



IBM Software Group

## IBM WebSphere Application Server Feature Pack for Communications Enabled Applications

### *REST API*



@business on demand.

© 2009 IBM Corporation  
Updated July 29, 2009

This presentation will discuss the REST API for the IBM WebSphere® Application Server Feature Pack for Communications Enabled Applications.

## Agenda

- Overview
- Communication services REST interface
- Examples
- Summary



This presentation will discuss an overview of the REST API and some examples of how to use the REST API to make calls to the communication services REST interface.

## Section

# *Overview*



This section will discuss an overview of the REST API.

## REST API

- 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

The communication services in the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications can be accessed using HTTP requests to the REST interface. Calls to the REST interface are similar to standard HTTP requests, only they have a specific format that is recognized by the REST interface implementation. The communication service has a servlet that is constantly listening for requests that parses the incoming request and returns a response to it in either JSON or XML format. If the REST data is sent initially using JSON, the response is in JSON format. If the data is sent to the REST interface in XML format in the request, the response is in XML format. There are REST requests available to perform many operations, including the ability to make a call, end a call in progress, gather information about a call in progress, and register and un-register for call notification.

## Section

# ***Communication services REST interface***



This section will discuss the communication services REST interface.

## 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



The communication services REST interface provides actions to make a call, end a call in progress, and gather information about a call. You can also use the REST API to perform call notification related activities, including registering for call notification, poll to see if a call is ready for call notification, and un-register for call notification.

## REST servlet Web collaboration 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



There are also REST requests available to handle actions related to Web collaboration, which are used by the co-browse and collaboration dialog widgets. The REST requests available to be used for Web collaboration actions include calls to enable a collaboration, get a collaboration status, and start a co-browsing session with a peer URI. You can also end a collaboration session, and to send and receive data to and from the peer in a co-browsing session.

## REST HTTP requests

- Each of the telephony actions has a corresponding REST HTTP request
- Information is included in the body of the request in either JSON or XML format
- Example: Make a call
  - ▶ A customer browsing a hardware store Web site selects the click to call action to connect to a customer service representative
  - ▶ The click to call widget takes the telephone number entered and sends a an HTTP request to the REST servlet on WebSphere Application Server



When making REST requests to the communication service, the request takes the form of an HTTP request. The URL of the request defines the REST call that is being made to the communication service, and information needed by the service to process the request is included as JSON or XML data. An example of a REST request that is made to the communication service is a call from the click to call widget. The click to call widget allows users of a Web site to input their telephone number into an entry field, and select a button that says call me. When they submit the request, their telephone will ring and they are connected to a customer service representative. When they input their telephone number into the entry field and press the call me button, the click to call widget assembles a REST request and sends it to the REST interface on the communication service. The REST request will indicate that a call should be established to the telephone number provided, and the communication service will extract the information and establish a call between that number and a customer service representative.



## Make a call

- This is the format of the request that is sent to the REST servlet
  - ▶ **PUT**  
**http://localhost:9080/commsvc.rest/CommServlet/call?JSON=true**
  - ▶ JSON body:  
{addressOfRecord:"sip:Customer@localhost",peerAddressOfRecord:"sip:CSR@localhost",enableCollaboration:"true"}
- Comprehensive list of supported REST calls available in the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications information center



As displayed in the example above, when a REST request is made by the click to call widget, the request is an HTTP put request that is sent to the communication service REST servlet. The URL to the servlet is shown above. After the CommServlet portion of the URL, you can see that there is a call ending to the URL, followed by a parameter JSON=true. The call portion indicates to the communication service that a request to establish a call is being made, and the parameter JSON=true indicates that the body of the response from the service will contain information about the request in JSON format. An example of the JSON body is provided above as well. You can see that the customer's telephone number, shown here in SIP format, sip customer at localhost, is included in the JSON body. The telephone number the customer should be connected to, the customer service representative's number, is included as the peer address. The communication service will gather this information, and will establish a call between the customer and the customer service representative. A response is returned to the click to call widget indicating that a call has been attempted. The service can then be polled to see if the status of the call has changed to make sure the call was established. The click to call widget can then update it's icons to allow the call to be ended, and to allow the customer and customer service representative to be joined in a co-browsing session.

## Sample REST API response

- REST calls provide responses to requests
- PUT request receives a JSON response

```
{
  "success": true,
  "infoMsg": "Call attempted between sip:Customer@localhost and sip:CSR@localhost.",
  "collaborationId": "tgr00bERd_ik5lyU6x4DtU",
  "callerAddressOfRecord": "sip:Customer@localhost",
  "calleeAddressOfRecord": "sip:Customer@localhost",
  "callServiceUri": "CommServlet/call;ibmappid=local.1235055868375_170",
  "collaborationStatus": "ESTABLISHED",
  "collaborationServiceUri":
    "CommServlet/collaborationSession;ibmappid=local.1235055868375_170",
  "forPeerCollaborationUri":
    "CommServlet/collaborationSession?addressOfRecord=tgr00bERd_ik5lyU6x4DtU",
  "eventUri": "CommServlet/event;ibmappid=local.1235055868375_170"
}
```

10

REST API

© 2009 IBM Corporation

Shown here is a sample response from the communication service in JSON format. The JSON response can be parsed and the information can be used to programmatically continue with the application processing of the communications enabled application. In the first two lines of the JSON response, you can see that a call has been attempted to be established between the customer and the customer service representative. A collaboration ID has also been generated to identify the collaboration.

## Sample requests created in JavaScript

- IBM WebSphere Application Server Feature Pack for Communications Enabled Applications information center also contains examples of creating JSON and XML requests
- JavaScript examples provided
- Requests created with JavaScript for quick reference to help you get started quickly



The IBM WebSphere Application Server Feature Pack for Communications Enabled Applications information center has a lot of information regarding building applications to use the REST APIs. There are examples provided in the information center, and sample JavaScript™ code that can be used to help build REST requests.

## *Section*

# *Summary*



This section will summarize the REST API presentation.

## Summary

- REST servlet handles HTTP requests from Web based applications
- Information for the REST request is included in the request body as JSON or XML data
- The communication service will return a response in either JSON or XML format depending on the request
- The Information center has sample JavaScript to help you build REST requests in your applications



The IBM WebSphere Application Server Feature Pack for Communications Enabled Applications communication service accepts REST calls as HTTP requests. The REST requests tell the communication service what actions to perform, and extra information to perform the actions are provided in the request body as either JSON or XML format. The communication service will return a JSON or XML response that can be parsed to gather information about the request transaction, including a collaboration ID and caller information. REST calls are available to perform telephony related actions, included establishing and ending a call. Co-browsing actions are also available using the REST interface. You can use these available REST calls to gather data from the peer, send data to the peer, and start and end a collaboration session, to name a few available actions.

## 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\\_RESTAPIs.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_CEAFP_RESTAPIs.ppt)

This module is also available in PDF format at: [../CEAFP\\_RESTAPIs.pdf](http://../CEAFP_RESTAPIs.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: 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>

JavaScript, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. 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.