



NewgenONE OmniOMS

Developer Guide

Version: 12.0

Disclaimer

This document contains information proprietary to Newgen Software Technologies Ltd. User may not disclose or use any proprietary information or use any part of this document without written permission from Newgen Software Technologies Ltd.

Newgen Software Technologies Ltd. makes no representations or warranties regarding any software or to the contents or use of this guide. It also specifically disclaims any express or implied warranties of merchantability, title, or fitness for any particular purpose. Even though Newgen Software Technologies Ltd. has tested the hardware and software and reviewed the documentation, it does not guarantee or imply that this document is error free or accurate regarding any particular specification. As a result, this product is sold as it is and user, the purchaser, is assuming the entire risk as to its quality and performance. Further, Newgen Software Technologies Ltd. reserves the right to revise this publication and make changes in its content without any obligation to notify any person, of such revisions or changes. Newgen Software Technologies Ltd. authorizes no Newgen agent, dealer or employee to make any modification, extension, or addition to the above statements.

Newgen Software Technologies Ltd. has attempted to supply trademark information about company names, products, and services mentioned in this document. Trademarks indicated below were derived from various sources.

Copyright © 2024 **Newgen Software Technologies Ltd.** All Rights Reserved.
No part of this publication may be reproduced and distributed without the prior permission of Newgen Software Technologies Ltd.

Newgen Software, Registered Office, New Delhi

E-44/13

Okhla Phase - II

New Delhi 110020

India

Phone: +91 1146 533 200

info@newgensoft.com

Contents

Preface	9
Revision history	9
About this guide.....	9
Intended audience	9
Related documents	10
Documentation feedback	10
REST web services	11
OMS session management (using connectCabinet).....	11
Sample input XML.....	12
Input parameters description	13
Sample output XML	15
Output parameters description.....	16
Error code description	16
OMS session management (using disconnectCabinet).....	17
Sample input XML.....	17
Input parameters description	18
Sample output XML	19
Output parameters description.....	19
Error code description	19
OMS document generation (using DB)	19
Sample input XML.....	20
Input parameters description	21
Sample output XML	25
Output parameters description.....	25
Error code description	26
OMS document generation (using XML)	27
Sample input XML.....	28
Input parameters description	29
Sample output XML	33
Output parameters description.....	34
Error code description	35
OMS document generation (using JSON)	36
Sample input XML.....	37
Input parameters description	38
Sample output XML	42
Output parameters description.....	42
Error code description	44
OMS job auto approval	45
Sample input XML.....	45
Input parameters description	46

Sample output XML	48
Output parameters description.....	48
Error code description	48
OMS message tracker	49
Sample input XML.....	50
Input parameters description	50
Sample output XML	51
Output parameters description.....	51
Error code description	52
OMS on demand XML validation	52
Sample input XML.....	53
Input parameters description	53
Sample output XML	53
Output parameters description.....	54
Error code description	54
OMS communication tracking.....	54
Sample input XML.....	55
Input parameters description	55
Sample output XML	56
Output parameters description.....	58
Error code description	58
OMS on demand data purge	59
Sample input XML.....	59
Input parameters description	60
Sample output XML	61
Output parameters description.....	61
Error code description	61
OMS retrieve details (for getting OmniDocs details).....	62
Sample input XML.....	62
Input parameters description	62
Sample output XML	63
Output parameters description.....	63
Error code description	63
OMS retrieve details (for decoding user data).....	64
Sample input JSON	64
Input parameters description	65
Sample output JSON.....	65
Output parameters description.....	65
Error code description	66
OMS secured DB credentials.....	66
Sample input JSON	67
Input parameters description	67
Sample output JSON.....	67
Output parameters description.....	67
Error code description	68

OMS secured data encryption.....	68
Sample input XML.....	68
Input parameters description	69
Sample output XML	69
Output parameters description.....	69
Error code description	70
OMS external content library	70
Adding external content.....	71
Sample input.....	71
Input parameters description	72
Sample output	73
Output parameters description.....	73
Error code description	73
Modifying external content	76
Sample input.....	76
Input parameters description	77
Sample output	78
Output parameters description.....	78
Error code description	79
Deleting external content.....	80
Sample input.....	80
Input parameters description	81
Sample output	81
Output parameters description.....	81
Error code description	82
Fetching external content.....	82
Sample input.....	83
Input parameters description	83
Sample output	83
Output parameters description.....	84
Error code description	84
APIs for CCM integration.....	85
OMS search templates.....	85
Sample input XML	85
Input parameters description	86
Sample output XML.....	86
Output parameters description.....	87
Error code description	88
OMS view template	88
Sample input XML	89
Input parameters description	89
Sample output XML.....	89
Output parameters description.....	90
Error code description	90
OMS get cycle job list	90
Sample input XML	91

Input parameters description	91
Sample output XML.....	91
Output parameters description	92
Error code description	92
OMS get dataset schemas.....	93
Sample input XML	93
Input parameters description	93
Sample output XML.....	94
Output parameters description	94
Error code description	95
Custom theme branding	95
addBrandingStyle.....	95
Sample input XML	96
Input parameters description	96
Sample output XML.....	97
Output parameters description	97
Error code description	97
modifyBrandingStyle	98
Sample input XML	98
Input parameters description	99
Sample output XML.....	99
Output parameters description	100
Error code description	100
deleteBrandingStyle.....	100
Sample input XML	101
Input parameters description	101
Sample output XML.....	101
Output parameters description	101
Error code description	102
fetchBrandingStyle.....	102
Sample input XML	102
Input parameters description	103
Sample output XML.....	103
Output parameters description	103
Error code description	104
SOAP web services	105
OMS session management (using connectCabinet).....	105
Sample input XML.....	105
Input parameters description	106
Sample output XML	107
Output parameters description.....	107
Error code description	107
OMS session management (using disconnectCabinet).....	108
Sample input XML.....	108
Input parameters description	109
Sample output XML	109

Output parameters description.....	109
Error code description	110
OMS document generation (using DB)	110
Sample input XML.....	111
Input parameters description	111
Sample output XML	113
Output parameters description.....	114
Error code description	115
OMS document generation (using XML)	116
Sample input XML.....	116
Input parameters description	117
Sample output XML	119
Output parameters description.....	120
Error code description	121
OMS document generation (using JSON)	122
Sample input XML.....	122
Input parameters description	123
Sample output XML	125
Output parameters description.....	126
Error code description	127
OMS job auto approval	128
Sample input XML.....	128
Input parameters description	129
Sample output XML	130
Output parameters description.....	130
Error code description	131
OMS message tracker	131
Sample input XML.....	132
Input parameters description	132
Sample output XML	133
Output parameters description.....	133
Error code description	134
OMS on demand XML validation	134
Sample input XML.....	134
Input parameters description	135
Sample output XML	135
Output parameters description.....	136
Error code description	136
OMS on demand data purge	136
Sample input XML.....	137
Input parameters description	137
Sample output XML	138
Output parameters description.....	138
Error code description	139

Implementation of Authentication Token and pre-Hook in OmniOMS REST Web

Services	140
Prerequisites	140
Implementing Authentication Token	140
Implementation steps	140
Execution impact and expectations.....	141
Implementing pre-Hook	142
Requirement of classes and components.....	142
Implementation steps	143
Custom tab integration.....	145
Integrating custom tabs in OmniOMS Admin Workspace.....	145
Configuring session validity	147
Implementing URL shortening.....	151
Integrating Facebook and Twitter channels.....	153
getAccessToken.....	153
Sample input JSON	153
Input parameters description	154
Sample output JSON.....	154
Output parameters description.....	154
Error code description	155
Configuring message and archival jobs using REST services.....	156
Configure Message jobs.....	156
Sample Message tester.....	157
Configure Archival jobs	158
Sample Archival tester	159
Configuring cobranding in OmniOMS	161
Adding play button image in email template.....	163

Preface

This section provides information about the revision history, about this guide, details on the intended audience, related documents, and documentation feedback for this guide.

Revision history

Revision date	Description
October 2024	Initial publication

About this guide

This guide provides information on the SOAP and REST web services used in OmniOMS. It also provides information on how to implement Authentication Token and pre-Hook in OmniOMS REST web services. Additionally, it provides information on integrating custom tabs, implementing URL shortening, configuring message and archival jobs using REST services, configuring co-branding in OmniOMS web modules, and adding a play button image in an Email template.

To ensure you are referring to the latest and most recent revision of this guide, download it from one of the following locations:



- [Newgen Internal Doc Portal](#), if you are a Newgen employee.
- [Newgen Partner Portal](#), if you are a Newgen partner.

Intended audience

This guide is intended for systems administrators and developers seeking information on the installation, configuration, maintenance, and development of the OmniOMS application. The reader must have moderate to high level knowledge of working with OmniOMS and be comfortable in developing the integration scenarios.

Related documents

The following documents provide additional information about the NewgenONE Developer Guide:

- NewgenONE OmniOMS Release Notes
- NewgenONE OmniOMS Administration User Guide
- NewgenONE OmniOMS Composition Designer User Guide
- NewgenONE OmniOMS Admin Workspace User Guide
- NewgenONE OmniOMS Communication Editor User Guide

Documentation feedback

To provide feedback or any improvement suggestions on technical documentation, write an email to docs.feedback@newgensoft.com.

To help capture your feedback effectively, share the following information in your email:

- Document name
- Version
- Chapter, topic, or section
- Feedback or suggestions

REST web services

This section describes the REST web services used in OmniOMS.



In case OAuth is enabled, send only the `appId`, `accessToken`, and `sessionValid`; do not include the `sessionId`. However, if OAuth is disabled, include only the `sessionId`.

OMS session management (using connectCabinet)

OMSSessionManagement class provides REST web service to establish and terminate a session with OmniOMS.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSSessionManagement/connectCabinet*
- **Services Method:**
 - **connectCabinet** — The purpose of this method is to take user credentials and connect the user returning *sessionId*.
 - **Parameters** — *OMSConnectCabinetInputBO* represents input details for establishing a session in OmniOMS. Request Object providing details for establishing a session with OmniOMS via service method '*connectCabinet*'.
 - **Returns** — *OMSConnectCabinetOutputBO* - provides the details of successful/unsuccessful sessions in OmniOMS.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

OMSCConnectCabinetInputBO

Case 1: if OAuth is enabled.

```
<OMSCConnectCabinetInputBO>
  <cabinetName>oms</cabinetName>
  <userName>abd</userName>
  <userPassword>ne123</userPassword>
  <connectForcefully>Y</connectForcefully>
  <encrFlag>Y</encrFlag>
  <passAlgoType>1</passAlgoType>
  <authToken>xxxx</authToken>
  <authTokenType>JWT</authTokenType>
  <appId>2b67b8a0-c784-4989-a474-8967861e4761</appId>
  <sessionExpiryTime>1440</sessionExpiryTime>
</OMSCConnectCabinetInputBO>
```

Case 2: if OAuth is disabled.

```
<OMSCConnectCabinetInputBO>
  <cabinetName>oms</cabinetName>
  <userName>abd</userName>
  <userPassword>ne123</userPassword>
  <connectForcefully>Y</connectForcefully>
  <encrFlag>Y</encrFlag>
  <passAlgoType>1</passAlgoType>
  <authToken>xxxx</authToken>
  <authTokenType>JWT</authTokenType>
</OMSCConnectCabinetInputBO>
```

Input parameters description

OMSCoconnectCabinetInputBO

Name	Mandatory (Yes/No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet in which session is to be established.	NA	NA
connectForcefully	Yes	Can have value either Y or N. Y - if the user wants to terminate the previously established connection with OmniOMS and create a new one. N - if the user does not want to terminate the previously established connection with OmniOMS.	'Y' or 'N'	'N'
userName	Yes	User Id of the user, who needs session.	NA	NA
userPassword	Yes	Password corresponding to user-id.	NA	NA

Name	Mandatory (Yes/No)	Description	Valid values	Default value
encrFlag	No	On the basis of this flag value as Y or N, password encryption will be applied.	NA	N
PassAlgoType	No	Its value can be 1 = MD5, 2-SHA 256, 3 = SHA 512. On the basis of the value, encrypted password needs to be passed.	NA	0
authToken	No	Authenticating the user before executing the API calls requires an authorization token. The user must input the authToken which then gets validated as per client implementation.	xxxx	-
authTokenType	No	To be implemented by implementation team. authToken is required to implement this parameter.	JWT, OAUTH	-

Name	Mandatory (Yes/No)	Description	Valid values	Default value
appid	No	Shows the app ID registered by the user. This parameter is required when OAuth is enabled.	Valid app ID	NA
sessionExpiryTime	No	Receives the time in minutes from the user in ConnectCabinet API, indicating the user's desired login duration in minutes for the application. This parameter is required when OAuth is enabled.	Valid session expiry time	NA

Sample output XML

OMSCConnectCabinetOutputBO

```

<OMSCConnectCabinetOutputBO>
  <description/>
  <sessionId>-12345</sessionId>
  <status>0</status>
</OMSCConnectCabinetOutputBO>

```

Output parameters description

OMSCoconnectCabinetOutputBO

Name	Description
description	Success
sessionId	The unique number is generated for each request.
status	0
accessDetails	This tag contains the following tags: • sessionValid — Validates the session expiry time. • accessToken — Validates a token provided by the user. • appld — Provides the app ID registered by the user.

Error code description

Code	Message	Comment
1	Unable to connect to the server, check logs.	Check the logs to rectify the server connection failure.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

OMS session management (using disconnectCabinet)

OMSSessionManagement class provides REST web service to establish and terminate a session with OmniOMS.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSSessionManagement/disconnectCabinet*
- **Services Method:**
 - **disconnectCabinet** — The purpose of this method is to remove valid sessions within OmniOMS.
 - **Parameters** — *OMSDisconnectCabinetInputBO* represents the input details for terminating a session with OmniOMS.
 - **Returns** — *OMSDisconnectCabinetOutputBO* provides the details of successful or unsuccessful requests processed.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

OMSDisconnectCabinetInputBO

Case 1: if OAuth is enabled.

```
<OMSDisconnectCabinetInputBO>
  <cabinetName>oms</cabinetName>
  <sessionId>12345</sessionId>
  <appId>2b67b8a0-c784-4989-a474-8967861e4761</appId>
  <sessionValid>HCciLTcMHFI6zt5MZJdkJCYqfPgoHN9K1C
/qlul0+0hAEM8r3WyrVyvFTpyAM2E+tntK4XIj5TmRbwo+RYe
/uvDaZqz6wToVQ+pxTtXNKqZ2OJ8sM7CDkyargirVgTwWj
/rLNf3HSjsID0g/+L2xoazu6h5LMUam1bk7KjCylkg4Y+
fCUptfP65oPcP7x+dSWybTghpxrMN+6yAqRko5Zkg0BnpLRH
/A6GhniSOYZptkCBdoF56K4mAN3sEj2kjsxDKWQzMDK4FFeUQOEJ7vM
/iHoY4yRRi5uC6EST7vj5P8gpEicBmSSRun0xcghm7U5JlzPTSQffw
sUK31lYEY7AQ==</sessionValid>
  <accessToken>eyJhbGciOiJIUzUxMiJ9.eyJh
```

```

cHBJZCI6IjJiNjdiOGEwLWM3ODQtNDk4OS1hNDc
0LTg5Njc4NjFlNDc2MSIsIlRva2VuRXhwaXJ5VGlt
ZSI6IklqQXlOQzB3TWkwd09WUXdOem93TlRveU5DNHhNe
1l3TVRnME5EWT0iLCJ0b2t1bk1kIjoioGE4MWJlNDYtZTYw
NC00YzQ2LWFKYjQtYjYwOTg4YTQyNzQ0Iiwic3ViIjoI
MmI2N2I4YTAtYzc4NC00OTg5LWE0NzQtODk2Nzg2MWU
0NzYxIiwianRpIjoioGE4MWJlNDYtZTYwNC00YzQ2L
WFKYjQtYjYwOTg4YTQyNzQ0IiwiaWF0IjoxNzA3Mzc1N
jIzLCJleHAiOjE3MDc0NjIzMjR9.HOD0622TAExVe_
swmEGvvyYGeaZmFCflnLMz9xrdN1hImxuyQiCFyUM4IwQX
kDm
</accessToken>
</OMSDisconnectCabinetInputBO>

```

Case 2: if OAuth is disabled.

```

<OMSDisconnectCabinetInputBO>
  <cabinetName>oms</cabinetName>
  <sessionId>12345</sessionId>
</OMSDisconnectCabinetInputBO>

```

Input parameters description

OMSDisconnectCabinetInputBO

Name	Mandatory (Yes/No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet in which session is to be established.	NA	NA
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA
sessionValid	Yes (in case OAuth is enabled)	Validates the session expiry time.	Valid sessionValid value	NA
accessToken	Yes (in case OAuth is enabled)	Validates a token provided by the user.	Valid accessToken	NA
appld	Yes (in case OAuth is enabled)	Provides the app ID registered by the user.	Valid app ID	NA

Sample output XML

SDisconnectCabinetOutputBO

```
<OMSDisconnectCabinetOutputBO>
  <description>Success</description>
  <status>0</status>
</OMSDisconnectCabinetOutputBO>
```

Output parameters description

OMSDisconnectCabinetOutputBO

Name	Description
description	success
status	0

Error code description

Code	Message	Comment
1	Unable to connect to the server, check logs.	Check the logs to rectify the server connection failure.
11	Invalid Session.	This message appears when an invalid session is passed in the input parameters.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

OMS document generation (using DB)

OMSDocumentGeneration class provides a REST web service that allows integration with other party components for generating documents and sending communications using OmniOMS.

- **Service URL:** `http://<IP>:<Port>/OMSWebServices/rest/OMSDocumentGeneration/processUsingDB`
- **Services Method:**
 - **processUsingDB** — This method is to generate documents that are based on a database dataset.
 - **Parameters** — *DocumentGenerationDBInputBO* - Object providing input for document generation based on database dataset. Request Object to receive input for document generation from service method 'processUsingDB'.
 - **DocumentGenerationDBOutputBO** — Object providing the details of the document generation request processed. Response object of document generation request from service method 'processUsingDB'.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

DocumentGenerationDBInputBO

```
<DocumentGenerationDBInputBO>
  <attachmentsLocation></attachmentsLocation>
  <cabinetName>oms</cabinetName>
  <dmrqId></dmrqId>
  <doArchive>Y</doArchive>
  <encodedPDFRequired>N</encodedPDFRequired>
  <fileName>testingFile</fileName>
  <filePath>D:/Files</filePath>
  <isDmrq></isDmrq>
  <productName>Prod</productName>
  <masterKey>1</masterKey>
  <reportVersion>0</reportVersion>
  <sendMail>Y</sendMail>
  <sessionId>423424</sessionId>
  <synchronousArchival>N</synchronousArchival>
  <authToken>xxxx</authToken>
  <authTokenType>JWT</authTokenType>
  <brandId>1</brandId>
</DocumentGenerationDBInputBO>
```

Input parameters description

DocumentGenerationDBInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
attachmentsLocation	No	Filepath of the External Document to be sent as an additional attachment in the Email.	NA	NA
cabinetName	Yes	Name of the cabinet where document layout is checked-in.	NA	NA
communicationId	No	Unique communication Id for the specific transaction.	NA	0
dateFrom	No	Date from which records will be picked for document generation.	NA	NA
dateTo	No	Date till which records will be picked for document generation.	NA	NA
dmrqId	No	If isDmrq is provided with value 'Y', then this 'dmrqId' will be considered as the masterkey.	NA	NA

Name	Mandatory (Yes or No)	Description	Valid values	Default value
isDmrq	No	To specify whether user data, for which document is to be generated is in the database tables mapped in the dataset or not. Can have value Y/N. Y - User data for generation will be picked from a different table. Example, if the mapped table is 'SAMPLE', then data will be picked from 'SAMPLE_DMRQ' N - User data for generation will be picked from the mapped table in the dataset.	'Y' or 'N'	'N'
doArchive	Yes	Can have value either Y or N. Y - if the document is to be queued for archival. N - if not required.	'Y' or 'N'	'N'
productName	Yes	Unique for each on-demand template.	NA	NA
encodedPDFRequired	No	Can have value either Y or N. Y - Document will be returned as a binary string but will be Base64 encoded. N - No binary form of generated document will be provided.	'Y' or 'N'	'N'

Name	Mandatory (Yes or No)	Description	Valid values	Default value
fileName	No	File Name of the document generated, along with the provided name a unique id will be added by CCM. Example, 'filename_requestId'.	NA	If fileName is blank then fileName = 'CurrentTimeInMilliseconds_requestId' otherwise fileName = 'fileName_RequestId'
filePath	No	File location where the document needs to be generated.	NA	NA
masterKey	Yes	ID/Primary Key of the user for which document is to be generated, this value will be used to distinguish users.	NA	NA
reportVersion	Yes	By default, or not provided then a recently base-lined layout of the document will be used and generated. For a specific layout, the version needs to be provided	NA	0
sendMail	Yes	For both Mail and SMS processes, this tag is used based upon the report type of the productname passed. Can have value either Y or N. Y - if the document is to be queued for sending mail or SMS. N - if not required.	'Y' or 'N'	'N'

Name	Mandatory (Yes or No)	Description	Valid values	Default value
sendWhatsapp	Yes	Can have value either Y or N. Y - if the document is to be queued for sending WhatsApp messages. N - if not required.	'Y' or 'N'	'N'
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA
synchronousArchival	No	This will be used for ondemand synchronous archival only for generation cases. For all other cases, archival will be asynchronous only.	NA	N
authToken	No	Authenticating the user before executing the API calls requires an authorization token. The user must input the authToken which then gets validated as per client implementation.	xxxx	-
authTokenType	No	To be implemented by implementation team. authToken is required to implement this parameter.	JWT, OAUTH	-
brandId	No	Brand ID for Theme Branding	Provide a valid Brand ID	-

Sample output XML

DocumentGenerationDBOutputBO

```
<DocumentGenerationDBOutputBO>
  <communicationId/>
  <description/>
  <dmrqId/>
  <encodedFileData/>
  <filePathName>D:\Files\testingFile_116956.pdf</filePathName>
  <latestBaselinedVersion/>
  <mailBodyPathName/>
  <requestId>116956</requestId>
  <status>0</status>
  <archivalDescription>Success</archivalDescription>
  <archivalDetails>
    <documentName>fdgjksdhjk</documentName>
    <documentIndex>3</documentIndex>
    <volumeIndex>4</volumeIndex>
    <imageIndex>10</imageIndex>
  </archivalDetails>
</DocumentGenerationDBOutputBO>
```

Output parameters description

DocumentGenerationDBOutputBO

Name	Description
communicationId	Same communication Id that has passed in the input XML.
description	It will provide the description corresponding to status, whether the request was processed successfully or not.
dmrqId	DmrqId provided in input for the user, will be returned.
encodedFileData	Base64 encoded binary-form of the generated document.
filePathName	The absolute path of the generated document (non-html).
latestBaselinedVersion	Will return Y/N. Y - Document is generated from the latest base-lined layout. N - The latest base-lined layout is not used for document generation.

Name	Description
mailBodyPathName	The absolute path of the '.html' file, which has the message that will be communicated to the end-user.
requestId	For every generation request, a unique Id generated by CCM.
status	Code to ensure whether generation was successful or not. 0 - successfully generated. Non-Zero - some error occurred while processing the request.
archivalDescription	Failed for failure cases, Success for successful archival cases
archivalDetails	This tag will return document name, volumeIndex , documentIndex, and ImageIndex values.

Error code description

Code	Message	Comments
1027	Invalid product name.	The product name is invalid.
1029	Document generation failed. Document type for the given product name is not supported.	Document type for the given product name is not supported.
1028	Either OnDemand Job is not specified for the ProductName, or details for the concerned operation(mail/archival) are incomplete.	Either OnDemand Job is not specified for the ProductName, or details for the concerned operation(mail/archival) are incomplete.
1012	The server is busy and unable to provide a Statement object.	The server is busy and unable to provide a Statement object.
11	Invalid Session.	This message appears when an invalid session is passed in the input parameters.
1034	Synchronous archival will work only for generation cases.	Synchronous archival works only for generation cases.

Code	Message	Comments
1035	Synchronous archival will not work for other dms cases.	Synchronous archival does not work for other DMS cases.
-1011	CabinetName not found	The cabinet name is invalid.
1037	Invalid REST token	The entered REST token is invalid.
1038	Domain user not found	The domain user is not found.
1039	REST TokenParser not implemented	The REST TokenParser parameter is enabled in the <i>OMSAdminConfig.ini</i> file. However, it is not implemented yet.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

OMS document generation (using XML)

OMSDocumentGeneration class provides a REST web service that allows integration with other party components for generating documents and sending communications using OmniOMS.

- **Service URL:** `http://<IP>:<Port>/OMSWebServices/rest/OMSDocumentGeneration/processUsingXML`

- **Services Method:**

- **processUsingXML** — Method is to generate documents that are based on an XML dataset.

- **Parameters** — DocumentGenerationXMLInputBO - Object providing input for document generation based on XML dataset. Request Object to receive input for document generation from service method 'processUsingXML'.

- **Returns** — DocumentGenerationXMLOutputBO - Object providing the details of the document generation request processed. Response object of document generation request from service method 'processUsingXML'.

- **Method Type:** POST

- **Consumes:** application/xml

- **Produces:** application/xml

Sample input XML

DocumentGenerationXMLInputBO

```
<DocumentGenerationXMLInputBO>
  <attachmentsLocation></attachmentsLocation>
  <cabinetName>oms</cabinetName>
  <doArchive>Y</doArchive>
  <encodedPDFRequired>N</encodedPDFRequired>
  <fileName>sampltest</fileName>
  <filePath>F:\pdfs</filePath>
  <productName>OD</productName>
  <reportVersion>0</reportVersion>
  <sendMail>N</sendMail>
  <sessionId>1234</sessionId>
  <skipApproval>Y</skipApproval>
  <skipEditing>Y</skipEditing>
  <xmlData>&lt;Root&gt;&lt;/Root&gt;</xmlData>
  <synchronousArchival>N</synchronousArchival>
  <authToken>xxxx</authToken>
  <authTokenType>JWT</authTokenType>
  <brandId>1</brandId>
  <sendTwitter>N</sendTwitter>
  <sendFacebook>N</sendFacebook>
</DocumentGenerationXMLInputBO>
```



Use this URL to get ESCAPED HTML to pass within <xmlData> tag - <https://www.freeformatter.com/xml-escape.html#ad-output>

Input parameters description

DocumentGenerationXMLInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
attachmentsLocation	No	Filepath of the External Document to be sent as an additional attachment in the Email.	NA	NA
cabinetName	Yes	Name of the cabinet where document layout is checked-in.	NA	NA
communicationId	No	Unique communication Id for the specific transaction.	NA	0
dateFrom	No	The date from which records are picked for document generation.	NA	NA
dateTo	No	The date until which records are picked for document generation.	NA	NA
doArchive	Yes	Can have value either Y or N. Y - if the document is to be queued for archival. N - if not required.	'Y' or 'N'	'N'

Name	Mandatory (Yes or No)	Description	Valid values	Default value
encodedPDFRequired	No	Can have value either Y or N. Y - Document will be returned as a binary string but will be Base64 encoded. N - No binary form of the generated document will be provided.	'Y' or 'N'	'N'
filename	No	File Name of the document to be generated, along with the provided name a unique id will be added by CCM. Example, 'filename_requestId'.	NA	If fileName is blank then fileName = 'CurrentTimeInMilliseconds_requestId' otherwise fileName = 'fileName_RequestId'
filePath	No	File location where the document needs to be generated.	NA	NA
productName	Yes	Unique for each on-demand XML dataset for the template.	NA	NA
reportVersion	Yes	By default, or not provided then the recently base-lined layout of the document will be used and generated. For a specific generation, a layout version needs to be provided.	NA	0
saveClientData	No	If Y: insert the xmlData provided into inputDataXML column of omsJobQueueItems table.	'Y' or 'N'	'N'

Name	Mandatory (Yes or No)	Description	Valid values	Default value
sendMail	Yes	For both Mail and SMS processes, this tag is used based upon the report type of the productName passed. Can have value either Y or N. Y - if the document is to be queued for sending mail or SMS. N - if not required.	'Y' or 'N'	'N'
sendWhatsapp	Yes	Can have value either Y or N. Y - if the document is to be queued for sending WhatsApp messages. N - if not required.	'Y' or 'N'	'N'
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA
skipApproval	Yes	Useful only for Interactive Editor flow. Y - if the document does not need to go through the maker-checker process. N - if the document needs to go through the maker-checker process.	'Y' or 'N'	'Y'
skipEditing	Yes	Useful only for Interactive Editor flow. Y - if the document will not be allowed for further editing after generation. N - if the document will be allowed to go through editing after generation.	'Y' or 'N'	'Y'

Name	Mandatory (Yes or No)	Description	Valid values	Default value
xmlData	Yes	Pass User data in ESCAPED HTML format for which the document is to be generated. Use this URL to get ESCAPED HTML to pass within tag - https://www.freeformatter.com/xml-escape.html#ad-output	NA	NA
synchronousArchival	No	This will be used for ondemand synchronous archival only for generation cases. For all other cases, archival will be asynchronous only.	'Y' or 'N'	'N'
authToken	No	Authenticating the user before executing the API calls requires an authorization token. The user must input the authToken which then gets validated as per client implementation.	xxxx	-
authTokenType	No	To be implemented by implementation team. authToken is required to implement this parameter.	JWT, OAUTH	-
brandId	No	Brand ID for Theme Branding	Provide a valid Brand ID	-

Name	Mandatory (Yes or No)	Description	Valid values	Default value
sendTwitter	Yes	Can have value either Y or N. • Y - if the document is to be queued for sending Twitter Communciation. • N - if not required.	'Y' or 'N'	'N'
sendFacebook	Yes	Can have value either Y or N. • Y - if the document is to be queued for sending Facebook Communciation. • N - if not required.	'Y' or 'N'	'N'

Sample output XML

DocumentGenerationXMLOutputBO

```
<DocumentGenerationXMLOutputBO>
  <communicationId/>
  <description/>
  <encodedDocxData/>
  <encodedFileData/>
  <encodedMailBodyData/>
  <filePathName>D:\Files\sampltest_116956.pdf</filePathName>
  <latestBaselinedVersion/>
  <mailBodyPathName/>
  <requestId>116956</requestId>
  <status>0</status>
  <transactionId>1567</transactionId>
  <archivalDescription>Success</archivalDescription>
  <archivalDetails>
    <documentName>fdgjksdhjk</documentName>
    <documentIndex>3</documentIndex>
    <volumeIndex>4</volumeIndex>
    <imageIndex>10</imageIndex>
  </archivalDetails>
</DocumentGenerationXMLOutputBO>
```

Output parameters description

DocumentGenerationXMLOutputBO

Name	Description
communicationId	Same communication Id that has passed in the input XML.
description	Will provide the description corresponding to status, whether the request was processed successfully or not.
encodedDocxData	Base64 encoded binary-form of generated docx document(non-html).
encodedFileData	Base64 encoded binary-form of generated PDF document(non-html).
encodedMailBodyData	Base64 encoded binary-form of the generated document (html only).
filePathName	The absolute path of the generated document(non-html).
latestBaselinedVersion	Will return Y/N. Y - The document is generated from the latest base-lined layout. N - The latest base-lined layout is not used for document generation.
mailBodyPathName	The absolute path of the '.html' file, which has the message that will be communicated to the end-user.
requestId	For every generation request, a unique ID is generated by CCM.
status	Code to ensure whether the generation was successful or not. 0 - successfully generated. Non-Zero - Some error occurred while processing the request.
transactionId	Will return a numeric ID of the generated document for contract and letter only.
archivalDescription	Failed for failure cases, Success for successful archival cases
archivalDetails	This tag will return the following values: • documentName — The name of the document. • volumeIndex — The volume index of the attachment.

Name	Description
	- documentIndex — A unique index of the document that the user wants to attach. - imageIndex — The image index of the attachment.

Error code description

Code	Message	Comment
1027	Invalid product name.	The product name is invalid.
1029	Document generation failed. Document type for the given product name is not supported.	Document type for the given product name is not supported.
1028	Either OnDemand Job is not specified for the ProductName, or details for the concerned operation(mail/archival) are incomplete.	Either OnDemand Job is not specified for the ProductName, or details for the concerned operation(mail/archival) are incomplete.
1012	The server is busy and unable to provide a Statement object.	The server is busy and unable to provide a Statement object.
11	Invalid Session.	This message appears when an invalid session is passed in the input parameters.
1034	Synchronous archival will work only for generation cases.	Synchronous archival works only for generation cases.
1035	Synchronous archival will not work for other dms cases.	Synchronous archival does not work for other DMS cases.
-1011	CabinetName not found	The cabinet name is invalid.
1037	Invalid REST token	The entered REST token is invalid.
1038	Domain user not found	The domain user is not found.

Code	Message	Comment
1039	REST TokenParser not implemented	The REST TokenParser parameter is enabled in the <i>OMSAdminConfig.ini</i> file. However, it is not implemented yet.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

OMS document generation (using JSON)

OMSDocumentGeneration class provides a REST web service that allows integration with other party components for generating documents and sending communications using OmniOMS.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSDocumentGeneration/processUsingJSON*
- **Services Method:**
 - **processUsingJSON** — Method is to generate documents that are based on JSON dataset.
 - **Parameters** — *DocumentGenerationJSONInputBO* - Object providing input for document generation based on JSON dataset. Request Object to receive input for document generation from service method 'processUsingJSON'.
 - **Returns** — *DocumentGenerationJSONOutputBO* - Object providing the details of document generation request processed. Response object of document generation request from service method 'processUsingJSON'.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

DocumentGenerationJSONInputBO

```
<DocumentGenerationJSONInputBO>
  <attachmentsLocation></attachmentsLocation>
  <cabinetName>oms</cabinetName>
  <doArchive>Y</doArchive>
  <encodedPDFRequired>N</encodedPDFRequired>
  <fileName>sampltest</fileName>
  <filePath>F:\pdfs</filePath>
  <productName>OD</productName>
  <reportVersion>0</reportVersion>
  <sendMail>N</sendMail>
  <sessionId>1234</sessionId>
  <skipApproval>Y</skipApproval>
  <skipEditing>Y</skipEditing>
  <jsonData>
    {
      "Agreements": {
        "RentAgreement": {
          "Name": "user1",
          "Rent": "2000",
          "MailTo": "user1@ccm.com"
        }
      }
    }
  </jsonData>
  <synchronousArchival>N</synchronousArchival>
  <authToken>xxxx</authToken>
  <authTokenType>JWT</authTokenType>
  <brandId>1</brandId>
  <sendTwitter>N</sendTwitter>
  <sendFacebook>N</sendFacebook>
</DocumentGenerationJSONInputBO>
```

Input parameters description

DocumentGenerationJSONInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
attachmentsLocation	No	Filepath of the External Document to be sent as an additional attachment in the Email.	NA	NA
cabinetName	Yes	Name of the cabinet where document layout is checked-in.	NA	NA
communicationId	No	Unique communication ID for the specific transaction.	NA	0
dateFrom	No	The date from which records are picked for document generation.	NA	NA
dateTo	No	The date until which records are picked for document generation.	NA	NA
doArchive	Yes	Can has a value of either Y or N. Y - if the document is to be queued for archival. N - if not required.	'Y' or 'N'	'N'
encodedPDFRequired	No	Can has a value of either Y or N. Y - Document will be returned as a binary string but will be Base64 encoded. N - No binary form of the generated document will be provided.	'Y' or 'N'	'N'

Name	Mandatory (Yes or No)	Description	Valid values	Default value
filename	No	File Name of the document to be generated, along with the provided name a unique id will be added by CCM. Example, 'filename_requestId'.	NA	If fileName is blank then fileName = 'CurrentTimeInMilliseconds_requestId' otherwise fileName = 'fileName_RequestId'
filePath	No	File location where the document needs to be generated.	NA	NA
productName	Yes	Unique for each on-demand JSON dataset for the template.	NA	NA
reportVersion	Yes	By default, or not provided then the recently base-lined layout of the document will be used and generated. For a specific generation, a layout version needs to be provided.	NA	0
saveClientData	No	If Y: insert the jsonData provided into inputDataXML column of omsJobQueueItems table.	'Y' or 'N'	'N'

Name	Mandatory (Yes or No)	Description	Valid values	Default value
sendMail	Yes	For both Mail and SMS processes, this tag is used based upon the report type of the productName passed. Can have value either Y or N. Y - if the document is to be queued for sending mail or SMS. N - if not required.	'Y' or 'N'	'N'
sendWhatsapp	Yes	Can have value either Y or N. Y - if the document is to be queued for sending WhatsApp messages. N - if not required.	'Y' or 'N'	'N'
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA
skipApproval	Yes	Useful only for Interactive Editor flow. Y - if the document does not need to go through the maker-checker process. N - if the document needs to go through the maker-checker process.	'Y' or 'N'	'Y'
skipEditing	Yes	Useful only for Interactive Editor flow. Y - if the document will not be allowed for further editing after generation. N - if the document will be allowed to go through editing after generation.	'Y' or 'N'	'Y'

Name	Mandatory (Yes or No)	Description	Valid values	Default value
jsonData	Yes	Pass User data in JSON format for which the document is to be generated.	NA	NA
synchronousArchival	No	This will be used for ondemand synchronous archival only for generation cases. For all other cases, archival will be asynchronous only.	'Y' or 'N'	'N'
authToken	No	Authenticating the user before executing the API calls requires an authorization token. The user must input the authToken which then gets validated as per client implementation.	xxxx	-
authTokenType	No	To be implemented by implementation team. authToken is required to implement this parameter.	JWT, OAUTH	-
brandId	No	Brand ID for Theme Branding	Provide a valid Brand ID	-
sendTwitter	Yes	Can have value either Y or N. - Y - if the document is to be queued for sending Twitter Communciation. - N - if not required.	'Y' or 'N'	'N'

Name	Mandatory (Yes or No)	Description	Valid values	Default value
sendFacebook	Yes	Can have value either Y or N. • Y - if the document is to be queued for sending Facebook Communciation. • N - if not required.	'Y' or 'N'	'N'

Sample output XML

DocumentGenerationJSONOutputBO

```

<DocumentGenerationJSONOutputBO>
  <communicationId/>
  <description/>
  <encodedDocxData/>
  <encodedFileData/>
  <encodedMailBodyData/>
  <filePathName>D:\Files\sampltest_116956.pdf</filePathName>
  <latestBaselinedVersion/>
  <mailBodyPathName/>
  <requestId>116956</requestId>
  <status>0</status>
  <transactionId>1567</transactionId>
  <archivalDescription>Success</archivalDescription>
  <archivalDetails>
    <documentName>fdgjksdhjk</documentName>
    <documentIndex>3</documentIndex>
    <volumeIndex>4</volumeIndex>
    <imageIndex>10</imageIndex>
  </archivalDetails>
</DocumentGenerationJSONOutputBO>

```

Output parameters description

DocumentGenerationJSONOutputBO

Name	Description
communicationId	Same communication ID that has passed in the input XML.

Name	Description
description	Will provide the description corresponding to status, whether the request was processed successfully or not.
encodedDocxData	Base64 encoded binary-form of generated docx document(non-html).
encodedFileData	Base64 encoded binary-form of generated PDF document(non-html).
encodedMailBodyData	Base64 encoded binary-form of the generated document (HTML only).
filePathName	The absolute path of the generated document(non-html).
latestBaselinedVersion	Will return Y/N. Y - The document is generated from the latest base-lined layout. N - The latest base-lined layout is not used for document generation.
mailBodyPathName	The absolute path of the '.html' file, which has the message that will be communicated to the end-user.
requestId	For every generation request, a unique ID is generated by CCM.
status	Code to ensure whether the generation was successful or not. 0 - successfully generated. Non-Zero - Some error occurred while processing the request.
transactionId	Will return a numeric ID of the generated document for contract and letter only.
archivalDescription	Failed for failure cases, Success for successful archival cases
archivalDetails	This tag will return the following values: • documentName — The name of the document. • volumeIndex — The volume index of the attachment. • documentIndex — A unique index of the document that the user wants to attach. • imageIndex — The image index of the attachment.

Error code description

Code	Message	Comment
1027	Invalid product name.	The product name is invalid.
1029	Document generation failed. Document type for the given product name is not supported.	Document type for the given product name is not supported.
1028	Either OnDemand Job is not specified for the ProductName, or details for the concerned operation(mail/archival) are incomplete.	Either OnDemand Job is not specified for the ProductName, or details for the concerned operation(mail/archival) are incomplete.
1012	The server is busy and unable to provide a Statement object.	The server is busy and unable to provide a Statement object.
11	Invalid Session.	This message appears when an invalid session is passed in the input parameters.
1034	Synchronous archival will work only for generation cases.	Synchronous archival works only for generation cases.
1035	Synchronous archival will not work for other dms cases.	Synchronous archival does not work for other DMS cases.
-1011	CabinetName not found	The cabinet name is invalid.
1037	Invalid REST token	The entered REST token is invalid.
1038	Domain user not found	The domain user is not found.
1039	REST TokenParser not implemented	The REST TokenParser parameter is enabled in the <i>OMSAdminConfig.ini</i> file. However, it is not implemented yet.

Code	Message	Comment
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <i><JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs</i> .

OMS job auto approval

OMSJobAutoApproval class provides REST web service for spooling the cycle instance and approving it without logging into OMS Admin Workspace.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSJobAutoApproval/approveJob*
- **Services Method:**
 - **approveJob** — This service method to approve a job instance.
 - **Parameters** — *OMSAcceptJobInputBO* - Object providing details for job instance to approved. Request Object to receive input for approving job instance by service method 'approveJob'.
 - **Returns** — *OMSAcceptJobOutputBO* - Object providing details of the approval request. Response Object providing details, how the request is processed using service method 'approveJob'.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

OMSAcceptJobInputBO

```
<OMSAcceptJobInputBO>
  <cabinetName>oms</cabinetName>
  <communicationId>0</communicationId>
  <cycleName>c1</cycleName>
  <filterCondition></filterCondition>
  <instanceId>2021-02-01 00:00:00.000</instanceId>
  <jobName>jobNew</jobName>
  <jobType>M</jobType>
  <sessionId>5345</sessionId>
```

```
<authToken>xxxx</authToken>
<authTokenType>JWT</authTokenType>
</OMSApproveJobInputBO>
```

Input parameters description

OMSApproveJobInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet, in which job exists.	NA	NA
communicationId	Yes	Unique communication Id for the specific transaction.	NA	NA
cycleName	Yes	Name of the cycle, in which the mentioned job is present.	NA	NA
filterCondition	No	If applied then based upon this filter condition, records will be processed.	NA	NA
instanceId	Yes	Instance id specified the date on which job instance is created. The format of id is 'YYYY-MM-DD 00:00:00'.	NA	NA

Name	Mandatory (Yes or No)	Description	Valid values	Default value
jobName	Yes	As the name suggests, the name of the job to be approved.	NA	NA
jobType	Yes	As the name suggests, type of the job. P - for print job. M - for mail job. T - for text job. W – for WhatsApp job.	NA	NA
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA
authToken	No	Authenticating the user before executing the API calls requires an authorization token. The user must input the authToken which then gets validated as per client implementation.	xxxx	-
authTokenType	No	To be implemented by implementation team. authToken is required to implement this parameter.	JWT, OAUTH	-

Sample output XML

OMSApproveJobOutputBO

```
<OMSApproveJobOutputBO>
  <jobSpoolId>100</jobSpoolId>
  <communicationId>10</communicationId>
  <status>0</status>
  <description>Success</description>
</OMSApproveJobOutputBO>
```

Output parameters description

OMSApproveJobOutputBO

Name	Description
communicationId	Same communication ID that has passed in the input XML.
description	Will provide the description corresponding to status, whether the request was processed successfully or not.
jobSpoolId	Every spooled job instance is given an ID in OmniOMS. This ID is related to instance Id.
status	Code to ensure whether the request is processed successfully or not. 0 - successfully generated. Non-Zero - Some error occurred while processing the request.

Error code description

Code	Message	Comment
400/804 (Sub error code)	The cycle name provided does not exist.	The cycle name provided does not exist.
400	InstanceId provided corresponding to the cycle name does not exist.	InstanceId provided corresponding to the cycle name does not exist.

Code	Message	Comment
400	JobType is not provided.	JobType is not provided.
1099	Duplicate Communication Id.	The communication ID is duplicate.
1037	Invalid REST token	The entered REST token is invalid.
1038	Domain user not found	The domain user is not found.
1039	REST TokenParser not implemented	The REST TokenParser parameter is enabled in the <i>OMSAdminConfig.ini</i> file. However, it is not implemented yet.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

OMS message tracker

OMSMessageTracker class provides REST web service to update the status of messages in DB if tracking is enabled in the database. Status against each message is updated in DB as Delivered or Not Delivered based upon the status passed in Input.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSMessageTracker/updateMessageTrackerStatus*
- **Services Method:**
 - **updateMessageTrackerStatus**
 - **Parameters** — *OMSMessageTrackerInputBO* - Request Object to receive input for document generation from service method 'updateMessageTrackerStatus'.
 - **Returns** — *OMSMessageTrackerOutputBO* – Response Object to update status of request.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

OMSMessageTrackerInputBO

```
<OMSMessageTrackerInputBO>
  <cabinetName>omspost1feb</cabinetName>
  <messageDetails>
    <messageDetail>
      <trackid>67567576</trackid>
      <status>C</status>
      <remarks></remarks>
    </messageDetail>
  </messageDetails>
  <sessionId>-40212061</sessionId>
  <authToken>xxxx</authToken>
  <authTokenType>JWT</authTokenType>
</OMSMessageTrackerInputBO>
```

Input parameters description

OMSMessageTrackerInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet where the message will be generated.	NA	NA
messageDetail	Yes	Message details as tracking Id and status.	NA	NA
Trackid	Yes	Tracking Id for communication.	NA	NA
remarks	No	In case any specific remarks to be updated in DB against each message.	NA	NA
Status	Yes	Based upon this status value, the message is marked Delivered or not in DB. C – Complete E – Error	C, E	E
SessionId	Yes	Session Id of the user logged in the cabinet.		NA

Name	Mandatory (Yes or No)	Description	Valid values	Default value
authToken	No	Authenticating the user before executing the API calls requires an authorization token. The user must input the authToken which then gets validated as per client implementation.	xxxx	-
authTokenType	No	To be implemented by implementation team. authToken is required to implement this parameter.	JWT, OAUTH	-

Sample output XML

OMSMessageTrackerOutputBO

```
<OMSMessageTrackerOutputBO>
  <description>Status updated successfully.</description>
  <mainCode>0</mainCode>
  <status>0</status>
  <updatedRecords>67567576</updatedRecords>
</OMSMessageTrackerOutputBO>
```

Output parameters description

OMSMessageTrackerOutputBO

Name	Description
description	Successfully updated
mainCode	'0'
status	'0'
updatedRecords	Status updated successfully.

Error code description

Code	Message	Comment
1010	Either the message Id is not provided or does not match with the database.	Either the message ID is not provided or does not match with the database.
1037	Invalid REST token	The entered REST token is invalid.
1038	Domain user not found	The domain user is not found.
1039	REST TokenParser not implemented	The REST TokenParser parameter is enabled in the <i>OMSAdminConfig.ini</i> file. However, it is not implemented yet.
-1011	CabinetName not found	The cabinet name is invalid.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <i><JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs</i> .

OMS on demand XML validation

OMSONDemandXMLValidation class provides REST web service to validate XML against an associated XSD schema file.

- **ServiceURL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSONDemandXMLValidation/xmlValidation*
- **Services Method:**
 - **xmlValidation**
 - **Parameters** — *OMSONDemandXMLValidationInputBO* – Request object to take XML as input for validation against a specific template.
 - **Returns** — *OMSONDemandXMLValidationOutputBO* – Response object to return whether input XML is valid or not.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

OMSONDemandXMLValidationInputBO

```
<OMSONDemandXMLValidationInputBO>
  <cabinetName>oms</cabinetName>
  <sessionId>5434</sessionId>
  <productName>new</productName>
  <xmlData></xmlData>
</OMSONDemandXMLValidationInputBO>
```

Input parameters description

OMSONDemandXMLValidationInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet where document layout is checked-in.	NA	NA
productName	Yes	Unique for each on-demand XML dataset for the template.	NA	NA
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA
xmlData	Yes	User data for which the document is to be generated.	NA	NA

Sample output XML

OMSONDemandXMLValidationOutputBO

```
<OMSONDemandXMLValidationOutputBO>
  <status>0</status>
  <description>Success</description>
  <validationStatus>true</validationStatus>
</OMSONDemandXMLValidationOutputBO>
```

Output parameters description

OMSONDemandXMLValidationOutputBO

Name	Description
description	Success
status	0
validationStatus	True or Error

Error code description

Code	Message	Comment
1011	Invalid XML corresponding to the Product Name provided.	The XML is invalid corresponding to the provided product name.
-1011	CabinetName not found	The cabinet name is invalid.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <i><JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs</i> .

OMS communication tracking

OMSCommunicationTracking class provides REST web service to get the updated status of communications sent from OmniOMS using communicationId when tracking within the system is enabled.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSCommunicationTracking/communicationTracking*
- **Services Method:**
 - **OMSCommunicationTracking** — This service method to track using communicationId.

- **Parameters** — OMSCommunicationTrackingInputBO - Object providing details for communication tracking. Request Object to receive input for approving job instance by service method 'communicationTracking'.
 - **Returns** — OMSCommunicationTrackingOutputBO - Object providing details of the communication tracking request.
- **Method Type:** POST
 - **Consumes:** application/xml
 - **Produces:** application/xml

Sample input XML

OMSCommunicationTrackingInputBO

```
<OMSCommunicationTrackingInputBO>
  <cabinetName>oms</cabinetName>
  <sessionId>56345</sessionId>
  <communicationId>10</communicationId>
  <requestId>433</requestId>
  <authToken>xxxx</authToken>
  <authTokenType>JWT</authTokenType>
  <masterKey>10</masterKey>
  <startFrom>2</startFrom>
  <noOfRecordsToFetch></noOfRecordsToFetch>
</OMSCommunicationTrackingInputBO>
```

Input parameters description

OMSCommunicationTrackingInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet, in which job exists.	NA	NA
communicationId	Yes	Unique communication Id for the specific transaction.	NA	NA

Name	Mandatory (Yes or No)	Description	Valid values	Default value
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA
RequestId	NA	A unique Id generated by system for every request.		0
authToken	No	Authenticating the user before executing the API calls requires an authorization token. The user must input the authToken which then gets validated as per client implementation.	xxxx	-
authTokenType	No	To be implemented by implementation team. authToken is required to implement this parameter.	JWT, OAUTH	-
masterKey	No	ID or Primary Key of the user for which document is to be generated, this value will be used to distinguish users.	Provide a valid Master Key.	Provide a valid Master Key.
startFrom	No	This is used as OFFSET i.e. if it is 10 then skip first 10 rows	NA	NA
noOfRecordsToFetch	No	Total Numebr of records to be fetched.	NA	NA

Sample output XML

OMSCommunicationTrackingOutputBO

```

<OMSCommunicationTrackingOutputBO>
  <status>-1376296581</status>
  <description>success</description>
  <TrackingXML>
    <OMSCommunicationTrackingList>
      <OMSCommunicationTracking>
        <requestId>4234</requestId>
        <masterKey>10</masterKey>
        <productName>3sde</productName>
      </OMSCommunicationTracking>
    </OMSCommunicationTrackingList>
  </TrackingXML>
</OMSCommunicationTrackingOutputBO>

```

```
<sentDate>2021-02-01 19:39:11.620 </sentDate>
<readDate> 2021-02-02 19:39:11.620 </readDate>
<status>0</status>
<mailId>abc@newgen.co.in</mailId>
<messageId>45</messageId>
<deliveredDate></deliveredDate>
<recipientNumber>5352341241</recipientNumber>
</OMSCommunicationTracking>>
<NoOfRecordsFetched>10</NoOfRecordsFetched>
<NextBatchExist>Y</NextBatchExist>
</OMSCommunicationTrackingList>
</TrackingXML>
</OMSCommunicationTrackingOutputBO>
```

Output parameters description

OMSCommunicationTrackingOutputBO

Name	Description
description	Success
status	'0' for success.
trackingXML	Will return basic details of the transaction like: Request Id, MasterKey, ProductName, SentDate, and more.

Error code description

Code	Message	Comment
1033	Record not present in the database.	The record is not present in the database.
1099	Provide a valid communicationId.	The provided communicationId is invalid.
11	Invalid Session	This message appears when an invalid session is passed in the input parameters.
-1011	CabinetName not found	The cabinet name is invalid.
1036	Provide a valid communicationId or RequestId	The communicationId or RequestId is invalid.
1037	Invalid REST token	The entered REST token is invalid.
1038	Domain user not found	The domain user is not found.
1039	REST TokenParser not implemented	The REST TokenParser parameter is enabled in the <i>OMSAdminConfig.ini</i> file. However, it is not implemented yet.

Code	Message	Comment
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

OMS on demand data purge

OMSONDemandDataPurge class provides REST web service to purge data from OmniOMS tables. The data of completed jobs move to the history table.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSONDemandDataPurge/dataPurge*
- **Services Method:**
 - **OMSONDemandDataPurge** — This service method is used to process data purge requests, of moving data of completed jobs to the history table.
 - **Parameters** — *OMSDDataPurgeInputBO* - Object providing details for a request to move completed job data. Request Object to receive input for purging data from service method 'dataPurge'.
 - **Returns** — *OMSDDataPurgeOutputBO* - Object providing details of whether a request was processed successfully or not.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

```
<OMSDDataPurgeInputBO>
  <cabinetName>oms</cabinetName>
  <dateFrom>2020-12-11 00:00:00.000</dateFrom>
  <dateTo>2020-12-21 00:00:00.000</dateTo>
</reportType></reportType>
<requiredFlag>N</requiredFlag>
  <sessionId>53442</sessionId>
<authToken>xxxx</authToken>
<authTokenType>JWT</authTokenType>
</OMSDDataPurgeInputBO>
```

Input parameters description

OMSDDataPurgeInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet, of which data need to be purged.	NA	NA
dateFrom	No	Completed jobs that are spooled on and after this, a date will be picked for purging. Format of Id is 'YYYY-MM-DD 00:00:00' If not provided, purging will begin from the very first completed job.	NA	NA
dateTo	Yes	Completed jobs that are spooled before this date will be purged. Format of Id is 'YYYY-MM-DD 00:00:00'	NA	NA
reportType	No	Type of the report that will be used for the purging. No need to pass any value except for Letter or Contract Report types.	L, C	Empty ("")
requiredFlag	Yes	Based upon this column, will insert records into the OMSJobQueueItemsHistory table.	'Y' or 'N'	'N'
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA
authToken	No	Authenticating the user before executing the API calls requires an authorization token. The user must input the authToken which then gets validated as per client implementation.	xxxx	-
authTokenType	No	To be implemented by implementation team. authToken is required to implement this parameter.	JWT, OAUTH	-

Sample output XML

```
<OMSDataPurgeOutputBO>
  <status>0</status>
  <description>Success</description>
</OMSDataPurgeOutputBO>
```

Output parameters description

OMSDataPurgeOutputBO

Name	Description
description	OnDemandGeneration data purged successfully.
status	'0' for success.

Error code description

Code	Message	Comment
1	Unable to connect to the server, check logs.	Check the logs to rectify the server connection failure.
11	Invalid Session.	This message appears when an invalid session is passed in the input parameters.
1037	Invalid REST token	The entered REST token is invalid.
1038	Domain user not found	The domain user is not found.
1039	REST TokenParser not implemented	The REST TokenParser parameter is enabled in the <i>OMSAdminConfig.ini</i> file. However, it is not implemented yet.
-1011	CabinetName not found	The cabinet name is invalid.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <i><JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs</i> .

OMS retrieve details (for getting OmniDocs details)

Useful to fetch OmniDocs details based upon the communicationId and requestId.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSRetrieveDetails/retrieveDetails*
- **Services Method:** OMSRetrieveDetails
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

```
<OMSRetrieveDetailsInputBO>
  <cabinetName>omsrishav_25nov</cabinetName>
  <sessionId>-1866771058</sessionId>
  <requestId>323</requestId>
  <communicationId>10</communicationId>
</OMSRetrieveDetailsInputBO>
```

Input parameters description

OMSRetrieveDetails_Input

Name	Mandatory (Yes or No)	Description	Valid value	Default value
cabinetName	Yes	Name of the cabinet where document layout is checked-in.	NA	NA
sessionId	Yes	Session ID for the user currently logged into the cabinet.	NA	NA
requestId	No	For every request generation, CCM generates a unique ID.	NA	NA
communicationId	No	A unique communication ID for the specific transaction.	NA	0

Sample output XML

```
<OMSRetrieveDetailsOutputBO>
  <description>Success</description>
  <documentIndex>1</documentIndex>
  <documentName>FileName</documentName>
  <imageIndex>10</imageIndex>
  <isArchived>Y</isArchived>
  <status>0</status>
  <volumeIndex>1</volumeIndex>
</OMSRetrieveDetailsOutputBO>
```

Output parameters description

OMSRetrieveDetails_Output

Name	Description
documentIndex	Archived document index in OmniDocs.
documentName	Archived document name in OmniDocs.
imageIndex	Archived document image index in OmniDocs.
volumeIndex	Archival volume index in OmniDocs.
isArchived	Document archived or not in OmniDocs. It returns Y for successful archival and N for failed archival.
description	Data retrieved successfully.
status	'0' for success.

Error code description

Code	Message	Comment
11	Invalid Session.	This message appears when an invalid session is passed in the input parameters.
1036	Provide a valid communicationId or RequestId	The communicationId or RequestId is invalid.

Code	Message	Comment
-1011	CabinetName not found	The cabinet name is invalid.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

OMS retrieve details (for decoding user data)

The *OMSRetrieveDetails* API allows you to fetch the decoded user data based on *requestid*.

- **Service URL:** *http://IP:Port/OMSWebServices/rest/OMSRetrieveDetails/getUserData*
- **Services Method:** OMSRetrieveDetails
- **Method Type:** POST
- **Consumes:** application/json
- **Produces:** application/json

Sample input JSON

```
{
  "CabinetName": "omsc187",
  "RequestId": 2014,
  "SessionId": -12697
}
```

Input parameters description

OMSGetDecodedData_Input

Name	Mandatory (Yes or No)	Description	Valid value	Default value
cabinetName	Yes	Name of the cabinet where document layout is checked-in.	NA	NA
sessionId	Yes	Session ID for the user currently logged into the cabinet.	NA	NA
requestId	Yes	For every request generation, CCM generates a unique ID.	NA	NA

Sample output JSON

```
{
  "Status": "0",
  "Description": "Content decoded Successfully",
  "DecodedData":
"<Root>\n<Agreements>\n<RentAgreement>\n<ContractStartdate>2016-03- 01</ContractStartdate>\n<Durationofcontract>11545323</Durationofcontract>\n<Name>It (injected humour and the like) .</Name>\n<Relationtype>Son</Relationtype>\n<Regiontype>Metro</Regiontype>\n<Capacityofnewprperty>2018-09-01</Capacityofnewprperty>\n<Maritalstatus>1</Maritalstatus>\n<Securityamount>373839</Securityamount>\n</RentAgreement>\n</Agreements>\n</Root>"
}
```

Output parameters description

OMSGetDecodedData_Output

Name	Description
DecodedData	Decoded user data.

Name	Description
Description	Content decoded successfully.
Status	'0' for success.

Error code description

Code	Message	Comment
11	Invalid Session.	This message appears when an invalid session is passed in the input parameters.
1021	NO DATA FOUND	The required data is unavailable in the database.
15	Pass valid inputs, getting database exception!!	The provided inputs do not match the existing data in the database.
400	Pass valid inputs!!	The provided inputs are invalid.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <i><JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs</i> .

OMS secured DB credentials

The *OMSSecuredDbCredentials* API allows you to encrypt the data passed in the *InputData* parameter of the request body.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSSecuredDataEncryption/OMSSecuredDbCredentials*
- **Services Method:** *OMSSecuredDataEncryption*
- **Method Type:** *POST*
- **Consumes:** *application/json*
- **Produces:** *application/json*

Sample input JSON

```
{
  "CabinetName": "name_of_cabinet",
  "InputData": "data_for_encryption"
}
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
CabinetName	Yes	Name of the cabinet whose database credentials you want to encrypt.	NA	NA
InputData	Yes	Value passed in the <i>InputData</i> parameter that you want to encrypt.	NA	NA

Sample output JSON

```
{
  "Status": "0",
  "Description": "Success",
  "EncryptedData": "encrypted_data"
}
```

Output parameters description

Name	Description
Description	Indicates the successful operation when the <i>Status</i> parameter value is "0".
Status	The "0" value of this parameter indicates a successful data encryption. Any other value indicates the failure of data encryption.
EncryptedData	The data is encrypted according to compliance.

Error code description

Code	Message	Comment
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <i><JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs</i> .

OMS secured data encryption

Use to fetch encrypted input data.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSSecuredDataEncryption/securedDataEncryption*
- **Services Method:** *OMSSecuredDataEncryption*
- **Method Type:** POST
- **Consumes:** *application/xml*
- **Produces:** *application/xml*

Sample input XML

```
<OMSSecuredDataEncryptionInputBO>
  <cabinetName>omspost06may</cabinetName>
  <sessionId>1612732954</sessionId>
  <inputDataList>
    <inputData>134256</inputData>
    <inputData>1234</inputData>
  </inputDataList>
</OMSSecuredDataEncryptionInputBO>
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet containing the database table whose data you want to encrypt.	NA	NA
inputData	Yes	The table data that you want to encrypt.	NA	NA
sessionID	Yes	Session ID of the user logged in the cabinet.	NA	NA

Sample output XML

```
<OMSSecuredDataEncryptionOutputBO>
  <description>Success</description>
  <outputDataList>
    <outputData>
      <encryptedData>KtiojE6Q3s6iiIoPXcDgUQ==</encryptedData>
      <inputData>134256</inputData>
    </outputData>
    <outputData>
      <encryptedData>+T33cQSyOaaTQzRCE8mfkw==</encryptedData>
      <inputData>1234</inputData>
    </outputData>
  </outputDataList>
  <status>0</status>
</OMSSecuredDataEncryptionOutputBO>
```

Output parameters description

Name	Description
description	Indicates the successful operation when the <i>status</i> parameter value is "0".
status	The "0" value of this parameter indicates a successful operation. Any other value indicates the failure of the operation.
inputData	The table data that you want to encrypt.
encryptedData	The data is encrypted according to compliance.

Error code description

Code	Message	Comment
11	Invalid Session.	This message appears when an invalid session is passed in the input parameters.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <i><JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs</i> .

OMS external content library

This chapter provides information on the following methods of the *OMSExternalContentLibrary* class:

- [AddExternalContent](#)
- [ModifyExternalContent](#)
- [DeleteExternalContent](#)
- [FetchExternalContent](#)

Adding external content

The *AddExternalContent* method provides REST web service to add external content (RTF or HTML) to the database.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSExternalContentLibrary/addExternalContent*
- **Services Method:** *addExternalContent*
Use this method to take the input parameter values from the user and add it to the database.
 - **Parameters:**
 - **fileStream** – Accepts attached RTF or HTML file as input (as *InputStream*).
 - **contentProperties** – Accepts required parameters as a string in JSON format.
 - **Returns:** JSON output containing status (code) and description of the result.
- **Method Type:** POST
- **Consumes:** *MediaType.MULTIPART_FORM_DATA*
- **Produces:** *MediaType.APPLICATION_JSON*

Sample input

- **fileStream:** Upload input file as an attachment.
- **contentProperties:**

```
{
  "CategoryName": "Loan",
  "CommunicationGroupName": "HomeLoan",
  "LabelName": "HomeLoan",
  "ContentType": "RTF",
  "CabinetName": "oms10sp1",
  "SessionId": "-124939540"
  "ForBranding": "N"
}
```

Input parameters description

Key	Name	Mandatory (Yes or No)	Description	Valid values	Default value
fileStream	Attached File	Yes	The file that must be added in RTF or HTML format.	NA	NA
content Properties	CategoryName	No	Name of the category where the RTF or HTML file must be added. In the case of enterprise components, pass an empty string value ("") for <i>CategoryName</i>.	NA	NA
	Communication GroupName	No	Name of the communication group where RTF or HTML file must be added. In the case of enterprise components, pass an empty string value ("") for <i>Communication GroupName</i>.	NA	NA
	LabelName	Yes	Unique label name associated with <i>contentFile</i> .	NA	NA
	ContentType	Yes	Content type of the attached file. For example, RTF or HTML.	RTF or HTML	NA
	CabinetName	Yes	Name of the cabinet where the user has logged in. This cabinet contains the report.	NA	NA
	SessionId	Yes	Session ID for the current cabinet login session for the user.	NA	NA

Key	Name	Mandatory (Yes or No)	Description	Valid values	Default value
	ForBranding	No	To be set as Y if the RTF/HTML content is intended for ThemeBranding. Once an RTF is created for ThemeBranding, it cannot be used for external content.	'Y' or 'N'	'N'

Sample output

```
{
  "Status": "0",
  "Description": "Added successfully"
}
```

Output parameters description

Name	Description
Status	Code represents the successful or unsuccessful addition of the <i>contentFile</i>: • 0 — Successfully added • Non-zero — Some error occurred while processing the request.
Description	Description corresponding to the status, that is success or failure of a request.

Error code description

Code	Message	Comment
11	Invalid Session	This message appears when an invalid session is passed in the input parameters.

Code	Message	Comment
1010	Invalid Request	This message appears when mandatory parameters are either null or empty.
1041	LabelName already exists	This message appears when LabelName is already added in the database.
1042	LabelName does not exists	This message appears while performing modification, deletion or fetching, when the LabelName is not present in the database.
1043	CommunicationGroupName does not exists	This message appears when CommunicationGroupName does not exist in the given category.
1044	Invalid Macros	This message appears when the input contentFile contains macros, but those macros are not valid.
1045	ContentType in database and input request does not match	This message appears when the user tries modifying content of other file type
1046	LabelName is mapped with report or component	This message appears when the user tries deleting a label that is mapped with a report or component.
1047	Input file and content type are not same	This message appears when the type of input file and contentType passed in the inputJSON is not same.
1048	Category name is null or empty	This message appears when the user tries making an entry for a non-enterprise component.
1049	Category does not exist	This message appears when the categoryName does not exist.
1050	Communication group is null or empty	This message appears when the user tries making an entry for a non-enterprise component.
1051	Some error occurred	This message appears when some unknow error occurs.
1052	Input file is not present	This message appears when the input file is not passed.
1053	File is neither RTF nor HTML type	This message appears when the input file is not in RTF or HTML format.
1055	JavaScript is present in input file	This message appears when the input file contains the JavaScript.

Code	Message	Comment
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

Modifying external content

The *ModifyExternalContent* method provides REST web service to modify external content (RTF or HTML) present in the database.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSExternalContentLibrary/modifyExternalContent*
- **Services Method:** *modifyExternalContent*
Use this method to take the input parameter values from the user and modify the entries for them in the database.
 - **Parameters:**
 - **fileStream** — Accepts attached RTF or HTML file as input (as *InputStream*).
 - **contentProperties** — Accepts required parameters as a string in JSON format.
 - **Returns** — JSON output containing status (code) and description of the result.
- **Method Type:** POST
- **Consumes:** *MediaType.MULTIPART_FORM_DATA*
- **Produces:** *MediaType.APPLICATION_JSON*

Sample input

- **fileStream:** Upload the input file as an attachment.
- **contentProperties:**

```
{
  "CategoryName": "Loan",
  "CommunicationGroupName": "HomeLoan",
  "LabelName": "HomeLoan",
  "CabinetName": "oms10sp1",
  "SessionId": "2078683191"
}
```

Input parameters description

Key	Name	Mandatory (Yes or No)	Description	Valid values	Default value
fileStream	Attached File	Yes	The file that must be added in RTF or HTML format.	NA	NA
content Properties	CategoryName	No	Name of the category where the RTF or HTML file must be added. In the case of enterprise components, pass an empty string value ("") for <i>CategoryName</i>.	NA	NA
	Communication GroupName	No	Name of the communication group where RTF or HTML file must be added. In the case of enterprise components, pass empty string value ("") for <i>CommunicationGroupName</i>.	NA	NA
	LabelName	Yes	Label name associated with <i>contentFile</i> that must be modified.	NA	NA

Key	Name	Mandatory (Yes or No)	Description	Valid values	Default value
	CabinetName	Yes	Name of the cabinet where the user has logged in. This cabinet contains the report.	RTF or HTML	NA
	SessionId	Yes	Session ID for the current cabinet login session for the user.	NA	NA

Sample output

```
{
  "Status": "0",
  "Description": "Modified successfully"
}
```

Output parameters description

Name	Description
Status	Code represents successful or unsuccessful modification of the contentFile: • 0 — Successfully modified • Non-zero — Some error occurred while processing the request.
Description	Description corresponding to the status, that is success or failure of a request.

Error code description

Code	Message	Comment
11	Invalid Session	This message appears when an invalid session is passed in the input parameters.
1010	Invalid Request	This message appears when mandatory parameters are either null or empty.
1042	LabelName does not exists	This message appears while performing modification, deletion or fetching, when the LabelName is not present in the database.
1043	CommunicationGroupName does not exists	This message appears when communicationgroupName does not exist in the given category.
1044	Invalid Macros	This message appears when the input contentFile contains macros, but those macros are not valid.
1045	ContentType in database and input request does not match	This message appears when the user tries modifying content of other file type
1047	Input file and content type are not same	This message appears when the type of input file and contentType passed in the inputJSON is not same.
1048	Category name is null or empty	This message appears when the user tries making an entry for a non-enterprise component.
1049	Category does not exist	This message appears when the categoryName does not exist.
1050	Communication group is null or empty	This message appears when the user tries making an entry for a non-enterprise component.
1051	Some error occurred	This message appears when some unknow error occurs.
1052	Input file is not present	This message appears when the input file is not passed.
1053	File is neither RTF nor HTML type	This message appears when the input file is not in RTF or HTML format.

Code	Message	Comment
1055	JavaScript is present in input file	This message appears when the input file contains the JavaScript.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <i><JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs</i> .

Deleting external content

The *DeleteExternalContent* method provides REST Web Service to delete external content (RTF or HTML) from the database.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSExternalContentLibrary/deleteExternalContent*
- **Services Method:** *deleteExternalContent*
Use this method to take the input parameter values from the user and delete them from the database.
 - **Parameters** — Accepts required parameters in JSON format.
 - **Returns** — JSON output containing status (code) and description of the result.
- **Method Type:** POST
- **Consumes:** MediaType.APPLICATION_JSON
- **Produces:** MediaType.APPLICATION_JSON

Sample input

```
{
  "LabelName": "HomeLoan",
  "CabinetName": "oms10sp1",
  "SessionId": "2002860323"
}
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
LabelName	Yes	Label name associated with contentFile that must be deleted.	NA	NA
CabinetName	Yes	Name of the cabinet where the user has logged in. This cabinet contains the report.	NA	NA
SessionId	Yes	Session ID for the current cabinet login session for the user.	NA	NA

Sample output

```
{
  "Status": "0",
  "Description": "Deleted successfully"
}
```

Output parameters description

Name	Description
Status	Code represents the successful or unsuccessful deletion of the <i>contentFile</i> : • 0 — Successfully deleted • Non-zero —Some error occurred while processing the request.
Description	Description corresponding to the status, that is success or failure of a request.

Error code description

Code	Message	Comment
11	Invalid Session	This message appears when an invalid session is passed in the input parameters.
1010	Invalid Request	This message appears when mandatory parameters are either null or empty.
1042	Label Name does not exists	This message appears while performing modification, deletion or fetching, when the LabelName is not present in the database.
1046	Label Name is mapped with report or component	This message appears when the user tries deleting a label that is mapped with a report or component.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <i><JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs</i> .

Fetching external content

The *FetchExternalContent* method provides REST web service to fetch external content (RTF or HTML) from the database.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSExternalContentLibrary/fetchExternalContent*
- **Services Method:** *fetchExternalContent*
Use this method to take input parameter values from the user and retrieve the required entries from the database.
 - **Parameters** — Accepts required parameters in JSON format.
 - **Returns** — JSON output containing status (code), description, label name, content type, and FileContent (Base64 encoded).
- **Method Type:** POST
- **Consumes:** MediaType.APPLICATION_JSON
- **Produces:** MediaType.APPLICATION_JSON

Sample input

```
{
  "LabelName": "HomeLoan",
  "CabinetName": "oms10sp1",
  "SessionId": "1372766394"
}
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
LabelName	Yes	Label name associated with <i>contentFile</i> that must be fetched.	NA	NA
CabinetName	Yes	Name of the cabinet where the user has logged in. This cabinet contains the report.	NA	NA
SessionId	Yes	Session ID for the current cabinet login session for the user.	NA	NA

Sample output

```
{
  "FileContent": "",
  "Status": "11",
  "Description": "Invalid Session",
  "ContentType": "",
  "LabelName": ""
}
```

Output parameters description

Name	Description
Status	Code represents successful or unsuccessful fetching of the <i>contentFile</i> : • 0 — Successfully fetched • Non-zero — Some error occurred while processing the request.
Description	Description corresponding to the status, that is success or failure of a request.
FileContent	Base64 encoded form of file (string).
ContentType	Content type of the fetched file.
LabelName	Label name associated with the fetched <i>contentFile</i> .

Error code description

Code	Message	Comment
11	Invalid Session	This message appears when an invalid session is passed in the input parameters.
1010	Invalid Request	This message appears when mandatory parameters are either null or empty.
1042	Label Name does not exists	This message appears while performing the modification, deletion or fetching when the <i>LabelName</i> is not present in the database.
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

APIs for CCM integration

This chapter provides information on the following APIs:

- OMS search templates
- OMS view template
- OMS get cycle job list
- OMS get data set schemas

OMS search templates

This API is useful to find out the information regarding the created template(s) on the specific cabinet. This API is used to get data using a search keyword.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSWrapperServices/getOMSXMLCall*
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

```
<OMSSearchTemplates_Input>
  <Option>OMSSearchTemplates</Option>
  <CabinetName>abd</CabinetName>
  <SessionId>1234</SessionId>
  <SearchParameter></SearchParameter>
  <CategoryName>cat2</CategoryName>
  <MetaDataSearchParameter>
    <MetaDataKey></MetaDataKey>
    <MetaDataValue></MetaDataValue>
  </MetaDataSearchParameter>
</OMSSearchTemplates_Input>
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet in which the session is to be established.	NA	NA
sessionId	Yes	The unique number generated for each session request.		
SearchParameter	Yes	Users can apply the filter on the <i>category name</i> , <i>communicationGroupName</i> , <i>reportName</i> , <i>ProductName</i> , and <i>Description</i> of the template.	NA	NA
CategoryName	No	User can give the <i>categoryName</i> to search for a specific category.	NA	NA
MetaDataSearchParameter	No	Allows to apply the filter on the metadata such as <i>MetaDataKey</i> and <i>MetaDataValue</i> , associated with the template.	NA	NA
MetaDataKey	Yes	Allows to apply the filter on the metadata based on <i>MetaDataKey</i> associated with the template. If the <i>MetaDataSearchParameter</i> parameter is included in the request, it is mandatory to also include the <i>MetaDataKey</i> in the request.	NA	NA
MetaDataValue	No	Allows to apply the filter on the metadata based on <i>MetaDataValue</i> associated with the template.	NA	NA

Sample output XML

```
<?xml version="1.0"?>
<OMSSearchTemplates_Output>
  <Option>OMSSearchTemplates</Option>
```

```

<Exception>
  <MainCode>0</MainCode>
</Exception>
<Status>0</Status>
<DataList>
  <Data>
    <CategoryName>New Category</CategoryName>
    <CommunicationGroupName>NewomunicationGroup</CommunicationGroupName>
    <ReportName>New Message Report</ReportName>
    <ReportType>T</ReportType>
    <ProductName>13</ProductName>
    <MailReady>N</MailReady>
    <Description>VRFG</Description>
    <DBType>SQL</DBType>
    <ReportVersions>1</ReportVersions>
    <ParameterMatch></ParameterMatch>
  </Data>
  <Data>
    <CategoryName>New Category (3)</CategoryName>
    <CommunicationGroupName>New CommunicationGroup</
CommunicationGroupName>
    <ReportName>New Report</ReportName>
    <ReportType>N</ReportType>
    <ProductName>34</ProductName>
    <Description>544</Description>
    <DBType>XML</DBType>
    <ReportVersions>1</ReportVersions>
    <ParameterMatch></ParameterMatch>
  </Data>
</DataList>
</OMSSearchTemplates_Output>

```

Output parameters description

Name	Description
CategoryName	The unique category name.
CommunicationGroupName	The unique communication group name associated with the unique category.
ReportName	Unique report name under unique communicationGroup.
ReportType	Type of the report.
ProductName	Unique for each on-demand XML dataset for the template.
MailReady	MailReady will be Y if this template can be used for sending mails.

Name	Description
Description	The Product description.
DBType	Configure database.
ReportVersion	Template baselined version.
ParameterMatch	If the user provides a filter using the <i>searchParameter</i> in the inputXML then this field represents the matching name of the <i>category</i> , <i>communicationGroupName</i> , <i>ReportName</i> , <i>ProductName</i> , and <i>Description</i> as comma separated.

Error code description

Code	Message	Error type
1021	No Data Found.	Temporary
11	Invalid Session	Temporary
15	Could Not get the list!!	Fatal
400	Could Not get the list!!	Temporary
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

OMS view template

This API is used to get encoded PDF data of the specific user-defined product name.

- **Service URL:** `http://<IP>:<Port>/OMSWebServices/rest/OMSWrapperServices/getOMSXMLCall`
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

```
<OMSViewTemplate_Input>
  <Option>OMSViewTemplate</Option>
  <CabinetName>abd</CabinetName>
  <SessionId>1234</SessionId>
  <ProductName></ProductName>
  <ReportVersion></ReportVersion>
</OMSViewTemplate_Input>
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet in which the session is to be established.	NA	NA
sessionId	Yes	The unique number generated for each request.	NA	NA
ReportVersion	No	Template baselined version.	Integer value	0
ProductName	Yes	Unique for each on-demand XML dataset for the template.	NA	NA

Sample output XML

```
<OMSViewTemplate_Output>
  <Option>OMSViewTemplate</Option>
  <Exception>
    <MainCode>0</MainCode>
  </Exception>
  <Status>0</Status>
  <OMSViewTemplate>
    <Template>SomeOutput</Template>
    <Reporttype>N</Reporttype>
  </OMSViewTemplate>
</OMSViewTemplate_Output>
```

Output parameters description

Name	Description
ReportType	Type of the report
Template	Encoded PDF data

Error code description

Code	Message	Error type
11	Invalid Session	Temporary
1023	Last Baselined Preview Not Saved.	Temporary
15	Attempt Failed!!!	Fatal
400	Attempt Failed!!!	Temporary
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

OMS get cycle job list

This API returns cycle and job configuration details for the specific product name defined in the request.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSWrapperServices/getOMSXMLCall*
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

```
<OMSGetCycleJobList_Input>
  <Option>OMSGetCycleJobList</Option>
  <CabinetName>asdf</CabinetName>
  <SessionId>1234</SessionId>
  <ProductName>13</ProductName>
</OMSGetCycleJobList_Input>
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet in which the session is to be established.	NA	NA
sessionId	Yes	The unique number generated for each request.	NA	NA
productName	Yes	Unique for each on-demand XML dataset for the template.	NA	NA

Sample output XML

```
<OMSCycleJobList_Output>
  <Option>OMSCycleJobList</Option>
  <Exception>
    <MainCode>0</MainCode>
  </Exception>
  <Status>0</Status>
  <CycleJobList>
    <List>
      <jobId>1</jobId>
      <JobType>M</JobType>
      <cycleId>3</cycleId>
      <cycleName>C1</cycleName>
      <jobname>q1</jobname>
    </List>
    <List>
      <JobSpoolID>1</JobSpoolID>
      <CycleSpoolId>1</CycleSpoolId>
      <jobId>1</jobId>
```

```

        <JobType>M</JobType>
        <cycleId>2</cycleId>
        <cycleName>C2</cycleName>
        <jobname>q1</jobname>
    </List>
</CycleJobList>
</OMSCycleJobList_Output>

```

Output parameters description

Name	Description
JobId	A unique ID for the Job.
JobType	As the name suggests, type of the job. P - for print job. M - for mail job T - for text job W – for WhatsApp job
JobName	As the name suggests, the name of the job to be approved.
CycleId	A unique ID for the Cycle.
CycleName	Name of the cycle, in which the mentioned job is present.

Error code description

Code	Message	Error type
11	Invalid Session	Temporary
15	Could Not get the list!!	Fatal
1021	No Data Found for Product	Temporary
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

OMS get dataset schemas

This API is used to get the associated Dataset Schemas for multiple report versions mentioned in the API request.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSWrapperServices/getOMSXMLCall*
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

```
<OMSGetDatasetSchemas_Input>
  <Option>OMSGetDatasetSchemas</Option>
  <CabinetName>abc</CabinetName>
  <SessionId>1234</SessionId>
  <RecordList>
    <Record>
      <ProductName>Simplify Solutions</ProductName>
      <ReportVersion>2</ReportVersion>
    </Record>
  </RecordList>
</OMSGetDatasetSchemas_Input>
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet in which the session is to be established.	NA	NA
sessionId	Yes	The unique number generated for each request.	NA	NA
ReportVersion	Yes	The latest baselined version of the template.	Integer values	0

Name	Mandatory (Yes or No)	Description	Valid values	Default value
ProductName	Yes	Unique for each on-demand XML dataset for the template.	NA	NA

Sample output XML

```

<OMSGetDatasetSchemas_Output>
  <Option>OMSGetDatasetSchemas</Option>
  <Exception>
    <MainCode>0</MainCode>
  </Exception>
  <Status>0</Status>
  <DataList>
    <Data>
      <ProductName>Samplify Solutions</ProductName>
      <ReportVersion>1</ReportVersion>
      <IsLatest>Y</IsLatest>
      <DatasetSchema>
        <xs:schema attributeFormDefault="unqualified" ...>
          <!-- Content of xs:schema goes here -->
        </xs:schema>
      </DatasetSchema>
    </Data>
  </DataList>
</OMSGetDatasetSchemas_Output>

```

Output parameters description

Name	Description
ReportVersion	Latest Baselined version of the template.
ProductName	Unique for each on-demand XML dataset for the template.
IsLatest	Y
DatasetSchema	Dataset schema of the XSD dataset.

Error code description

Code	Message	Error type
11	Invalid Session	Temporary
1021	No Data Found	Temporary
15	Could Not get the list!!	Fatal
400	Could Not get the list!!	Temporary
541	Invalid input parameter, check the logs	Whitelisting issue occurs due to invalid input parameter. Refer to the <i>APIConsole.log</i> file located at <JBoss EAP home directory>\bin\Newgen\NGLogs\<cabinet name>\OmniOMS_Logs\ServerLogs.

Custom theme branding

This method of OMSThemeBranding class provides REST Web service to provide CRUD operation to our database. This chapter provides information on the following APIs:

- [addBrandingStyle](#)
- [modifyBrandingStyle](#)
- [deleteBrandingStyle](#)
- [fetchBrandingStyle](#)

addBrandingStyle

This API is used to add branding for use within OMS.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSThemeBranding/addBrandingStyle*
- **Method Type:** POST
- **Consumes:** application/json
- **Produces:** application/json

Sample input XML

```
{
  "CabinetName": "oms11splorcl",
  "SessionId": "1850576355",
  "BrandID": "brand",
  "Attribute": "sdcdd",
  "FontStyle": {
    "FontName": "Arial",
    "FontSize": "14",
    "FontColor": "rgb(12,12,12)",
    "BackColor": "rgb(12,13,13)"
  },
  "ExternalHTMLLabelName": ""
}
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet in which the session is to be established.	NA	NA
sessionId	Yes	The unique number is generated for each session request.	NA	NA
BrandID	Yes	Represents the unique identifier for the brand name.	NA	""
Attribute	Yes	Specifies the key attribute of the brand.	NA	""

Name	Mandatory (Yes or No)	Description	Valid values	Default value
FontStyle	No	Defines the font styling to be applied. This includes several optional fields: • FontName: Specifies the name of the font to be used. • FontSize: Indicates the size of the font. • FontColor: Determines the color of the font in RGB format. • BackColor: Sets the background color behind the text in RGB format.	NA	""
ExternalHTMLLabelName	No	Identifies the external content label name when branding is enabled.	NA	""

Sample output XML

```
{
  "Status": "1041",
  "Description": "Attribute  already exists"
}
```

Output parameters description

Name	Description
description	Provides the description corresponding to status, whether the request was processed successfully or not.
status	Code to ensure whether the request is processed successfully or not.

Error code description

Code	Message	Error type
11	Invalid Session	The session is invalid or has expired.

Code	Message	Error type
-1011	Cabinet Not Found	The specified cabinet could not be located.
541	Invalid input parameters	The provided input parameters are incorrect or incomplete.
1041	Attribute already exists	The attribute you are trying to add already exists.
400	Operation Failed	A message with the reason for API service failure.

modifyBrandingStyle

This API is used to take input parameters from the user and modify the database entries.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSThemeBranding/modifyBrandingStyle*
- **Method Type:** POST
- **Consumes:** application/json
- **Produces:** application/json

Sample input XML

```
{
  "CabinetName": "oms11splorcl",
  "SessionId": "1850576355",
  "BrandID": "brand",
  "Attribute": "sdcdd",
  "FontStyle": {
    "FontName": "Arial",
    "FontSize": "14",
    "FontColor": "rgb(12,12,12)",
    "BackColor": "rgb(12,13,13)"
  },
  "ExternalHTMLLabelName": ""
}
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet in which the session is to be established.	NA	NA
sessionId	Yes	The unique number is generated for each session request.	NA	NA
BrandID	Yes	Represents the unique identifier for the brand name.	NA	""
Attribute	Yes	Specifies the key attribute of the brand.	NA	""
FontStyle	No	Defines the font styling to be applied. This includes several optional fields: • FontName: Specifies the name of the font to be used. • FontSize: Indicates the size of the font. • FontColor: Determines the color of the font in RGB format. • BackColor: Sets the background color behind the text in RGB format.	NA	""
ExternalHTMLLabelName	No	Identifies the external content label name when branding is enabled.	NA	""

Sample output XML

```
{
  "Status": "1041",
  "Description": "Attribute  already exists"
}
```

Output parameters description

Name	Description
description	Provides the description corresponding to status, whether the request was processed successfully or not.
status	Code to ensure whether the request is processed successfully or not.

Error code description

Code	Message	Error type
11	Invalid Session	The session is invalid or has expired.
-1011	Cabinet Not Found	The specified cabinet could not be located.
541	Invalid input parameters	The provided input parameters are incorrect or incomplete.
1041	Attribute already exists	The attribute you are trying to add already exists.
400	Operation Failed	A message with the reason for API service failure.

deleteBrandingStyle

This API is used to take input parameters from the user and delete required entries from the database.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSThemeBranding/deleteBrandingStyle*
- **Method Type:** POST
- **Consumes:** application/json
- **Produces:** application/json

Sample input XML

```
{
  "CabinetName": "oms11splhotfix",
  "SessionId": "1066535899",
  "BrandID": "brand124",
  "Attribute": "subheading"
}
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet in which the session is to be established.	NA	NA
sessionId	Yes	A unique number is generated for each request.	NA	NA
BrandID	Yes	Represents the unique identifier for the brand name.	NA	""
Attribute	Yes	Specifies the key attribute of the brand.	NA	""

Sample output XML

```
{
  "Status": "1041",
  "Description": "Attribute  already exists"
}
```

Output parameters description

Name	Description
description	Provides the description corresponding to status, whether the request was processed successfully or not.

Name	Description
status	Code to ensure whether the request is processed successfully or not.

Error code description

Code	Message	Error type
11	Invalid Session	The session is invalid or has expired.
-1011	Cabinet Not Found	The specified cabinet could not be located.
541	Invalid input parameters	The provided input parameters are incorrect or incomplete.
1041	Attribute already exists	The attribute you are trying to add already exists.
400	Operation Failed	A message with the reason for API service failure.

fetchBrandingStyle

This API is used to take input parameters from the user and retrieve required entries from the database.

- **Service URL:** *http://<IP>:<Port>/OMSWebServices/rest/OMSThemeBranding/fetchBrandingStyle*
- **Method Type:** POST
- **Consumes:** application/json
- **Produces:** application/json

Sample input XML

```
{
  "CabinetName": "oms11splhotfix",
  "SessionId": "1066535899",
  "BrandID": "brand124",
  "Attribute": "subheading"
}
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet in which the session is to be established.	NA	NA
sessionId	Yes	The unique number generated for each request.	NA	NA
BrandID	Yes	Represents the unique identifier for the brand name.	NA	""
Attribute	Yes	Specifies the key attribute of the brand.	NA	""

Sample output XML

```
{
  "Status": "0",
  "Description": "Content Fetched Successfully",
  "BrandingDetails": [
    {
      "FontStyle": {
        "FontSize": "14",
        "BackColor": "rgb(12,13,13)",
        "FontName": "Arial",
        "FontColor": "rgb(12,12,12)"
      },
      "Attribute": "sdcd",
      "ExternalHTMLLabelName": "",
      "BrandID": "brand"
    }
  ]
}
```

Output parameters description

Name	Description
description	Provides the description corresponding to status, whether the request was processed successfully or not.
status	Code to ensure whether the request is processed successfully or not.

Name	Description
BrandingDetails	Contains detailed branding information, including font styling and brand attributes. • FontStyle: Specifies the styling of the font, including size, color, and background. • FontSize: Indicates the size of the font in pixels. • BackColor: Sets the background color in RGB format. • FontName: Specifies the name of the font used. • FontColor: Determines the color of the font in RGB format.
Attribute	Represents the specific attribute or characteristic of the brand.
ExternalHTMLLabelName	Represents the name of the external content when branding is enabled.
BrandID	Unique identifier for the brand.

Error code description

Code	Message	Error type
11	Invalid Session	The session is invalid or has expired.
-1011	Cabinet Not Found	The specified cabinet could not be located.
541	Invalid input parameters	The provided input parameters are incorrect or incomplete.
1041	Attribute already exists	The attribute you are trying to add already exists.
400	Operation Failed	A message with the reason for API service failure.

SOAP web services

This chapter describes the SOAP web services used in OmniOMS.

OMS session management (using connectCabinet)

OMSSessionManagement class provides SOAP web service to establish and terminate a session with OmniOMS.

- **Service URL:** *http://localhost:8080/OMSWebServices/soap/OMSSessionManagement?wsdl*
- **Service Methods:**
 - **connectCabinet** — The purpose of this method is to take user credentials and connect the user returning sessionId.
 - **Parameters** — *OMSConnectCabinetInputBO* - represents input details for establishing a session in OmniOMS. Request Object providing details for establishing a session with OmniOMS via service method 'connectCabinet'.
 - **Returns** — *OMSConnectCabinetOutputBO* - provides the details of successful/unsuccessful sessions in OmniOMS.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

OMSConnectCabinetInputBO

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ses="http://sessionManagement.soap.oms.omni.newgen.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <ses:connectCabinet>
```

```

    <arg0>
      <cabinetName>oms</cabinetName>
      <connectForcefully>Y</connectForcefully>
      <userName>abc</userName>
      <userPassword>qwe</userPassword>
      <encrFlag>Y</encrFlag>
      <passAlgoType>1</passAlgoType>
    </arg0>
  </ses:connectCabinet>
</soapenv:Body>
</soapenv:Envelope>

```

Input parameters description

OMSCConnectCabinetInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet in which the session is to be established.	NA	NA
connectForcefully	Yes	Can have a value either, Y or N. Y - if the user wants to terminate the previously established connection with OmniOMS and create a new one. N - if the user does not want to terminate the previously established connection with OmniOMS.	'Y' or 'N'	'N'
userName	Yes	User Id of the user, who needs a session.	NA	NA
userPassword	Yes	Password corresponding to user-id.	NA	NA
encrFlag	No	On the basis of this flag value as Y or N, password encryption will be applied.	NA	N
PassAlgoType	No	Its value can be 1 = MD5, 2-SHA 256, 3 = SHA 512. On the basis of the value, the encrypted password needs to be passed.	NA	0

Sample output XML

OMSCConnectCabinetOutputBO

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:connectCabinetResponse xmlns:ns2="http://
sessionManagement.soap.oms.omni.newgen.com/">
      <return>
        <description/>
        <sessionId>12344</sessionId>
        <status>0</status>
      </return>
    </ns2:connectCabinetResponse>
  </soap:Body>
</soap:Envelope>
```

Output parameters description

OMSCConnectCabinetOutputBO

Name	Description
description	Success
sessionId	The unique number generated for each request.
status	0

Error code description

Code	Message	Comment
1	Unable to connect to the server, check logs.	Check the logs to rectify the server connection failure.

OMS session management (using disconnectCabinet)

OMSSessionManagement class provides SOAP web service to establish and terminate a session with OmniOMS.

- **Service URL:** *http://localhost:8080/OMSWebServices/soap/OMSSessionManagement?wsdl*
- **Service Methods:**
 - **disconnectCabinet** — The purpose of this method is to remove valid sessions within OmniOMS.
 - **Parameters** — *OMSDisconnectCabinetInputBO* - represents the input details for terminating a session with OmniOMS.
 - **Returns** — *OMSDisconnectCabinetOutputBO* - provides the details of successful or unsuccessful requests processed.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

OMSDisconnectCabinetInputBO

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ses="http://sessionManagement.soap.oms.omni.newgen.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <ses:disconnectCabinet>
      <arg0>
        <cabinetName>oms</cabinetName>
        <sessionId>12123</sessionId>
      </arg0>
    </ses:disconnectCabinet>
  </soapenv:Body>
</soapenv:Envelope>
```

Input parameters description

OMSDisconnectCabinetInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet in which the session is to be established.	NA	NA
sessionId	Yes	Session Id of the user logged in to the cabinet.	NA	NA

Sample output XML

OMSDisconnectCabinetOutputBO

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:disconnectCabinetResponse
      xmlns:ns2="http://sessionManagement.soap.oms.omni.newgen.com/">
      <return>
        <description>Success</description>
        <status>0</status>
      </return>
    </ns2:disconnectCabinetResponse>
  </soap:Body>
</soap:Envelope>
```

Output parameters description

OMSDisconnectCabinetOutputBO

Name	Description
description	success
status	0

Error code description

Code	Message	Comment
1	Unable to connect to the server, check logs.	Check the logs to rectify the server connection failure.
11	Invalid Session.	This message appears when an invalid session is passed in the input parameters.

OMS document generation (using DB)

OMSDocumentGeneration class provides SOAP web service which allows integration with other party components for generating documents and sending communications using OmniOMS.

- **Service URL:** *http://localhost:8080/OMSWebServices/soap/OMSDocumentGeneration?wsdl*
- **Service Methods:**
 - **processUsingDB** — This method is to generate documents that are based on a database dataset.
 - **Parameters** — *DocumentGenerationDBInputBO* - Object providing input for document generation based on database dataset. Request Object to receive input for document generation from service method 'processUsingDB'.
 - **Returns** — *DocumentGenerationDBOutputBO* - Object providing the details of the document generation request processed. Response object of document generation request from service method 'processUsingDB'.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:gen="http://generation.soap.oms.omni.newgen.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <gen:processUsingDB>
      <arg0>
        <attachmentsLocation/>
        <cabinetName>oms</cabinetName>
        <dmrqId/>
        <doArchive>N</doArchive>
        <encodedPDFRequired/>
        <fileName>1234</fileName>
        <filePath>D:\Files</filePath>
        <isDmrq/>
        <masterKey>1</masterKey>
        <productName>TST</productName>
        <reportVersion>0</reportVersion>
        <sendMail>Y</sendMail>
        <sessionId>-1234</sessionId>
        <synchronousArchival>N</synchronousArchival>
      </arg0>
    </gen:processUsingDB>
  </soapenv:Body>
</soapenv:Envelope>
```

Input parameters description

DocumentGenerationDBInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
attachmentsLocation	No	Filepath of the External Document to be sent as an additional attachment in the Email.	NA	NA
cabinetName	Yes	Name of the cabinet where document layout is checked in.	NA	NA
communicationId	No	Unique communication ID for the specific transaction.	NA	0

Name	Mandatory (Yes or No)	Description	Valid values	Default value
dateFrom	No	Date from which records will be picked for document generation.	NA	NA
dateTo	No	Date till which records will be picked for document generation.	NA	NA
dmrqlId	No	If isDmrq is provided with value 'Y', then this 'dmrqlId' will be considered as the <i>masterkey</i> .	NA	NA
isDmrq	No	To specify whether user data, for which document is to be generated is in the database tables mapped in the dataset or not. Can have value Y/N. Y - User data for generation will be picked from a different table. For example, if the mapped table is 'SAMPLE', then data will be picked from 'SAMPLE_DMRQ'. N - User data for generation will be picked from the mapped table in the dataset.	'Y' or 'N'	'N'
doArchive	Yes	Can have a value of either Y or N. Y - if the document is to be queued for archival. N - if not required.	'Y' or 'N'	'N'
doRealTimeUpdate	No	Not applicable.	'Y' or 'N'	'N'
productName	Yes	Unique for each on-demand xml dataset for the template.	NA	NA
encodedPDFRequired	No	Can have value either Y or N. Y - Document will be returned as a binary string but will be Base64 encoded. N - No binary form of generated document will be provided.	'Y' or 'N'	'N'
fileName	No	File Name of the document generated, along with the provided name a unique id will be added by CCM. Example, 'filename_requestId'.	NA	NA

Name	Mandatory (Yes or No)	Description	Valid values	Default value
filePath	No	File location where the document needs to be generated.	NA	NA
masterKey	Yes	ID/Primary Key of the user for which document is to be generated, this value will be used to distinguish users.	NA	NA
reportVersion	Yes	By default, or not provided then a recently base-lined layout of the document will be used and generated. For a specific layout, the version needs to be provided	NA	0
sendMail	Yes	For both Mail and SMS processes, this tag is used based upon the report type of the productname passed. Can have value either Y or N. Y - if the document is to be queued for sending mail or SMS. N - if not required.	'Y' or 'N'	'N'
sendWhatsapp	Yes	Can have value either Y or N. Y - if the document is to be queued for sending WhatsApp messages. N - if not required.	'Y' or 'N'	'N'
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA
synchronousArchival	No	This will be used for ondemand synchronous archival only for generation cases. For all other cases, archival will be asynchronous only.	NA	NA

Sample output XML

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:processUsingDBResponse
      xmlns:ns2="http://generation.soap.oms.omni.newgen.com/">
      <return>
        <communicationId/>
      </return>
    </ns2:processUsingDBResponse>
  </soap:Body>
</soap:Envelope>
```

```

        <description/>
        <dmrqId/>
        <encodedFileData/>
        <filePathName/>
        <latestBaselinedVersion/>
        <mailBodyPathName/>
        <requestId/>
        <status>0</status>
        <archivalDescription>Success</archivalDescription>
        <archivalDetails>
            <documentName>archivalDoc</documentName>
            <documentIndex>4</documentIndex>
            <imageIndex>6</imageIndex>
            <volumeIndex>1</volumeIndex>
        </archivalDetails>
    </return>
</ns2:processUsingDBResponse>
</soap:Body>
</soap:Envelope>

```

Output parameters description

DocumentGenerationDBOutputBO

Name	Description
communicationId	Same communication ID that has passed in the input XML.
description	It will provide the description corresponding to status, whether the request was processed successfully or not.
dmrqId	DmrqId provided in input for the user, will be returned.
encodedFileData	Base64 encoded binary form of the generated document.
filePathName	The absolute path of the generated document(non-html).
latestBaselinedVersion	Will return Y or N. Y - The document is generated from the latest base-lined layout. N - The latest base-lined layout is not used for document generation.
mailBodyPathName	The absolute path of the '.html' file, which has the message that will be communicated to the end-user.
requestId	For every generation request, a unique ID is generated by CCM.

Name	Description
status	Code to ensure whether the generation was successful or not. 0 - successfully generated. Non-Zero - some error occurred while processing the request.
archivalDescription	Failed for failure cases, Success for successful archival cases
archivalDetails	This tag will return <i>document name</i> , <i>volumeIndex</i> , <i>documentIndex</i> , and <i>ImageIndex</i> values.

Error code description

Code	Message	Comments
1027	Invalid product name.	The product name is invalid.
1029	Document generation failed. Document type for the given product name is not supported.	Document type for the given product name is not supported.
1028	Either OnDemand Job is not specified for the ProductName, or details for the concerned operation(mail/archival) are incomplete.	Either OnDemand Job is not specified for the ProductName, or details for the concerned operation(mail/archival) are incomplete.
1012	The server is busy and unable to provide a Statement object.	The server is busy and unable to provide a Statement object.
11	Invalid Session.	This message appears when an invalid session is passed in the input parameters.
1034	Synchronous archival will work only for generation cases.	Synchronous archival works only for generation cases.
1035	Synchronous archival will not work for other dms cases.	Synchronous archival does not work for other DMS cases.
-1011	CabinetName not found	The cabinet name is invalid.

OMS document generation (using XML)

OMSDocumentGeneration class provides SOAP web service which allows integration with other party components for generating documents and sending communications using OmniOMS.

- **Service URL:** *http://localhost:8080/OMSWebServices/soap/OMSDocumentGeneration?wsdl*
- **Service Methods:**
 - **processUsingXML** — The method is to generate documents that are based on an XML dataset.
 - **Parameters** — *DocumentGenerationXMLInputBO* - Object providing input for document generation based on XML dataset. Request Object to receive input for document generation from service method '*processUsingXML*'.
 - **Returns** — *DocumentGenerationXMLOutputBO* - Object providing the details of the document generation request processed. Response object of document generation request from service method '*processUsingXML*'.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

DocumentGenerationXMLInputBO

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:gen="http://generation.soap.oms.omni.newgen.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <gen:processUsingXML>
      <arg0>
        <attachmentsLocation/>
        <cabinetName>oms</cabinetName>
        <doArchive>N</doArchive>
        <encodedPDFRequired/>
      </arg0>
    </gen:processUsingXML>
  </soapenv:Body>
</soapenv:Envelope>
```

```

<fileName>abc123</fileName>
<filePath>D:/Files</filePath>
<productName>MNT</productName>
<reportVersion>0</reportVersion>
<sendMail>Y</sendMail>
<sessionId>234</sessionId>
<skipApproval>Y</skipApproval>
<skipEditing>Y</skipEditing>
<xmlData>
    <![CDATA[<Root></Root>]]>
</xmlData>
<synchronousArchival>N</synchronousArchival>
</arg0>
</gen:processUsingXML>
</soapenv:Body>
</soapenv:Envelope>

```

Input parameters description

DocumentGenerationXMLInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
attachmentsLocation	No	Filepath of the External Document to be sent as an additional attachment in the Email.	NA	NA
cabinetName	Yes	Name of the cabinet where document layout is checked-in.	NA	NA
communicationId	No	Unique communication Id for the specific transaction.	NA	0
dateFrom	No	Not applicable.	NA	NA
dateTo	No	Not applicable.	NA	NA
doArchive	Yes	Can have value either Y or N. Y - if the document is to be queued for archival. N - if not required.	'Y' or 'N'	'N'

Name	Mandatory (Yes or No)	Description	Valid values	Default value
encodedPDFRequired	No	Can have a value of either Y or N. Y - The document will be returned as a binary string but will be Base64 encoded. N - No binary form of the generated document will be provided.	'Y' or 'N'	'N'
doRealTimeUpdate	No	Not applicable.	'Y' or 'N'	'N'
filename	No	File Name of the document to be generated, along with the provided name a unique id will be added by CCM. Example, 'filename_requestId'.	NA	NA
filePath	No	File location where the document needs to be generated.	'Y' or 'N'	'N'
productName	Yes	Unique for each on-demand XML dataset for the template.	NA	NA
reportVersion	Yes	By default, or not provided then the recently base-lined layout of the document will be used and generated. For a specific generation, a layout version needs to be provided.	NA	0
saveClientData	No	If Y: insert the xmlData provided into the <i>inputDataXML</i> column of the <i>omsJobQueueItems</i> table.	'Y' or 'N'	'N'
sendMail	Yes	For both Mail and SMS processes, this tag is used based upon the report type of the <i>productname</i> passed. Can have a value of either Y or N. Y - if the document is to be queued for sending mail or SMS. N - if not required.	'Y' or 'N'	'N'
sendWhatsapp	Yes	Can have a value of either Y or N. Y - if the document is to be queued for sending WhatsApp messages. N - if not required.	'Y' or 'N'	'N'

Name	Mandatory (Yes or No)	Description	Valid values	Default value
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA
skipApproval	Yes	Useful only for Interactive Editor flow. Y - if the document does not need to go through the maker-checker process. N - if the document needs to go through the maker-checker process.	'Y' or 'N'	'Y'
skipEditing	Yes	Useful only for Interactive Editor flow. Y - if the document will not be allowed for further editing after generation. N - if the document will be allowed to go through editing after generation.	'Y' or 'N'	'Y'
xmlData	Yes	User data for which the document is to be generated.	NA	NA
synchronousArchival	No	This will be used for ondemand synchronous archival only for generation cases. For all other cases, archival will be asynchronous only.	NA	NA

Sample output XML

DocumentGenerationXMLOutputBO

```

<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:processUsingXMLResponse
      xmlns:ns2="http://generation.soap.oms.omni.newgen.com/">
      <return>
        <communicationId/>
        <description/>
        <dmrqId/>
        <encodedFileData/>
        <filePathName/>
        <latestBaselinedVersion/>
        <mailBodyPathName/>
        <requestId/>
        <status>0</status>
        <archivalDescription>Success</archivalDescription>
      </return>
    </ns2:processUsingXMLResponse>
  </soap:Body>
</soap:Envelope>

```

```

    <archivalDetails>
      <documentName>archivalDoc</documentName>
      <documentIndex>4</documentIndex>
      <imageIndex>6</imageIndex>
      <volumeIndex>1</volumeIndex>
    </archivalDetails>
  </return>
</ns2:processUsingXMLResponse>
</soap:Body>
</soap:Envelope>

```

Output parameters description

DocumentGenerationXMLOutputBO

Name	Description
communicationId	Same communication ID that has passed in the input XML.
description	Will provide the description corresponding to status, whether the request was processed successfully or not.
encodedDocxData	Base64 encoded binary-form of generated docx document(non-html).
encodedFileData	Base64 encoded binary-form of generated PDF document(non-html).
encodedMailBodyData	Base64 encoded binary-form of the generated document (HTML only).
filePathName	The absolute path of the generated document(non-html).
latestBaselinedVersion	Will return Y/N. Y - The document is generated from the latest base-lined layout. N - The latest base-lined layout is not used for document generation.
mailBodyPathName	The absolute path of the '.html' file, which has the message that will be communicated to the end-user.
requestId	For every generation request, a unique ID is generated by CCM.
status	Code to ensure whether the generation was successful or not. 0 - successfully generated. Non-Zero - Some error occurred while processing the request.
transactionId	Will return a numeric ID of the generated document for contract and letter only.
archivalDescription	Failed for failure cases, Success for successful archival cases

Name	Description
archivalDetails	This tag will return the following values: • <code>documentName</code> — The name of the document. • <code>volumeIndex</code> — The volume index of the attachment. • <code>documentIndex</code> — A unique index of the document that the user wants to attach. • <code>imageIndex</code> — The image index of the attachment.

Error code description

Code	Message	Comment
1027	Invalid product name.	The product name is invalid.
1029	Document generation failed. Document type for the given product name is not supported.	Document type for the given product name is not supported.
1028	Either OnDemand Job is not specified for the ProductName, or details for the concerned operation(mail/archival) are incomplete.	Either OnDemand Job is not specified for the ProductName, or details for the concerned operation(mail/archival) are incomplete.
1012	The server is busy and unable to provide a Statement object.	The server is busy and unable to provide a Statement object.
11	Invalid Session.	This message appears when an invalid session is passed in the input parameters.
1034	Synchronous archival will work only for generation cases.	Synchronous archival works only for generation cases.
1035	Synchronous archival will not work for other dms cases.	Synchronous archival does not work for other DMS cases.
-1011	CabinetName not found	The cabinet name is invalid.

OMS document generation (using JSON)

OMSDocumentGeneration class provides SOAP web service which allows integration with other party components for generating documents and sending communications using OmniOMS.

- **Service URL:** *http://localhost:8080/OMSWebServices/soap/OMSDocumentGeneration?wsdl*
- **Service Methods:**
 - **processUsingJSON** — The method is to generate documents that are based on the JSON dataset.
 - **Parameters** — *DocumentGenerationJSONInputBO* - Object providing input for document generation based on JSON dataset. Request Object to receive input for document generation from service method 'processUsingJSON'.
 - **Returns** — *DocumentGenerationJSONOutputBO* - Object providing the details of the document generation request processed. Response object of document generation request from service method 'processUsingJSON'.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

DocumentGenerationJSONInputBO

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:gen="http://generation.soap.oms.omni.newgen.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <gen:processUsingJSON>
      <arg0>
        <attachmentsLocation/>
        <cabinetName>oms</cabinetName>
        <doArchive>N</doArchive>
      </arg0>
    </gen:processUsingJSON>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        <encodedPDFRequired/>
        <fileName>abc123</fileName>
        <filePath>D:/Files</filePath>
        <productName>MNT</productName>
        <reportVersion>0</reportVersion>
        <sendMail>Y</sendMail>
        <sessionId>234</sessionId>
        <skipApproval>Y</skipApproval>
        <skipEditing>Y</skipEditing>
        <jsonData>
        {
        "Agreements": {
        "RentAgreement": {
        "Name": "user1",
        "Rent": "2000",
        "MailTo": "abc@domain.com"}
        }
        }
        </jsonData>
        <synchronousArchival>N</synchronousArchival>
    </arg0>
</gen:processUsingJSON>
</soapenv:Body>
</soapenv:Envelope>

```

Input parameters description

DocumentGenerationJSONInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
attachmentsLocation	No	Filepath of the External Document to be sent as an additional attachment in the Email.	NA	NA
cabinetName	Yes	Name of the cabinet where document layout is checked-in.	NA	NA
communicationId	No	Unique communication Id for the specific transaction.	NA	0
dateFrom	No	Not applicable.	NA	NA
dateTo	No	Not applicable.	NA	NA

Name	Mandatory (Yes or No)	Description	Valid values	Default value
doArchive	Yes	Can have a value of either Y or N. Y - if the document is to be queued for archival. N - if not required.	'Y' or 'N'	'N'
encodedPDFRequired	No	Can have a value of either Y or N. Y - The document will be returned as a binary string but will be Base64 encoded. N - No binary form of the generated document will be provided.	'Y' or 'N'	'N'
doRealTimeUpdate	No	Not applicable.	'Y' or 'N'	'N'
filename	No	File Name of the document to be generated, along with the provided name a unique ID will be added by CCM. Example, 'filename_requestId'.	NA	NA
filePath	No	File location where the document needs to be generated.	'Y' or 'N'	'N'
productName	Yes	Unique for each on-demand JSON dataset for the template.	NA	NA
reportVersion	Yes	By default, or not provided then the recently base-lined layout of the document will be used and generated. For a specific generation, a layout version needs to be provided.	NA	0
saveClientData	No	If Y: insert the <i>jsonData</i> provided into <i>inputDataXML</i> column of <i>omsJobQueueItems</i> table.	'Y' or 'N'	'N'
sendMail	Yes	For both Mail and SMS processes, this tag is used based upon the report type of the <i>productname</i> passed. Can have a value of either Y or N. Y - if the document is to be queued for sending mail or SMS. N - if not required.	'Y' or 'N'	'N'

Name	Mandatory (Yes or No)	Description	Valid values	Default value
sendWhatsapp	Yes	Can have a value of either Y or N. Y - if the document is to be queued for sending WhatsApp messages. N - if not required.	'Y' or 'N'	'N'
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA
skipApproval	Yes	Useful only for Interactive Editor flow. Y - if the document does not need to go through the maker-checker process. N - if the document needs to go through the maker-checker process.	'Y' or 'N'	'Y'
skipEditing	Yes	Useful only for Interactive Editor flow. Y - if the document will not be allowed for further editing after generation. N - if the document will be allowed to go through editing after generation.	'Y' or 'N'	'Y'
jsonData	Yes	User data for which the document is to be generated.	NA	NA
synchronousArchival	No	This will be used for ondemand synchronous archival only for generation cases. For all other cases, archival will be asynchronous only.	NA	NA

Sample output XML

DocumentGenerationJSONOutputBO

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:processUsingJSONResponse
      xmlns:ns2="http://generation.soap.oms.omni.newgen.com/">
      <return>
        <communicationId/>
        <description/>
        <dmrqId/>
        <encodedFileData/>
        <filePathName/>
      </return>
    </ns2:processUsingJSONResponse>
  </soap:Body>
</soap:Envelope>
```

```

        <latestBaselinedVersion/>
        <mailBodyPathName/>
        <requestId/>
        <status>0</status>
        <archivalDescription>Success</archivalDescription>
        <archivalDetails>
            <documentName>archivalDoc</documentName>
            <documentIndex>4</documentIndex>
            <imageIndex>6</imageIndex>
            <volumeIndex>1</volumeIndex>
        </archivalDetails>
    </return>
</ns2:processUsingJSONResponse>
</soap:Body>
</soap:Envelope>

```

Output parameters description

DocumentGenerationJSONOutputBO

Name	Description
communicationId	Same communication Id that has passed in the input XML.
description	Will provide the description corresponding to status, whether the request was processed successfully or not.
encodedDocxData	Base64 encoded binary-form of generated docx document(non-html).
encodedFileData	Base64 encoded binary-form of generated PDF document(non-html).
encodedMailBodyData	Base64 encoded binary-form of the generated document (HTML only).
filePathName	The absolute path of the generated document(non-html).
latestBaselinedVersion	Will return Y or N. Y - The document is generated from the latest base-lined layout. N - The latest base-lined layout is not used for document generation.
mailBodyPathName	The absolute path of the '.html' file, which has the message that will be communicated to the end-user.

Name	Description
requestId	For every generation request, a unique ID is generated by CCM.
status	Code to ensure whether the generation was successful or not. 0 - successfully generated. Non-Zero - Some error occurred while processing the request.
transactionId	Will return a numeric ID of the generated document for contract and letter only.
archivalDescription	Failed for failure cases, Success for successful archival cases
archivalDetails	This tag will return the following values: • documentName — The name of the document. • volumeIndex — The volume index of the attachment. • documentIndex — A unique index of the document that the user wants to attach. • imageIndex — The image index of the attachment.

Error code description

Code	Message	Comment
1027	Invalid product name.	The product name is invalid.
1029	Document generation failed. Document type for the given product name is not supported.	Document type for the given product name is not supported.
1028	Either OnDemand Job is not specified for the ProductName, or details for the concerned operation(mail/archival) are incomplete.	Either OnDemand Job is not specified for the ProductName, or details for the concerned operation(mail/archival) are incomplete.
1012	The server is busy and unable to provide a Statement object.	The server is busy and unable to provide a Statement object.
11	Invalid Session.	This message appears when an invalid session is passed in the input parameters.
1034	Synchronous archival will work only for generation cases.	Synchronous archival works only for generation cases.

Code	Message	Comment
1035	Synchronous archival will not work for other dms cases.	Synchronous archival does not work for other DMS cases.
-1011	CabinetName not found	The cabinet name is invalid.

OMS job auto approval

OMSJobAutoApproval class provides SOAP web service for spooling the cycle Instance and approving it without logging into OMS Admin Console.

- **Service URL:** *http://localhost:8080/OMSWebServices/soap/OMSJobAutoApproval?wsdl*
- **Service Methods:**
 - **approveJob** — This service method is used to approve a job instance.
 - **Parameters** — *OMSAApproveJobInputBO* - Object providing details for job instance to be approved. Request Object to receive input for approving job instance by service method '*approveJob*'.
 - **Returns** — *OMSAApproveJobOutputBO* - Object providing details of the approval request. Response Object provides details, on how the request is processed using the service method '*approveJob*'.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

OMSAApproveJobInputBO

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:job="http://jobapproval.soap.oms.omni.newgen.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <job:approveJob>
      <arg0>
        <cabinetName>oms</cabinetName>
        <communicationId>19</communicationId>
        <cycleName>cycle1</cycleName>
      </arg0>
    </job:approveJob>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        <doRealTimeUpdate>N</doRealTimeUpdate>
        <filterCondition></filterCondition>
        <instanceId>2021-02-01 00:00:00.000</instanceId>
        <jobName>new</jobName>
        <jobType>M</jobType>
        <sessionId>1234</sessionId>
    </arg0>
</job:approveJob>
</soapenv:Body>
</soapenv:Envelope>

```

Input parameters description

OMSApproveJobInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet, in which the job exists.	NA	NA
communicationId	No	Unique communication Id for the specific transaction.	NA	0
cycleName	Yes	Name of the cycle, in which the mentioned job is present.	NA	NA
doRealTimeUpdate	No	Not applicable.	'Y' or 'N'	'N'
filterCondition	No	If applied then based upon this filter condition, records will be processed.	NA	NA
instanceId	Yes	Instance ID specifies the date on which the job instance is created. The format of the ID is 'YYYY-MM-DD 00:00:00'.	NA	NA
jobName	Yes	As the name suggests, the name of the job to be approved.	NA	NA
jobType	Yes	As the name suggests, type of the job. P - for print job. M - for mail job. T - for text job. W – for WhatsApp job.	NA	NA
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA

Sample output XML

OMSApproveJobOutputBO

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  <soap:Body>
    <ns2:approveJobResponse
      xmlns:ns2="http://jobapproval.soap.oms.omni.newgen.com/"
      <return>
        <communicationId/>
        <description/>
        <jobSpoolId xsi:nil="true"/>
        <status>0</status>
      </return>
    </ns2:approveJobResponse>
  </soap:Body>
</soap:Envelope>
```

Output parameters description

OMSApproveJobOutputBO

Name	Description
communicationId	Same communication Id that has passed in the input XML.
description	Will provide the description corresponding to status, whether the request was processed successfully or not.
jobSpoolId	Every spooled job instance is given an Id in OmniOMS. This Id is related to instance Id.
status	Code to ensure whether the request is processed successfully or not. 0 - successfully generated. Non-Zero - some error occurred while processing the request.

Error code description

Code	Message	Comment
400/804 (Sub error code)	The cycle name provided does not exist.	The cycle name provided does not exist.
400	InstanceId provided corresponding to the cycle name does not exist.	InstanceId provided corresponding to the cycle name does not exist.
400	JobType is not provided.	JobType is not provided.
1099	Duplicate Communication Id.	The communication ID is duplicate.

OMS message tracker

OMSMessageTracker Class provides a SOAP web service to update the status of messages in DB if tracking is enabled in the database. Status against each message is updated in DB as Delivered or Not Delivered based upon the status passed in Input.

- **Service URL:** *http://localhost:8080/OMSWebServices/soap/OMSMessageTracker?wsdl*
- **Service Methods:**
 - **updateMessageTrackerStatus:**
 - **Parameters** — *OMSMessageTrackerInputBO* - Request Object to receive input for document generation from service method 'updateMessageTrackerStatus'.
 - **Returns** — *OMSMessageTrackerOutputBO* - Response Object to update the status of the request.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

OMSMessageTrackerInputBO

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:mes="http://MessageTracker.soap.oms.omni.newgen.com/"
  <soapenv:Header/>
  <soapenv:Body>
    <mes:updateMessageTrackerStatus>
      <arg0>
        <cabinetName>oms</cabinetName>
        <messageDetails>
          <messageDetail>
            <remarks></remarks>
            <status>C</status>
            <trackid>1234</trackid>
          </messageDetail>
          <messageDetail>
            <remarks></remarks>
            <status>E</status>
            <trackid>4567</trackid>
          </messageDetail>
        </messageDetails>
        <SessionId>5434</SessionId>
      </arg0>
    </mes:updateMessageTrackerStatus>
  </soapenv:Body>
</soapenv:Envelope>
```

Input parameters description

OMSMessageTrackerInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet where the message will be generated.	NA	NA
messageDetail	Yes	Message details as tracking ID and status.	NA	NA
Trackid	Yes	Tracking ID for communication.	NA	NA

Name	Mandatory (Yes or No)	Description	Valid values	Default value
remarks	No	In case any specific remarks to be updated in DB against each message.	NA	NA
Status	Yes	Based upon this status value, the message is marked Delivered or not in DB. C – Complete E – Error	C, E	E
SessionId	Yes	Session Id of the user logged in the cabinet.		NA

Sample output XML

OMSMessageTrackerOutputBO

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:updateMessageTrackerStatusResponse
      xmlns:ns2="http://MessageTracker.soap.oms.omni.newgen.com/">
      <return>
        <description/>
        <mainCode/>
        <status>0</status>
        <updatedRecords/>
      </return>
    </ns2:updateMessageTrackerStatusResponse>
  </soap:Body>
</soap:Envelope>
```

Output parameters description

OMSMessageTrackerOutputBO

Name	Description
description	Successfully updated.
mainCode	'0'
status	'0'

Name	Description
updatedRecords	Status updated successfully.

Error code description

Code	Message	Comment
1010	Either the message Id is not provided or does not match with the database.	Either the message ID is not provided or does not match with the database.

OMS on demand XML validation

OMSONDemandXMLValidation Class provides SOAP web service to validate XML against an associated XSD schema file.

- **Service URL:** *http://localhost:8080/OMSWebServices/soap/OMSONDemandXMLValidation?wsdl*
- **Service Methods:**
 - **xmlValidation:**
 - **Parameters** — *OMSONDemandXMLValidationInputBO* - Request object to take XML as input for validation against a specific template.
 - **Returns** — *OMSONDemandXMLValidationOutputBO* - Response object to return whether input XML is valid or not.
- **Method Type:** POST
- **Consumes:** application/xml
- **Produces:** application/xml

Sample input XML

OMSONDemandXMLValidationInputBO

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ond="http://ondemandxmlvalidation.soap.oms.omni.newgen.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <ond:xmlValidation>
```

```

    <arg0>
      <cabinetName>oms</cabinetName>
      <productName>new</productName>
      <sessionId>4566</sessionId>
      <xmlData></xmlData>
    </arg0>
  </ond:xmlValidation>
</soapenv:Body>
</soapenv:Envelope>

```

Input parameters description

OMSONDemandXMLValidationInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet where document layout is checked-in.	NA	NA
productName	Yes	Unique for each on-demand XML dataset for the template.	NA	NA
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA
xmlData	Yes	User data for which the document is to be generated.	NA	NA

Sample output XML

OMSONDemandXMLValidationOutputBO

```

<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:xmlValidationResponse
      xmlns:ns2="http://ondemandxmlvalidation.soap.oms.omni.newgen.com/">
      <return>
        <description xsi:nil="true"/>
        <status>0</status>
        <validationStatus>true</validationStatus>
      </return>
    </ns2:xmlValidationResponse>
  </soap:Body>
</soap:Envelope>

```

Output parameters description

OMSONdemandXMLValidationOutputBO

Name	Description
description	Success
status	0
validationStatus	True or Error

Error code description

Code	Message	Comment
1011	Invalid XML corresponding to the Product Name provided.	The XML is invalid corresponding to the provided product name.
-1011	CabinetName not found	The cabinet name is invalid.

OMS on demand data purge

OMSONdemandDataPurge class provides SOAP web service to purge data from OmniOMS tables. The data of completed jobs move to the history table.

- **Service URL:** *http://localhost:8080/OMSWebServices/soap/OMSONdemandDataPurge?wsdl*
- **Service Methods:**
 - **dataPurge** — This service method is used to process data purge requests of moving data of completed jobs to the history table.
 - **Parameters** — *OMSDDataPurgeInputBO* - Object providing details for the request to move completed job data. Request Object to receive input for purging data from service method 'dataPurge'.
 - **Returns** — *OMSDDataPurgeOutputBO* - Object providing details of whether the request was processed successfully or not.
- **Method Type:** POST
- **Consumes:** application/xml

- **Produces:** application/xml

Sample input XML

OMSDDataPurgeInputBO

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:dat="http://dataPurge.soap.oms.omni.newgen.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <dat:dataPurge>
      <arg0>
        <cabinetName>oms</cabinetName>
        <dateFrom>2020-12-11 00:00:00.000</dateFrom>
        <dateTo>2020-12-21 00:00:00.000</dateTo>
        <reportType></reportType>
        <requiredFlag>N</requiredFlag>
        <sessionId>1234</sessionId>
      </arg0>
    </dat:dataPurge>
  </soapenv:Body>
</soapenv:Envelope>
```

Input parameters description

OMSDDataPurgeInputBO

Name	Mandatory (Yes or No)	Description	Valid values	Default value
cabinetName	Yes	Name of the cabinet, of which data need to be purged.	NA	NA
dateFrom	No	Completed jobs that are spooled on and after this, a date will be picked for purging. Format of Id is 'YYYY-MM-DD 00:00:00' If not provided, purging will begin from the very first completed job.	NA	NA
dateTo	Yes	Completed jobs that are spooled before this date will be purged. The format of ID is 'YYYY-MM-DD 00:00:00'	NA	NA

Name	Mandatory (Yes or No)	Description	Valid values	Default value
reportType	No	Type of the report that will be used for the purging. No need to pass any value except for Letter or Contract Report types.	L, C	Empty ("")
requiredFlag	Yes	Based upon this column, will insert records into the OMSJobQueueItemsHistory table.	'Y' or 'N'	'N'
sessionId	Yes	Session Id of the user logged in the cabinet.	NA	NA

Sample output XML

OMSDDataPurgeOutputBO

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:dataPurgeResponse
      xmlns:ns2="http://dataPurge.soap.oms.omni.newgen.com/">
      <return>
        <description/>
        <status>0</status>
      </return>
    </ns2:dataPurgeResponse>
  </soap:Body>
```

Output parameters description

OMSDDataPurgeOutputBO

Name	Description
Description	OnDemandGeneration data purged successfully.
Status	'0' for success.

Error code description

Code	Message	Comment
1	Unable to connect to the server, check logs.	Check the logs to rectify the server connection failure.
11	Invalid Session	This message appears when an invalid session is passed in the input parameters.

Implementation of Authentication Token and pre-Hook in OmniOMS REST Web Services

This chapter provides information on how to implement Authentication Token and pre-Hook in OmniOMS REST Web Services.

Prerequisites

OmniOMS 12.0 must be installed in the system.

Implementing Authentication Token

This chapter describes the implementation of Authentication Token in OmniOMS REST Web Services.

Implementation steps

To implement Authentication Token Parser, follow the below steps:

1. Setup a Java project in Eclipse or any other supported Integrated Development Environment (IDE).
2. Create a class with the name *TokenParser* in the package *com.newgen.authToken*. This class implements an interface *AuthTokenParser*.



The *AuthTokenParser* interface is available in the *OMSUtility.jar* file present in the common library of the application server.

3. Implement the following methods of the *AuthTokenParser* interface:

```
boolean isValidToken(String authToken)
{
}

HashMap<String, String> fetchUserPropfromToken(String authToken)
{
}
```

4. After implementing the *TokenParser.java* class:

- Stop the application server.
- Compile the Java project and create a .jar file. Then place this file at the library location of the respective application server.

5. Start the application server.

Execution impact and expectations

Implementation of an Authentication Token Parser in REST web services has the following execution impacts and expectations:

- Pass the *<authToken>* and *<authTokenType>* key or tag values in the REST Web Services request call.



When you pass *authToken* and *authTokenType* tags to an API other than *executeAPI*, it validates the authentication token and then proceeds only if it is valid.

- The Web Service expects the following while generation of session ID using *connectCabinet* call in *OMSSessionManagement* API REST Web Service:
 - Use a valid Authentication Token.
 - The Authentication Token must return a HashMap having the key value “username” and its value a “Domain UserName”.
 - The *NGOConnectCabinet* call input request must be in the following XML format:

```
<OMSConnectCabinetInputBO><cabinetName>cabinet_name</cabinetName><userName>username</userName><userPassword>password</userPassword><connectForcefully>Y</connectForcefully><authToken>xxxx</authToken><authTokenType>JWT</authTokenType></OMSConnectCabinetInputBO>
```

Implementing pre-Hook

This chapter describes the implementation of pre-hook in OMS REST Web Services. It further explains how to intercept and extend OmniOMS API. The component created for this is henceforth referred to as API Hook.

OmniOMS server architecture provides a framework to extend or customize the behavior of standard XML-based APIs. You can use this framework to perform tasks in addition to the standard behavior as a part of the same transaction. For example, accessing or updating tables, extending user authentication policies, validations, and so on.

Requirement of classes and components

The following are the requirements of classes and components for hook implementations:

- Hook implementation requires *HookClass*, *HookHandler*, and *HookImplementer* classes. These classes are available with OmniOMS server setup in the *omnishared.jar* file at the following locations:
 - For OmniOMS-JBOSS application server setup -
`<Drive :>/<Installation directory of JBOSS application server>/modules/
o2ms_library/main`
 - For OmniOMS-WebSphere application server setup –
`<Drive :>/<Installation directory of WebSphere application server>/
AppServer/profiles/<WebSphere application server profile directory>/
<OmniOMS Shared Library directory>`
 - For OmniOMS-WebLogic application server setup –
`<Drive :>/<Installation directory of WebLogic application server>/
user_projects/domains/<OmniOMS domain directory> Jars`
- Add the corresponding *HookName* tag in *HookData.xml*.
- Restart the application server instance.

Implementation steps

The *omnishared.jar* file contains *HookClass*, *HookHandler*, and *HookImplementer* classes in the package *com.newgen.omni.jts.srvr*.

HookHandler contains an object of *HookClass* namely *hookObject*. *HookImplementer* is a singleton for reading Hook properties files that stores the names of all API Hooks that must get executed.

To create a hook for a particular transaction, follow the below steps:

1. Specify the name of the Hook. If the name of the transaction is *OMSxyz*, then name the Hook class as *HOOKxyz*.

For example, if the transaction class is *OMSConnectCabinet*, then the corresponding hook class must be *HOOKConnectCabinet*.



Replace the first three characters of the transaction name with the string "Hook".

2. Define the Hook class under the package name *com.newgen.omni.jts.hook*.
3. Extend the abstract base class *com.newgen.omni.jts.srvr.HookHandler* using the hook class.
4. Define the abstract base class *HookHandler* using the following methods:
 - *abstract public String executeIn(String strInputXML);*
 - *abstract public String executeOut(String strOutputXML);*

The user-defined hook class must provide the implementation for both methods.

5. Use the *executeIn()* method for any processing that must be done prior to the execution of the actual OmniOMS transaction.
The minimum implementation for these two methods must be to return the input String, that is, the string passed in the input parameter.
6. Call the *setStatus(int)* and *setMessage(String)* on the object *hookObject* to throw any error within the following methods:
 - *hookObject.setStatus(-78999);*
 - *hookObject.setMessage("Test For Error");*

In this case, the outputXML returned to the client contains the error code along with the given message.



"hookObject" is the object of "HookClass" declared in "HookHandler" class that is available by default.

7. After creating the hook class and packaging it appropriately, add the Hook class name to the *HookData.xml* file available at the following locations:

- For OmniOMS-JBOSS application server setup –
<Drive :>/< Installation directory of JBOSS application server>/bin/ngdbini
- For OmniOMS-WebSphere application server setup –
*<Drive :>/< Installation directory of WebSphere application server>/AppServer/
profiles/<WebSphere application server profile directory>/ ngdbini*
- For OmniOMS-WebLogic application server setup –
*<Drive:>/<Installation directory of WebLogic application server>/user_projects/
domains/<OmniDocs domain Directory>/ ngdbini*



HookData.xml files must exist at the specified locations and each Hook entry must be complete before the Hook gets executed.

Example of Hook

Code for HOOKAddUser class -

It is necessary to close the *Statements* and *ResultSet* objects only using the Connection Object, not the Connection Object itself, if it is used to pass to Hook class.

Custom tab integration

This section provides information on the steps involved in adding custom tabs to the menu bar within the OmniOMS Admin Workspace. These customized tabs enable the display of information from external pages within the Admin Workspace module.

The recommended layout guidelines for the custom tabs are as follows:



- For an enhanced user experience, there must be no header, footer, or left menu in the custom application.
- Design the custom application as a complete web page, occupying the entire width and height of the screen.
- Support for scrollable content allows for seamless navigation.
- The custom tab must have a valid http or https URL that hosts an external React application.

Integrating custom tabs in OmniOMS Admin Workspace

To integrate a custom tab in the OmniOMS Admin Workspace module, follow the below steps:

1. Open the *OMSAdminCustomConfig.json* file available in the *OMSConfig* folder on the application server.
2. Refer to the table below to input values for the parameters listed in the *OMSAdminCustomConfig.json* file:

Parameter	Description
name	Specify the name for the custom tab to appear in the Admin Workspace module. For example: CustomTab. • It is mandatory to provide the name of the custom tab. If the name value is left empty, the custom tab is not displayed in the Admin Workspace module. • The name of the custom tab must be limited to a maximum of nine characters, including spaces. If the name exceeds nine characters, the system does not display the custom tab. • Special characters are not permitted when naming the custom tab.

Parameter	Description
link	Specify the name of the tab that must appear in the path of the external application URL. ! Each custom tab must have a unique link.
URL	Specify the URL of the external React application to integrate and display it in the Admin Workspace. For example: <i>http://127.0.0.1:8080/counter-app</i> • The external React application must be hosted on the same server, that is, it must have the same IP address and port as the Admin Workspace. If the external application is hosted on a different IP address and port, the system displays the "Request Denied" error. • In case of an invalid URL, the system displays the "Page not found" error.
iconURL	Specify the URL of the icon to add it to the custom tab. For example: <i>https://live.staticflickr.com/4561/38054606355_26429c884f_b.jpg</i> ! The system automatically adds a placeholder image as the default icon if you do not specify a URL.

3. Save the changes and close the file.
4. Restart the application server to reflect the changes in the OmniOMS Admin Workspace.

Sample of OMSAdminCustomConfig.json without any custom tab integration

```
{
  "customTabs" : [
  ]
}
```

Sample of OMSAdminCustomConfig.json with custom tab integration

```
{
  "customTabs" : [
  {
    "name": "CustTab1",
    "link": "customTab1",
    "URL": "http://127.0.0.1:8080/counter-app",
    "iconURL": ""
  },
  {
```

```

"name": "CustTab2",
"link": "customTab2",
"URL": "http://127.0.0.1:8080/authenticator-app",
"iconURL": " https://live.staticflickr.com/4561/38054606355_26429c884f_b.jpg"
},
]
}

```

Configuring session validity

Before navigating to the external page, it is essential to verify the validity of the session. Once verified, pass the *SessionID*, *Cabinet Name*, and *ThemeConfig* (basic color codes used within the Admin Workspace) to the external React application.

Here are the different scenarios for session validity:

- If the value of the parameter *isEncryptForCustom* in the *AdminConfig.ini* file is set to "N," refer to [Sample for fetching the data in a child external application](#).
- If the value of the parameter *isEncryptForCustom* in the *AdminConfig.ini* file is set to "Y," refer to [Sample for fetching the data in the child external application in the case of encryption](#).

Ensure to follow the appropriate sample based on the encryption configuration to handle the session validation and data retrieval process.

Sample for fetching the data in a child external application

To fetch the data in a child external application, follow the below steps:

1. Create a state variable within your functional components using the state hook with initial values.

```

const [sessionId, setSessionId] = React.useState("")
const [cabinetName, setCabinetName] = React.useState("")
const [ themeConfig, setThemeConfig] = useState("")

```

2. Add WindowListener to fetch the parameters coming from OMSAdminWorkspace.

```

window.addEventListener('message', function(event) {
    setSessionId(event.data.SessionId)
    setCabinetName(event.data.CabinetName)

```

```
setThemeConfig(event.data.themeConfig)
});
```

3. Call this API for session validation before making any other API call:

- a. Install Axios package if not present using `npm install axios`.

Add import for axios at the top.

```
import axios from 'axios';
```

- b. Add the following handler before making any API call:

```
const body = JSON.stringify({
  CabinetName: cabinetName,
  SessionId: sessionId,
  AuthID: sessionId
});
const config = {
  headers: { 'Content-Type': 'application/json' }
}
const url =
//IP address and port must be same on which Admin Workspace and Custom
application is deployed//
"http://<IP Address>:<Port>/OMSAdminConsole/rest/OMSAdminServices/
getUserDetails"
axios.post(url, body, config)
  .then(response => {
    if(parseInt(response.data.Status) === 0) {
      { YOUR API CALL}
    } else if(parseInt(response.data.Status) === -1) {
      window.parent.postMessage("Invalid Session", "*");
    } else {
      window.parent.postMessage("Some Error Occured", "*");
    }
  })
  .catch(error => {
    console.log(error);
  });
```

4. After the call, notify the parent window (OMSAdminWorkspace) that the session is invalid by using the above method and invalid session page is displayed based on that.

Sample for fetching the data in a child external application in the case of encryption

To fetch the data in a child external application in the case of encryption, follow the below steps:

1. Create a state variable within your functional components using the state hook with initial values.

```
const [sessionId, setSessionId] = React.useState("")
const [cabinetName, setCabinetName] = React.useState("")
const [themeConfig, setThemeConfig] = useState("")
const [TokenValue, setTokenValue] = useState("")
```

2. Add WindowListener to fetch the parameters coming from the OMSAdminWorkspace.

```
window.addEventListener('message', function(event) {
    setTokenValue(event.data.SessionId)
    setCabinetName(event.data.CabinetName)
    setThemeConfig(event.data.themeConfig)
    const body = JSON.stringify({
        CabinetName: event.data.CabinetName,
        SessionId: event.data.SessionId,
        ActionFlag:"D",
    });
    const config = {
        headers: { 'Content-Type': 'application/json' }
    }

    const url =
//IP address and port must be same on which Admin Workspace and Custom
application is deployed//
"http://<IP Address>:<Port>/OMSAdminConsole/rest/OMSAdminServices/ManageToken"
    axios.post(url, body, config)
    .then(res => {
        setSessionId(res.data.sessionid);
    })
    .catch(error => {
        console.log(error);
    })
});
```

3. Call this API for session validation before making any other API call:
 - a. Install Axios package if not present using `npm install axios`.
Add import for axios at the top.

```
import axios from 'axios';
```

- b. Add the following handler before making any API call:

```
const body = JSON.stringify({
  CabinetName: cabinetName,
  SessionId: sessionId,
  AuthID: sessionId
});
const config = {
  headers: { 'Content-Type': 'application/json' }
}
const url =
//IP address and port must be same on which Admin Workspace and Custom
application is deployed//
"http://<IP Address>:<Port>/OMSAdminConsole/rest/OMSAdminServices/
getUserDetails"
axios.post(url, body, config)
  .then(response => {
    if(parseInt(response.data.Status) === 0) {
      { YOUR API CALL}
    } else if(parseInt(response.data.Status) === -1) {
      window.parent.postMessage("Invalid Session", "*");
    } else {
      window.parent.postMessage("Some Error Occured", "*");
    }
  })
  .catch(error => {
    console.log(error);
  });
```

4. After the call, notify the parent window (OMSAdminWorkspace) that the session is invalid by using above method and invalid session page is displayed based on that.

Implementing URL shortening

NewgenONE OmniOMS enables the implementation of URL shortening within reports designed in the Composition Designer. It includes built-in support for the Bitly URL shortening service. Additionally, it offers the flexibility to configure and use other URL-shortening services.

This chapter provides information on implementing URL shortening when using services other than Bitly.

To implement and use URL shortening service other than Bitly, follow the below steps:

1. Specify the implementation class.

You can refer to the Bitly parameter configuration as a reference to understand the required format and structure. Depending on the URL shortening service, specify the credentials or authentication details required to access and use the service. This includes API keys, access tokens, or other relevant authentication information.



The initial parameter, *UrlShortImplClass*, must be assigned the path of the implementation class. For example: *UrlShortImplClass: com.newgen.omni.oms.shorturl.OMSShortUrlOthers*

2. Write the implementation class that implements the *OMSUrlShortner* interface and overrides the following methods as demonstrated below:

```
public class OMSShortUrlOthers implements OMSUrlShortner {
    @Override
    public String shortenUrl(String engine, String longUrl, String
urlShortCred) {
        String shortenedUrl = null;
        try {
            /*
             * Implement the code for shortening URL here.
             * Example:
             * shortenedUrl = YourShorteningAlgorithm.shorten(longUrl);
             */
        } catch (Exception ex) {
            // Handle any exceptions that might occur during URL shortening
            ex.printStackTrace(); // Consider logging the exception instead of
printing
        }
    }
}
```

```

    }
    return shortenedUrl;
}
@Override
public boolean isValidCred(String engine, String urlShortCred) {
    try {
        /*
         * Implement the code for validation of the URL shortening
credentials here.
         * Example:
         * boolean isValid =
YourValidationLogic.isValidCredentials(engine, urlShortCred);
         * return isValid;
         */
    } catch (Exception ex) {
        // Handle any exceptions that might occur during credential
validation
        ex.printStackTrace(); // Consider logging the exception instead of
printing
        return false;
    }
    return true;
}
}

```

3. Place the jar file into the library folder of your application server.
If you are using the JBoss application server, ensure to add the jar entry in the *module.xml* file as follows: `<resource-root path="{jarName}.jar"/>`.
4. Restart the application server to apply the changes.

For further information on enabling URL shortening, refer to the *NewgenONE OmniOMS 12.0 Admin Workspace User Guide*.

Integrating Facebook and Twitter channels

For creating a Facebook and Twitter channel in the Admin Console user needs to generate AccessToken using NewgenONESocialConnector application.

getAccessToken

The *getAccessToken* API allows you to fetch the AccessToken used to create Facebook or Twitter channel.

- **Service URL:** *https://IP:<PORT>/NewgenONESocialConnector/getAccessToken*
- **Service Methods:** NewgenONESocialConnector
- **Method Type:** POST
- **Consumes:** application/json
- **Produces:** application/json

Sample input JSON

Case 1 : Facebook

```
{
  "SocialMediaName": "facebook",
  "AccessToken": "YOUR_ACCESS_TOKEN",
  "PageId": "YOUR_PAGE_ID"
}
```

Case 2 : Twitter

```
{
  "SocialMediaName": "twitter",
  "ConsumerKey": "YOUR_CONSUMER_KEY",
  "ConsumerSC": "YOUR_CONSUMER_SECRET",
  "AccessToken": "YOUR_ACCESS_TOKEN",
  "AccessTokenSecret": "YOUR_ACCESS_TOKEN_SECRET"
}
```

Input parameters description

Name	Mandatory (Yes or No)	Description	Valid values	Default value
SocialMediaName	Yes	Name of the Social Media Type.	"facebook"/"twitter"	NA
AccessToken	Yes	Access token fetched from Facebook or Twitter Developer Account.	Valid Value for Access Token	NA
PageId	No	Mandatory for Facebook & fetched from Facebook Developer Account.	Valid Value for PageId	NA
ConsumerKey	No	Mandatory for Twitter & fetched from Twitter Developer Account.	Valid Value for ConsumerKey	NA
ConsumerSC	No	Mandatory for Twitter & fetched from Twitter Developer Account.	Valid Value for ConsumerSecret	NA
AccessTokenSecret	No	Mandatory for Twitter & fetched from Twitter Developer Account.	Valid Value for AccessTokenSecret	NA

Sample output JSON

```
{
  "Status": "success",
  "Access-Token": "YOUR_ACCESS_TOKEN"
}
```

Output parameters description

Name	Description
Status	Success for successfully generated token
Access-Token	Token generated for Facebook or Twitter

Error code description

Code	Message	Comment
"failure"	Invalid input for socialMedia.	Use the Appropriate social media name.

Configuring message and archival jobs using REST services

This section describes the configuration of the following types of jobs in the Composition Designer (CDG) using REST Web Services:

- [Message Job](#)
- [Archival Job](#)

Configure Message jobs

You can configure Message Jobs in CDG in the following ways :

- WSDL Configuration
- REST Service Configuration

This section explains the configuration of the Message Job using the REST Service Configuration option. For information on creating a Message Job using WSDL Configuration, refer to the *NewgenONE OmniOMS 12.0 Composition Designer User Guide*.

To configure Message Job in CDG using the REST services, follow the below steps:

1. Create REST Web Services for Custom Message implementation that accepts input in XML or JSON format and hosts the Web Service on a server.
Ensure that the Web Service returns the same output as shown in the [Sample Message Tester](#).
2. Create a Message Job using the REST Service Configuration option in CDG:
 - a. In the Message Job dialog, go to the **Message** tab.
 - b. Select **REST Service Configuration**.
 - c. In the **REST URL**, specify the URL of the other DMS service.
 - d. Click **Test** to test the connection to the URL.
 - e. From the **Media Type** dropdown, select an appropriate option for the output: **XML** or **JSON**.

- f. In the **Forward Mapping** field, specify the field as defined in the message tester.
- g. Select the required fields to map against this field and click **Add**.
Similarly, map all required fields.
- h. Click **OK** to save the message job configuration.

Sample Message tester

```
@Path("/RestMessageTester/")
@Produces("application/xml")
public class RestMessageTester {
    @POST
    @Path("/MessageTester")
    public String processMessageTester(String input) {
        String mobileNumber = null;
        String text = null;
        try {
            XMLParser parser = new XMLParser(input);
            mobileNumber = parser.getValueOf("mobileNumber");
            text = parser.getValueOf("text");
            System.out.println(decode(text));
            /*
             * Write your custom logic here
             */
        } catch (Exception e) {
            e.printStackTrace();
        }
        return "<Response><Attributes><return><SMSWSResponse><Status>0</Status><MessageId>67567576</MessageId><Response>Response with phone number and message<MobileNumber>" + mobileNumber + "</MobileNumber><Text>" + text + "</Text></Response></SMSWSResponse></return></Attributes></Response>";
    }

    @POST
    @Path("/MessageTesterJSON")
    public String processMessageTesterJSON(String input) {
        String mobileNumber = null;
        String text = null;
        String json = null;
        try {
            JSONObject inputJsonObj = new JSONObject(input);
            mobileNumber = inputJsonObj.get("mobileNumber").toString();
            text = inputJsonObj.get("text").toString();
            System.out.println(decode(text));
        }
    }
}
```

```

* Write your custom logic here
*
*/
String retOutput = "<Response><Attributes><return><SMSWSResponse><Status>0</Status><MessageId>67567576</MessageId><Response>Response with phone number and message<MobileNumber>" + mobileNumber + "</MobileNumber><Text>" + text + "</Text></Response></SMSWSResponse></return></Attributes></Response>";
JSONObject jsonObj = XML.toJSONObject(retOutput);
json = jsonObj.toString(0);
} catch (Exception e) {
e.printStackTrace();
}
return json;
}
public String decode(String text) throws UnsupportedEncodingException {
byte[] decoded = DatatypeConverter.parseBase64Binary(text);
Arrays.toString(decoded);
String decodedString = new String(decoded);
return decodedString;
}
}

```

Configure Archival jobs

You can configure Archival Jobs in CDG in the following ways :

- WSDL Configuration
- REST Service Configuration

This section explains the configuration of Archival Job using the REST Service Configuration option for other DMS. For information on creating an Archival Job using WSDL Configuration, refer to the *NewgenONE OmniOMS 12.0 Composition Designer User Guide*.

To configure Archival Job in CDG for other DMS using the REST services, follow the below steps:

1. Create REST Web Services for Custom Archival implementation that accepts input in XML or JSON format and hosts the Web Service on a server.
Ensure that the Web Service returns the same output as shown in the [Sample Archival Tester](#).
2. Create an Archival Job using the REST Service Configuration option in CDG:
 - a. For the required type of job, go to the **Archival** tab.
 - b. From the DMS dropdown, select **Other DMS**.
 - c. Select **REST Service Configuration**.
 - d. In the **REST URL**, specify the URL of the other DMS service.
 - e. Click **Test** to test the connection to the URL.
 - f. From the **Media Type** dropdown, select an appropriate option for the output: **XML** or **JSON**.
 - g. In the **Forward Mapping** field, specify the field as defined in the archival tester.
 - h. Select the required fields to map against this field and click **Add**.
Similarly, map all required fields.
 - i. Click **OK** to save the archival job configuration.

Sample Archival tester

```
@SuppressWarnings("static-access")
@Path("/RestArchivalTester/")
@Produces("application/xml")
public class RestArchivalTester {
    @POST
    @Path("/ArchivalTester")
    public String processArchivalTester(String input) {
        String outputXML = null;
        try {
            /*
             * write yor custom logic here
             */
            outputXML = "<Status>0</Status>";
        } catch (Exception e) {
            e.printStackTrace();
        }
        return outputXML;
    }
}
```

```
}
@POST
@Path("/ArchivalTesterJSON")
public String processArchivalTesterJSON(String input) {
    String json = null;
    try {
        /*
         * Write your custom logic here
         */
        json = "{\"Status\":\"0\"}";
    } catch (Exception e) {
        e.printStackTrace();
    }
    return json;
}
```

Configuring cobranding in OmniOMS

You can configure co-branding in OmniOMS web modules to customize the appearance and create a personalized environment that reflects the branding of your organization. Co-branding allows you to apply your own logo to the following modules:

- OmniOMS Admin Workspace (OmniOMS Admin Console)
- OmniOMS Communication Editor (OmniOMS Interactive Editor)
- OmniOMS Service Workspace (OmniOMS Service Manager)

To configure co-branding in OmniOMS web modules, follow the below steps:

1. Open the *OMSAdminConfig.ini* file located at *C:\JBoss\JBoss_Build_XXX\jboss-eap-X.0.0\jboss-eap-X.0\Newgen\NGConfig\OMSConfig (JBoss)* and set the value of the *CoBranding* parameter to *Y*.
2. Create a new folder named **OMSCustomConfig** within the *OMSConfig* folder.
3. Create the following subfolders within the *OMSCustomConfig* folder:
 - OMSAdminConsole
 - OMSInteractiveEditor
 - OMSServices
4. Create a folder named **Images** within each of the subfolders created in the previous step.
5. Place the required brand images in SVG format inside the Images folder. Name the images as follows:
 - **HeaderImage** — The image gets placed on the header of the module.
 - **LoginImage** — The image gets placed on the login screen of the module.
The logo appears on the login page and header of the module.

Along with the tag “Powered by Newgen”, the logo images appear on the login page and header of the module, providing users with a customized and branded experience.



- Create folders and file images using the exact names specified in this guide.
- Images for co-branding are only supported in SVG format.
- The login image is not applicable to the OmniOMS Interactive Editor (OmniOMS Communication Editor).
- For your branding logo to appear on the login page and header of the NewgenONE OmniOMS modules, the following conditions must be met:
- The value of the parameter *CoBranding* in the *OMSAdminConfig.ini* file must be set to *Y*.
 - The images intended for branding must be correctly named and placed in the specified folder locations.
 - In the case of a folder containing no images, the system displays Newgen's logo.

Adding play button image in email template

To add the play button image to the videos added in the Email templates created using the OmniOMS Communication Designer, follow the below steps:

1. Copy the play button images present at the following location in the application server directory: *bin/OMSConfig/EmaillImages*
Alternatively, you can copy any required image for the play button.
2. Host the play button image on an image hosting service such as Dropbox, Google Photos, and more.
3. Copy the URL of the hosted play button image from the hosting service.



Ensure that the image is made easily available and reachable to the general public, such that individuals can access it without any barriers or limitations. This means taking steps to ensure that the image can be viewed by anyone, regardless of their location, device, or any specific requirements they may have.

4. While configuring video properties for an email template, locate the **Play Button URL** field and paste the copied image URL in it.



For every video control added to an email template, it is mandatory to provide the play button URL separately.

For more information on adding video control and configuring its properties, refer to the *Video* section in the *NewgenONE OmniOMS 12.0 Communication Designer User Guide*.