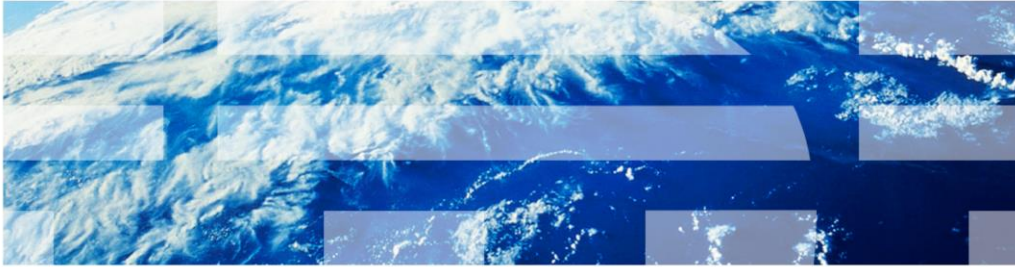


# IBM WebSphere Application Server Communications Enabled Applications

## REST API



This presentation will discuss the REST API for the Communications Enabled Applications feature of IBM WebSphere Application Server.

## 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.

# ***Overview***

This section will discuss an overview of the REST API.

## REST API

- The CEA communication services are server side services, available to drive communication in enterprise applications
- REST APIs
  - WebSphere system application invoked using REST APIs
  - Listening for requests
  - Drives CEA Web widgets

The communication services in the Communications Enabled Applications feature of WebSphere Application Server 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.

## ***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 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 here 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 its 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": "tgtr00bERd_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=tgtr00bERd_ik5lyU6x4DtU",
  "eventUri": "CommServlet/event;ibmappid=local.1235055868375_170"
}
```

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 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 information center contains extensive 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 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, including 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, just to name a few available actions.

## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

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

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.