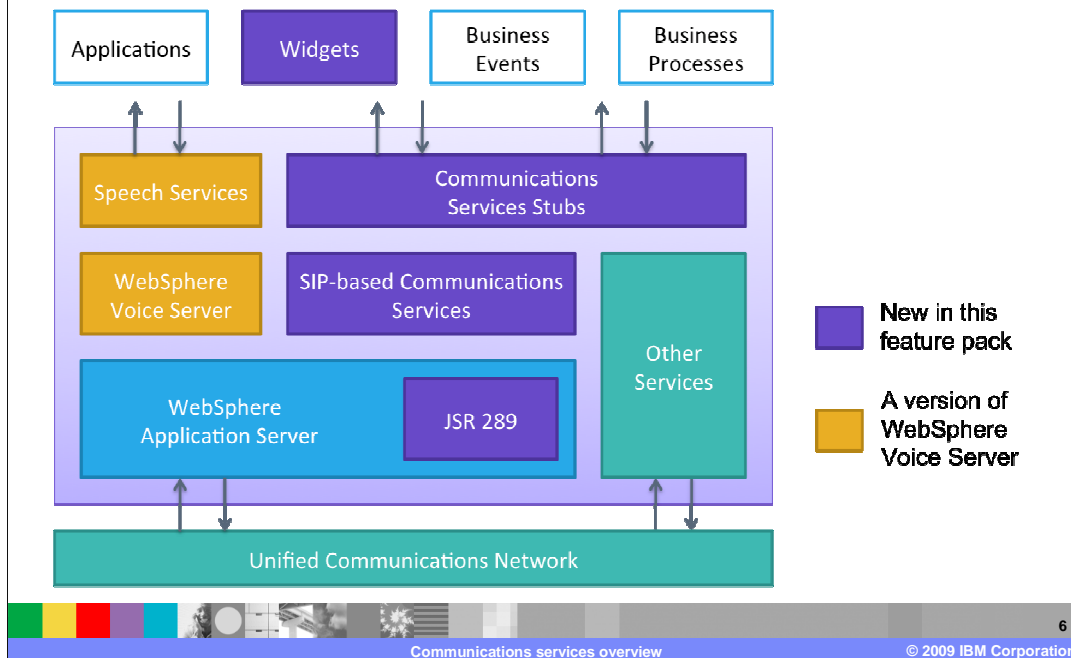
Communication services

- Telephony and non-telephony enterprise application communication components access communication services
  - ‣ Telephone related activities routed through REST calls to the IP-PBX
  - ‣ Web co-browsing data passed through REST calls as well
- IP-PBX sample application
  - ‣ Ready to integrate testing solution
  - ‣ Rapid start to development
- Third party IP telephony tools can be integrated for testing

5

The communication services provide the back end functionality for both telephony and non-telephony actions for the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications. Telephone related activities performed by the communication services are routed through an IP-PBX to handle voice communication information processing. Non telephony related activities include handling co-browsing Web data for the instance where two parties are browsing the same Web content in a co-browsing session using the collaboration dialog widget. For testing purposes, a sample IP-PBX application is provided with the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications. This sample application allows developers to test telephony related programming components using vendor IP telephony softphones to simulate telephone calls. Having the sample IP-PBX application included in the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications allows for rapid start to development of telephony related application components.

Overview diagram

The IBM WebSphere Application Server Feature Pack for Communications Enabled Applications communication services serve as the integration layer between the Web widgets and the application server layer. The communication services accept requests either using the REST APIs or Web services, and connect the requesting application layers to the unified communication network using an IP-PBX. The services also handle non-telephony related activities, such as passing data back and forth between two collaboration dialog widgets during a co-browsing Web session. For example, two parties are co-browsing a Web page. One party in the co-browsing session chooses to send a page they are viewing to the other party. That information is relayed by way of REST requests to the communication service layer. The communication service then passes the information along to the peer's collaboration dialog box. The peers collaboration dialog box will then display the page that the other party sent in the request.

Section

Communication services interfaces

This section will discuss the communication services interfaces, including the REST and Web services interfaces.

Communication services interfaces

- Two ways to access the communication services
  - ▶ REST API
  - ▶ Web services
- REST APIs
  - ▶ Representational state transfer
  - ▶ Servlet accepts HTTP requests
  - ▶ Requests define operations to perform
    - Example: register a telephone number to monitor with the IP-PBX
  - ▶ Information sent to the servlet in either JSON or XML format

8

The communication services provided with the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications has two ways of being used. They can be accessed using REST API requests, or Web services requests. The REST requests are sent using HTTP requests, so they are perfect for any components that reside in the Web layer of an application. The HTTP requests that are sent to the communication services have a distinct format that indicates to the services what activities are being requested. Information needed by the communication services to perform the operation is provided in either JSON or XML format. JSON is a standard way to pass data that organizes data in key value pairs in a simple list format.

IBM

Web services interface

- Business applications can access the communication services using Web service calls

- Web service calls are available for each telephony based REST application service call

- Performs in the same manner as REST API calls with one exception:
  - ▸ Web service defines endpoint listener that does not need to constantly poll REST servlet

9

The Web services interface that is supported by the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications communication services is intended for use in the non-Web business application layer. For parts of an application that are not Web related, Web service calls can be used to access the telephony services provided by the feature pack. The operations available on the Web service interface are the same as the calls that are available for the REST APIs.

IBM

Telephony actions

- These telephony actions are available from the REST servlet
    - ▸ Make a call
    - ▸ End a call
    - ▸ Get info on a call
    - ▸ Register for call notification
    - ▸ Polling for a call notification
    - ▸ Un-register for call notification

10

The REST API provided by the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications communication service include several actions. These actions include the ability to make a call, end a call, and gather information related to an ongoing telephone call. APIs can be used to register a user for call notification, poll for available calls for call notification, and un-register for call notification. Long standing requests are used when a REST call to the communication services are used. If, for example, a call notification widget is polling the communication services to see if there are any available calls, there might not be any immediate calls to connect to. The request will remain active for 30 seconds to wait to see if a call will arrive before timing out and returning a response to the call notification widget indicating no calls are available. If a telephone call is received by the communication services within that 30 second window when a request is active, the service will return a response to the request as soon as the incoming call arrives.

Web service actions

- Web services has these actions
  - ▶ openSession
  - ▶ makeCall
  - ▶ endCall
  - ▶ closeSession
  - ▶ WS-Notification based asynchronous notifications

Web service calls can be made by non-Web components in an enterprise application to access the communication services from that application layer. Web service calls are available to open a session, make a call, end a call, close a telephony session, and to use WS-Notification based asynchronous notifications. When Web services are used to open a session or close a session, call notification is registered and unregistered as part of the open and closing of the session.

REST servlet co-browsing actions

- Enable collaboration
- Get collaboration status
- Start collaboration session with peer URI
- End collaboration session
- Fetch data from the peer
- Send data to the peer

12

© 2009 IBM Corporation

The communication services provide functionality for more than just the telephony actions provided in the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications. The REST APIs can also be used to access the Web collaboration functions of the communication services. REST requests exist for enabling a new collaboration session, to gather data about collaboration status of a co-browsing session, and can start a collaboration session with a peer URI. REST calls can also be used to end a collaboration session that is active, or send or fetch data from the peers collaboration dialog widget. This functionality can be seen in the collaboration dialog widget icons. Icons are shown in the collaboration dialog widget to show if the peer window is open, if the collaboration session is connected, and to display who is driving the co-browsing Web navigation. For more information on the collaboration dialog widget and these icon statuses, see the collaboration dialog widget presentation.

Developing applications with Web services

- WSDL file is provided with the IBM WebSphere Application Server Feature Pack for CEA
  - ▶ http://host:port/commsvc.rest/ControllerService?wsdl
- Using the WSDL file a Web services client can be generated using an enterprise development IDE
- Rational® Application Developer V7.5 supports WebSphere Application Server V7.0.0.1 Web service client generation

Web services clients can be developed using the WSDL file that is provided with the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications. You can access the WSDL definition by pointing your browser to your application server where the feature pack is installed at the URL indicated above. The WSDL file can then be downloaded and used to generate a Web service client. Integrated development environment tools such as Rational Application Developer can be used to generate Web service clients using the WSDL file, and that client can then be used to integrate Web service calls into your applications.

Example scenario

- A hardware store business wants to track how many times they receive service calls in a day
- Business application can be notified of each call that comes in and a counter can be incremented
- Business application registers with the IP-PBX to be notified of incoming calls with a Web service call
- Upon receiving incoming calls, the IP-PBX notifies the CEA system which then sends a notification to the waiting listener registered using the Web service
- The business application receives the notice, increments the counter keeping track of incoming calls
- Data can be stored and later accessed to create a report of how many calls the business receives daily

14

Here is an example of when you might want to integrate Web services into your business applications to use the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications features. In this example, a hardware store wants to track the number of service calls they receive in a day. They have a business application with a non-Web layer that can be notified each time a call arrives, and that business application can keep track of the number of incoming calls using a counter. The business application uses a Web service client generated from the WSDL file provided by the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications. The application registers with the IP-PBX to be notified of incoming calls using the Web service client. When calls are received, the IP-PBX notifies the communication service, which then sends a notification to the listener that was registered with the initial Web service call by the business application. The business application receives the notification in its listener, and increments a counter. The data that the application gathers in its counter is logged and stored to be used in a report that is generated at the end of the business day.

Section

IP-PBX sample application

This section will discuss the sample IP-PBX application shipped with the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications.

IP-PBX overview

- IP-PBX stands for internet protocol private branch exchange
- An IP-PBX is a business telephone system designed to deliver voice over a data network
- It interoperates with the PSTN (public switched telephone network)
- REST back end talks to an IP-PBX using SIP (session initiation protocol)
- The data exchanged includes XML messages based on CSTA (computer-supported telephony applications)
- SIP is used in telephony enterprise applications
- Telephony applications that perform telephone related actions communicate using an IP-PBX

16

In order to start working with telephony related development, an IP-PBX that supports CSTA is needed in order to establish telephone calls. An IP-PBX is used to deliver voice data over a network, and interoperate with the public switched telephone network, to connect telephone calls from regular public telephones, to private telephone networks. In the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications, the REST service back end talks to an IP-PBX using session initiation protocol to perform telephony related activities.

IP-PBX sample application

- IBM WebSphere Application Server Feature Pack for CEA provides a sample IP-PBX as an enterprise archive (EAR)

- This sample allows for a quick start to development and testing

- In production, an IP-PBX that supports CSTA is used to drive telephony in enterprise applications

- In production an IP-PBX is loaded on a Linux® blade server or similar system on a dedicated box

- Third party PBX systems can be used during development as well
  - ▸ Complex setup of PBX is required
  - ▸ Possibly time consuming for development purposes

It costs tens of thousands of dollars for a real IP-PBX system that supports CSTA, which can be very costly especially for development purposes. When developing telephony applications, for testing purposes a mock IP-PBX can reduce costs and make light weight environments available for developers to begin development activities and test telephony components in enterprise applications. A sample IP-PBX is provided with the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications. This sample allows for rapid start to development of applications with embedded telephony components. It allows developers to begin working with telephony related technologies in their applications, and test telephony components without having to connect to a real IP-PBX. Applications developed in this manner can be thoroughly tested, and then can be ported to an environment where a real IP-PBX is installed for final verification testing. In a production environment, a production IP-PBX is used to drive the telephony related activities in a business application. Although the sample IP-PBX is provided with the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications, you are not limited to using it. You have the option of using other vendor PBX systems, although setup can be complex and time consuming.

Section

Summary

18

This section will summarize the communication services overview presentation.

Summary

- IBM WebSphere Application Server Feature Pack for CEA communication services provide telephony and Web collaboration support for enterprise applications

- REST servlet handles HTTP requests from Web based applications

- Web service to integrate telephony into business applications

- Sample IP-PBX application provided for rapid start to development

The IBM WebSphere Application Server Feature Pack for Communications Enabled Applications communication services provide the back end server side support for communication applications. The communication services can be accessed using REST APIs and Web services in non-Web front end portions of enterprise applications. The REST servlet handles HTTP requests from Web components to access telephony actions.

A sample IP-PBX is provided with the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications, and can be used to get started quickly to test and develop communication enabled applications. Using the sample IP-PBX allows developers to test telephony related components without having the complex and often time consuming task of connecting to a real IP-PBX.

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_CEAFP_CommServices.ppt

This module is also available in PDF format at: ../CEAFP_CommServices.pdf

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

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

Rational   WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

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.

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 NONINFRINGEMENT. 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 2009. 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.

CEAFP_CommServices.ppt   Page 21 of 21