A MITEL
PRODUCT
GUIDE

Unify OpenScape Contact Center Enterprise

Virtual Agent REST API Framework V11

Programming Guide
10/2024

A31003-S22B0-R103-01-7620

Mitel

Notices

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Europe Limited. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes. No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

Trademarks

The trademarks, service marks, logos, and graphics (collectively “Trademarks”) appearing on Mitel’s Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively “Mitel), Unify Software and Solutions GmbH & Co. KG or its affiliates (collectively “Unify”) or others. Use of the Trademarks is prohibited without the express consent from Mitel and/or Unify. Please contact our legal department at iplegal@mitel.com for additional information. For a list of the worldwide Mitel and Unify registered trademarks, please refer to the website: http://www.mitel.com/trademarks.

Contents

Contents

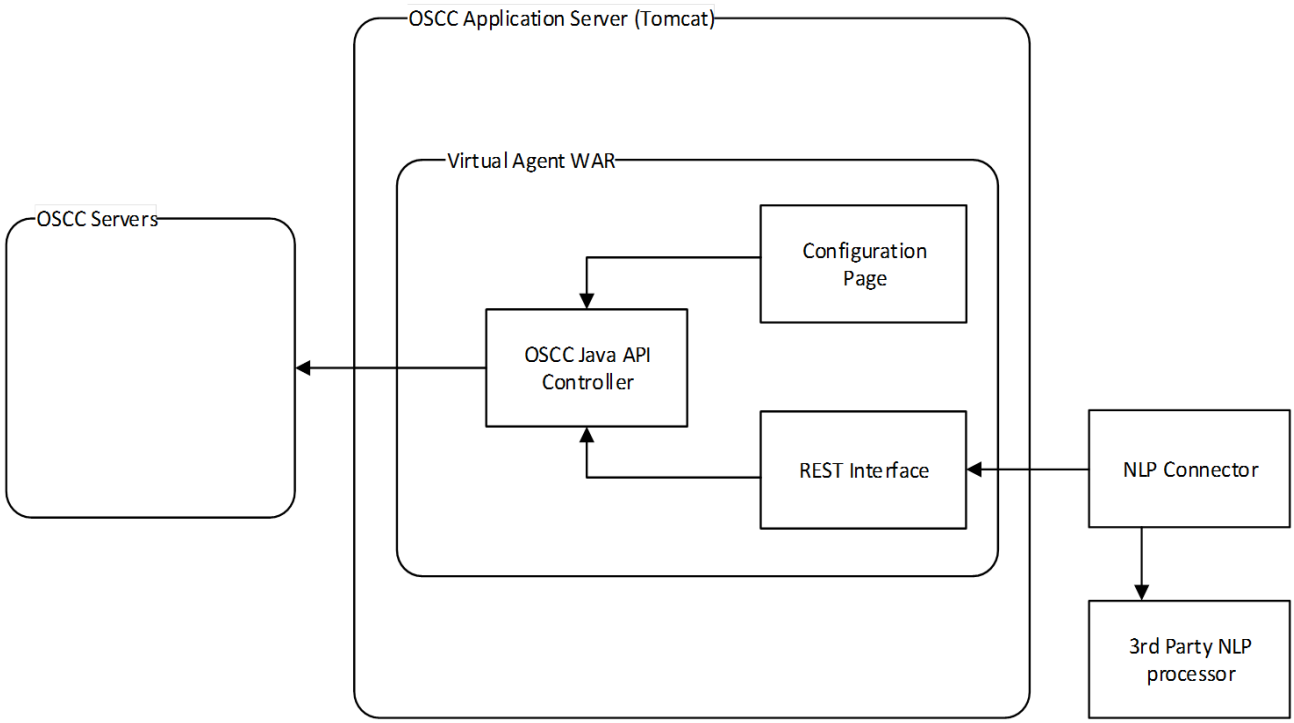
1 About the Virtual Agent Framework

The Virtual Agent Framework allows the creation of Connectors, which perform the integration of Artificial Intelligence (AI) / Natural Language Provider (NLP) providers with the OpenScape Contact Center.

The framework consists of a REST interface, which allows sending text from customer to AI/NLP providers and receive either messages to be sent to the customers or actions to be executed by the Contact Center.

1.1 Virtual Agent Framework solution overview

The figure below shows a high level overview of the Virtual Agent solution:

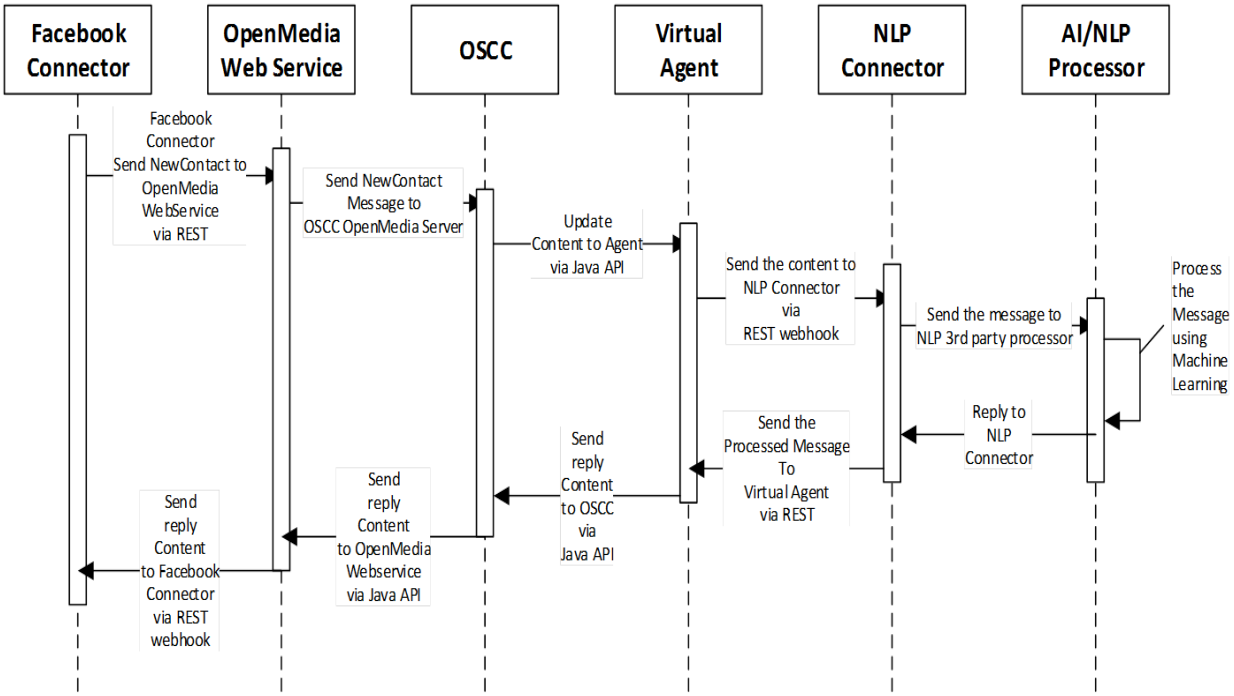


High level overview of the Virtual Agent solution

1.2 Simple Virtual Agent flow example

Below we can see a simple sequence flow that demonstrates how the entire system will integrate with each system processes.

Simple New message request from the OSCC system to AI/NLP provider.



Flow example

2 Virtual Agent Framework pre requisites

For Virtual Agent REST interface there are some steps that must be performed before using the API:

- Knowledge on REST (Representational state transfer) web services.
- Configure a Virtual Agent profile via the Web Manager.
- Configure Agent Users as Virtual Agents via the Web Manager.
- Configure Callback Actions, Re-queue Actions and External System Request Actions for the Virtual Agents.
- Install the **OpenScape Contact Center Application Server** (it can be collocated into OSCC server or into another machine).
- Use Virtual Agents with Web Chat or OpenMedia interface to interwork with the customer.

INFO: For more details about the configuration, see *OpenScape Contact Center Manager Administration Guide*.

INFO: For more details about installation of the Application Server, see *OpenScape Contact Center Installation Guide*.

3 Virtual Agent commands

Virtual Agent commands are requests sent from the NLP Connector to the Virtual Agent Service in the OpenScape Contact Center. The requests are listed below:

Command Name	HTTP Command Type	REST URL and Description
NLP Connector Registration Request	POST	**https://oscchostaddress/virtualagent/api/register** Command used to send the request to register the connector to Virtual Agent Service on OpenScape Contact Center.
Update Content Response	POST	**https://oscchostaddress/virtualagent/api/contentResponse** Command used to send the response to the content messages from the NLP Connector to the Virtual Agent Service.
Keep Alive	POST	**https://oscchostaddress/virtualagent/api/keepalive** Command used to keep the session between the Connector and the OpenScape Contact Center alive.

The following commands are sent from the Virtual Agent Service to the NLP Connector:

Command Name	HTTP Command Type	REST URL and Description
Update Content Request	POST	**https://<webhookURL>** Command used to send the request to the content messages from the NLP Connector to the Virtual Agent Service. It is configured in the webhookUrl attribute of Registration Request object.

3.1 NLP Connector Registration Request

Object used to register the Connector to the Virtual Agent Service at OpenScape Contact Center. The Connector Registration process will enable the connectivity to the Virtual Agent service, which will return the authorization session token to be used on the other REST requests to OpenScape Contact Center

NLP Connector Registration Request Object

Attribute name	Attribute type	Description
nlpConnectorName	String	This parameter has been configured in Web Manager that defines the name of the connector. The name is used to authenticate the connector during the registration process
nlpConnectorToken	String	This parameter has been automatically generated by the Virtual Agent service via the Web Manager. It is a hash value with 256 bits which is used to authenticate the connector during the registration process
webhookUrl	String	This parameter shall contain the URL to which the Virtual Agent Service at OSCC shall send the requests.

NLP Connector Registration Requeste JSON body example:

```
{     "nlpConnectorName": "NLPProvider",     "nlpConnectorToken":
"15162ec836a045b99ffb2e64bd6cbfe6",     "webhookUrl": "https://
nlpconnector:8084/dialog-flow-connector/webhook"}
```

3.2 NLP Connector Registration Response

The response after the registration request. The response is a JSON object returned synchronously by OSCC with the following data:

NLP Connector Registration Response Object

Attribute name	Attribute type	Description
sessionToken	String	This is a String value that is processed by virtual agent service and retrieved to the connector. This value must be used into the authorization header of the HTTP POST request for every connector message exchange.
errorCode	Integer	This is an Integer value that is sent by the virtual agent service to connector with the proper error code.
errorText	String	This is a is a String value that is sent by the virtual agent service to connector with the proper error text.

NLP Connector Registration Response JSON body example:

```
{     "sessionToken":
"a5e272bcfa964c6dea958b583da4e721703bb2a74bff8533a5818ccaedd71669
",     "errorCode": 0,     "errorText": "NO_ERROR",}
```

3.3 Update Content Request

Object used to send a content from the Virtual Agent service at the OpenScape Contact Center to the NLP connector

IMPORTANT: For New Contact requests, the HTTP must have the following headers:

- Content-Type = application/json
- Authorization = "session token returned by OSCC"

Update Content Request Object

Attribute name	Attribute type	Description
contactId	String	This string identifies the contact for which the messages are being exchanged.
content	String	This string has the content of the messages exchanged between the customer and the contact center.

Update Content Request JSON body Example:

```
{    "contactId": "W881220190009155318001",    "content": "Could
you please call me back in 1 hour at 4133445566?"}
```

3.4 Update Content Response

Response to UpdateContentRequest.

The response is a JSON object returned asynchronously by OpenScape Contact Center with the following data:

Update Content Response Object

Attribute name	Attribute type	Description
contactId	String	This string identifies the contact for which the messages are being exchanged.
replyContent	String	This string has the content of the messages exchanged between the customer and the contact center.
action	String	This string identifies one of the actions which were configured in the Web Manager
parameters	Map<String,String>	This is a list of key + value objects which identify parameters to be used for the execution of the action
quickReplies	List<String>	FUTURE: This list of strings contain a list of options to be selected by the customer

Update Content Response JSON body Example:

```
This is an example of a simple content response to the customer:{
"contactId": "W881220190009155318001",    "replyContent":"What is
the version of your system?",     "action": "",     "parameters":
{},     "quickReplies": []}This is an example of a callback to a
customer:{     "contactId": "W881220190009155318001",
"replyContent":"We will call you back.",     "action": "Callback
customer",   "parameters": {     "phonenumber": "4133445566",
"time": "17:00BRT"     },     "quickReplies": []}
```

3.5 Keep Alive

The HTTP REST command is used to keep the connection between the Connector and the OpenScape Contact Center alive.

IMPORTANT: In the Keep Alive request the HTTP request must be sent with the following headers:

- Content-Type = application/json
- Authorization = "<session token returned by OSCC>"

NOTICE: Send an empty JSON object

Keep Alive JSON body example:

```
{}
```

3.6 Keep Alive Response

Response after every request. The response is a JSON object returned synchronously by OpenScape Contact Center with the following data:

Keep Alive Response Object

Attribute name	Attribute type	Description
type	String	Indicates the type of the response. The string value supported by the OpenScape Contact Center is: "keepAlive".
errorCode	Integer	Indicates the returned error code number. See error code enum definition below.
errorText	String	Indicates the returned error code text.See error code enum definition below.

Keep Alive Response JSON body example:

```
{     "type": "keepAlive",     "errorCode": 0,     "errorText":
"NO_ERROR"}
```

4 Error Code

All the error codes and error text are set as follow:

Error Codes

Number	Text
0	NO_ERROR
1	GENERAL_ERROR
2	ALREADY_REGISTERED_ON
4	WRONG_TITLE_OR_TOKEN_NOT_FOUND
6	AUTH_STATEMENT_NOT_VALID
7	NOT_READY
8	NOT_ENABLED
9	CONNECTOR_BLOCKED
10	CONNECTOR_NOT_REGISTERED
11	WRONG_API_REQUEST
12	STRING_OUT_OF_BOUND
13	WRONG_DATE_FORMAT

Index