NewgenONE

Workitem Microservices

Developer Guide

Version: 2024.2

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.

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

This chapter provides information about the purpose of this guide, details of the intended audience, revision history, and related documentation for NewgenONE Microservices.

Revision history

Revision date	Description
October 2024	Initial publication

About this document

This developer guide explains the development and integration of NewgenONE Microservices. This guide outlines the Microservices necessary for implementing the workitem and data objects created in the data model and process designer. Also, it provides more flexibility and the ability to run the services independently.

This guide assumes you have a working knowledge of the Microsoft SQL, PostgreSQL, and Oracle database server.

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 Newgen partner.

Intended audience

This guide is intended for the developers from internal or external product implementation teams responsible for integrating Application Programming Interface (APIs) exposed as NewgenONE Microservices with custom Portal Applications created using the NewgenONE Process Designer. The reader must be comfortable with API Signature formats in JSON to understand API requests and responses.

Related documents

The following documents are related to NewgenONE Microservices:

- NewgenONE 2024.2 Configuration and Deployment Guide for AWS
- NewgenONE 2024.2 Newgen Enterprise Products Containerization for AWS
- NewgenONE 2024.2 Developer 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

Introduction to NewgenONE Microservices

Using the exposed APIs in NewgenONE Microservices, Portal applications perform the following operations:

- Workitem operation-related APIs are as follows:
  - Creating a new workitem
  - Unlocking a workitem
  - Opening existing workitem
  - Completing a workitem
  - Saving and fetching data of workitems
  - Fetching workitems from the queue
  - Search operation in complex array while fetching data of workitems
  - Searching workitems based on various filters

You can either create a portal application from Process Designer in the existing NewgenONE installation or can implement a custom portal application and call the available NewgenONE Workitem Microservices to derive the required results.

The exposed Microservices use OmniDocs Microservices for the creation and validation of user session, folder, and document management system operations.

Workitem service

This chapter includes the following APIs:

WFUploadWorkitem

The *WFUploadWorkitem* API allows you to create workitems, and their corresponding DMS folders in the underlying DMS. It also allows you to set the workitem attribute variables, upload documents in the workitem folders and initiate workitems in a single call. For more details, you can refer to the JSON request and response parameters provided in the next sub-sections.

- Service URL— *http://<GatewayIP:GatewayPort >/workitem-service-context/workitem/uploadworkitem*
  Where, GatewayIp is the IP of the machine, and GatewayPort is the port of the machine where the workitem service is deployed.
- Request Media Type — application/json
- Response Media Type — application/json
- Method Type — POST

Header parameters description

The following table describes the header parameters of the *WFUploadWorkitem* API, if OAuth is disabled:

Name	Parameter Scope	Type	Description	Mandatory
Authorization	HeaderParam	String	The session ID of the signed-in user.	Yes
Accept-Language	HeaderParam	Locale	Represents the specific region or language. Example, en-US	No
Tenant-Id	HeaderParam	String	The name of the cabinet.	Yes

If OAuth is enabled:

Name	Parameter scope	Type	Description	Mandatory
AppId	HeaderParam	String	Registered AppId of the product	No
Access Token	HeaderParam	String	AccessToken for the signed-in user	No
SessionValid	HeaderParam	String	SessionValid for the signed-in user	No

Request body parameter description

The following table describes the request body parameters of the *WFUploadWorkitem* API:

Name	Parent Tag	Type	Description	Valid Values	Default Values	Mandatory
ProcessDefId	root	Integer	Process ID to create or upload a workitem.		NA	Yes

Name	Parent Tag	Type	Description	Valid Values	Default Values	Mandatory
QueueId	root	Integer	A queue ID that requires workitem introduction.		NA	No
InitiateFrom ActivityId	root	Integer	Refers to the work-introduction workstep ID of the specified process in which the workitem is initiated in the process. After omitting this tag, the system initiates the workitem from the default work-introduction workstep for the process.		NA	No
InitiateAlso	root	String	Indicates whether workitem initiation is required or not.	Y-Yes N-No	N	No
Attribute	root	Map<String, Object>	Set the attribute parameter value such as queue, external and complex variables.		NA	No
Documents	root	List<WF Document Request>	Refers to the documents list present on the image server and you need to add them to the workitem.		NA	No
Case 1. Below inputs are required to attach the existing document in the workitem. The Document-Name-ID list for all the documents present to the Image Server and add them to the workitem in case documents are added using the reference of an existing document in DMS.						
Document TypeName	Documents	String	The Name of the Document.		NA	Yes
Document Index	Documents	String	Specify the Document Index.		NA	Yes

Name	Parent Tag	Type	Description	Valid Values	Default Values	Mandatory
ParentFolder Index	Documents	String	Specify the Parent Folder Index of the document.		NA	Yes
Case 2: Below inputs are required to attach the new document in workitem in case document addition is required that already exists on the image server so the ISIndex is available for the document.						
AccessType	Documents	String	Specify the scope to access the document.	S-Shared P-Private I-Inherited	S	No
Document Name	Documents	String	The name of the document type created in NewgenONE process.	N/A	NA	Yes
Comment	Documents	String	The name of the actual document. For example, if the actual document name is AadharCard2022.png then comment is AadharCard2022.)	N/A	NA	No
Document Type	Documents	String	Specify the document type.	I-Image N-Non-Image A-Reserved	N	No
CreatedBy AppName	Documents	String	The document extension. For Example: JPG, GIF, PNG, and so on.		NA	Yes

Name	Parent Tag	Type	Description	Valid Values	Default Values	Mandatory
ISIndex	Documents	String	Specify a “#” separated unique combination of Image Index and Volume Index for the document in the folder. This is used to upload a document from the Image Server to the workitem.		NA	Yes
EnableLog	Documents	String	Indicates when the logging ID is complete or not.	Y-Logging is complete N-Logging is incomplete	NA	No
Document Size	Documents	Long	Document size in bytes.		NA	Yes

Response body parameters description

The following table describes the response body parameters of the *WFUploadWorkitem* API:

Name	Parent Tag	Parameter Type	Description
ProcessInstanceId	Root	String	The process instance ID is generated corresponding to the workitem.
URN	Root	String	URN generated corresponding to the workitem.
WorkStageId	Root	Integer	Workstage to introduce the workitem.
CreationDateTime	Root	Date	Date-Time when the process instance is created in the system.
FolderIndex	Root	Integer	The ID of the DMS Folder is created for the Process Instance.
Documents	Root	List<WFDocument>	Document details list for each of the documents that are provided in the input of the WFUploadWorkitem API.
DocumentName	WFDocument	String	Contains the document name provided in the input JSON of API.
ISIndex	WFDocument	String	Contains the ISIndex of the image server in which the document is present.

Error codes and description

The following table describes the error codes that can occur while executing the *WFUploadWorkitem* API:

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_502	400	Bad Request	The process definition ID was not valid.	Pass a valid ProcessDefId value in the request body.
NGONE_WFS_4018	400	Bad Request	Input Queue is Non-Introduction Type or Invalid Queue	Pass the valid QueueID value in the request body.
N/A	401	Unauthorized	Invalid Session	Regenerate the session and pass the valid session Id in the authorization header.
NGONE_WFS_801	500	Internal Server Error	SQL Exception Occurred.	An SQL exception occurred which stopped the operation. Refer to the Error logs for further analysis.

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_3001	400	Bad Request	Invalid Input Parameters.	Refer to the detailed error message to correct input parameters which have invalid values. You can refer to the below sample for reference (FieldName: ErrorDescription), 1. ProcessDefId: ProcessDefId must contain value greater than equal to zero and positive integer. 2. QueueId: QueueId must have value greater than equal to zero and positive integer. 3. InitiateAlso: InitiateAlso value must be Y or N. There can be multiple scenarios for this error code. To resolve the issue, refer to the detailed error message.
NGONE_WFS_503	503	Temporary	Service is either down or slow.	Service is either unavailable or slow.
NGONE_WFS_830	401	Unauthorized	No rights	Either user has no rights on the current object (process).

Sample request and response

Below is the sample request and response body for *WFUploadWorkitem* API as per the following scenarios.

Uploading workitem with attributes

Request:


```
{
  "InitiateFromActivityId": "1",
  "InitiateAlso": "N",
  "Documents": [],
  "Attributes": {
    "InsuranceAmount": "4000000",
    "ApplicantType": "Adult",
    "Applicant": [
      {
        "HashId": "19",
        "ApplicantId": "4",
        "InsertionOrderId": "0",
        "ApplicantName": "Alex",
        "Location": [
          {
            "HashId": "20",
            "Pin": "13",
            "InsertionOrderId": "0",
            "City": "Seattle"
          },
          {
            "HashId": "21",
            "Pin": "31",
            "InsertionOrderId": "0",
            "City": "California"
          }
        ]
      },
      {
        "HashId": "22",
        "ApplicantId": "31",
        "InsertionOrderId": "0",
        "ApplicantName": "Phil",
        "Location": [
```

```
          {
            "HashId": "23",
            "Pin": "110017",
            "InsertionOrderId": "0",
            "City": "Seattle"
          },
          {
            "HashId": "24",
            "Pin": "232",
            "InsertionOrderId": "0",
            "City": "California"
          }
        ]
      }
    ],
    "CreatedDateTime": "2022-07-03 00:00:00"
  },
  "ProcessDefId": "4030",
  "QueueId": "4063"
}
```

Response:

```
{
  "ProcessInstanceId": "CarInsurance-0000000001210-Demo",
  "URN": "CarInsurance-1210",
  "WorkStageId": 1,
  "CreationDateTime": "2022-07-05 07:40:43",
  "FolderIndex": "8513",
  "Documents": []
}
```

Uploading workitem without any attributes or documents

Request:

```
{
  "InitiateFromActivityId": "1",
  "InitiateAlso": "N",
  "Documents": [],
  "Attributes": null,
  "ProcessDefId": "4030",
  "QueueId": "4063"
}
```

Response:


```
{
  "ProcessInstanceId": "CarInsurance-0000000001211-Demo",
  "URN": "CarInsurance-1211",
  "WorkStageId": 1,
  "CreationDateTime": "2022-07-05 10:37:46",
  "FolderIndex": "8514",
  "Documents": []
}
```


Uploading workitem with attributes and document

This section contains the request and response body to upload workitems with attributes and document present on the image server.

Request:


```
{
  "InitiateFromActivityId": "1",
  "InitiateAlso": "N",
  "Documents": [
    {
      "Comment": "aadhaar",
      "NoOfPages": "1",
      "DocumentSize": "10857",
      "ISIndex": "315#1#",
      "DocumentType": "I",
      "EnableLog": "Y",
      "CreatedByAppName": "jpg",
      "AccessType": "I",
      "DocumentName": "Aadhaar"
    },
    {
      "Comment": "pan",
      "NoOfPages": "1",
      "DocumentSize": "7568",
      "ISIndex": "316#1#",
      "DocumentType": "I",
      "EnableLog": "Y",
      "CreatedByAppName": "jpg",
      "AccessType": "I",
```

```
      "DocumentName": "PAN"
    }
  ],
  "Attributes": {
    "InsuranceAmount": "15000",
    "ApplicantType": "Minor",
    "Applicant": [
      {
        "HashId": "0",
        "ApplicantId": "12",
        "InsertionOrderId": "0",
        "ApplicantName": "Alex"
      },
      {
        "HashId": "1",
        "ApplicantId": "21",
        "InsertionOrderId": "0",
        "ApplicantName": "Phil",
        "Location": [
          {
            "HashId": "2",
            "Pin": "110017",
            "InsertionOrderId": "0",
            "City": "Seattle"
          },
          {
            "HashId": "3",
            "Pin": "110018",
            "InsertionOrderId": "0",
            "City": "California"
          }
        ]
      },
      {
        "HashId": "4",
        "ApplicantId": "12",
        "InsertionOrderId": "0",
        "ApplicantName": "Gloria",
        "Location": {
          "HashId": "5",
          "Pin": "110019",
          "InsertionOrderId": "0",
          "City": "Pheonix"
        }
      }
    ],
    "CreatedDateTime": "2022-07-02 00:00:00"
  },
  "ProcessDefId": "4030",
  "QueueId": "4063"
}
```

Response:

```
{
  "ProcessInstanceId": "CarInsurance-0000000001217-Demo",
  "URN": "CarInsurance-1217",
  "WorkStageId": 1,
  "CreationDateTime": "2022-07-05 11:32:53",
  "FolderIndex": "8519",
  "Documents": [
    {
      "DocumentName": "Aadhaar",
      "DocumentIndex": "5184",
      "ISIndex": "315#1#"
    },
    {
      "DocumentName": "PAN",
      "DocumentIndex": "5185",
      "ISIndex": "316#1#"
    }
  ]
}
```

WFGetWorkitemDataExt

The *WFGetWorkitemDataExt* allows you to lock the specified workitem and fetch complete data for a workitem at once. To lock and fetch the data of the workitem, you must specify the Process Instance ID and Workitem ID of that workitem.

- Service URL — *http://<GatewayIp:GatewayPort >/workitem-service-context/workitem/getworkitem*

  Where, GatewayIp is the IP of the machine, and GatewayPort is the port of the machine where the workitem service is deployed.
- Request Media Type — application/json
- Response Media Type — application/json
- Method Type — PUT

Header parameters description

The following table describes the header parameters description of the *WFGetWorkitemDataExt* API, if OAuth is disabled:

Name	Parameter Scope	Type	Description	Mandatory
Accept-Language	HeaderParam	Locale	Represents the natural language and locale that the client prefers. Example, en-US	No
Tenant-Id	HeaderParam	String	Name of the cabinet.	Yes
Authorization	HaderParam	String	The session ID of the signed-in user.	Yes

If OAuth is enabled:

Name	Parameter scope	Type	Description	Mandatory
AppId	HeaderParam	String	Registered AppId of the product	No
Access Token	HeaderParam	String	AccessToken for the signed-in user	No
SessionValid	HeaderParam	String	SessionValid for the signed-in user	No

Request body parameters description

The following table describes the error codes that can occur while executing the *WFGetWorkitemDataExt* API:

Name	Parent Tag	Type	Description	Default Values	Valid Values	Mandatory
ArchiveSearch	Root	String	Flag to enable the search archival.	N	Y- Yes N-No	No

Name	Parent Tag	Type	Description	Default Values	Valid Values	Mandatory
NoOfRecords ToFetch	Root	Integer	Represents the fetched number of records.	-1	NA	No
Process InstanceId	Root	String	Process instance ID corresponding to the workitem to lock and fetch its data. Specify this value to lock a particular workitem. Additionally, if the tag value is not specified, then specify the Queue ID and Queue Type Tags. User must pass either ProcessinstanceId and workitemid to open the workitem or pass LastProcessInstanceId and LastWorkitem to fetch the next eligible workitem in Previous-Next functionality or bring next workitem functionality.	NA	NA	No
WorkItemId	Root	Integer	Workitem ID to lock and fetch its data. Specify this value for locking a particular workitem.	0	NA	No

Name	Parent Tag	Type	Description	Default Values	Valid Values	Mandatory
QueueID	Root	Integer	A queue ID from which the first available workitem must be fetched. This tag must be specified for fetching the first unlocked workitem from a specific queue along with its complete data and locking the workitem. This tag can be ignored if the Process instance ID and Workitem ID are specified.	-1	NA	No
Queue Type	Root	String	It is a type of queue from the above specified QueueId.	NA	• I -Introduction • F - FIFO • D - (WIP) Dynamic • N - (WIP) No Assignment • S - (WIP) Permanent WIP stands for work in progress queues. Also, WIP is categorized into three queue type such as Dynamic, No Assignment and Permanent as mentioned above.	No

Name	Parent Tag	Type	Description	Default Values	Valid Values	Mandatory
Activity Type	Root	Integer	The type of activity.	0	An activity type can be: 1 -Work Introduction 2 - Work Exit 3 - Discard 4 - Hold 5 - Distribute 6 - Collect 7 - Decision 18 - SubProcess 10 - Custom 11:Query 19: Integration	No

Name	Parent Tag	Type	Description	Default Values	Valid Values	Mandatory
DocOrder By	Root	String	The field name to order documents in the workitem.	NA	1–Index 2–Object Name 3–Owner 5–Revised DateTime 9–CreatedBy AppName 10–NoOfPages 11–DocumentSize 18–DocOrderNo There is also a provision to pass column name on which order by is to be applied. Allowed column names are: • Index • Object • Name • Owner • Revised • DateTime • CratedBy • AppName • NoOfPages • Document • Size • DocOrderNo	No
DocSort Order	Root	String	The sorting order of the documents in the workitem.	NA	A: Ascending D: Descending	No
DataAlso Flag	Root	String	This tag returns a data class associated with the document.	N	Y - Yes N-No	No

Name	Parent Tag	Type	Description	Default Values	Valid Values	Mandatory
Object Preference List	Root	String	The list of Object Types for fetching the preferences of the signed-in user.	NA	Comma-separated Object Types. For example: W, D.	No
Generate Log	Root	String	The flag indicates whether to generate logs for locking and fetching the complete data of the workitem.	NA	Y - Yes to generate logs N - No to not to generate logs.	No
Last WorkItemId	Root	Integer	The workitem ID of the previous workitem.	NA	NA	No
DataExtFlag	Root	String	The flag whether to show additional details of the workitem.	N	Y - Yes N - No	No
LastWorkItem	Root	Integer	The ID of the last workitem for fetching the next workitem.	0	N/A	No
LastProcess Instance	Root	String	The process instance ID of the last workitem to fetch the next workitem.	NA	NA	No
OrderBy	Root	Integer	Order By field for sorting the activities list.	2	1 - ActivityID 2 - Activity Name 3 - Number of Workitems 4 - Number of Delayed Workitems 5 - Process ID	No
SortOrder	Root	String	Indicates whether to sort the fetched workitems in an ascending or descending order.	A	A - Ascending D - Descending	No

Name	Parent Tag	Type	Description	Default Values	Valid Values	Mandatory
LastValue	Root	String	The name of the last activity that is retrieved in the previous batch.	NA	NA	No
ClientOrder Flag	Root	String	Indicates whether the client has clicked on some field or Alias for sorting.	N	Y - Yes N - No	No
PDAFlag	Root	String	Indicates whether the call is coming from a personal digital assistant like a mobile.	N	Y - Mobile N - Web desktop	No
AssignMe	Root	String	Indicates whether the workitem must be reassigned to the signed-in user or not.	N	Y - Yes N - No	No
ExternalTable Name	Root	String	The name of the external table for fetching the alias values defined on external variables.	NA	NA	No

Response body parameters description

The following table describes the response body parameters of the *WFGetWorkitemDataExt* API:

Name	Parent Tag	Parameter Type	Description
ActivityType	Root	Integer	The type of activity. It can be of the following types: • 1 — Work Introduction • 2 — Work Exit • 3 — Discard • 4 — Hold • 5 — Distribute • 6 — Collect • 7 — Decision • 10 — Custom • 11 — Query • 18 — SubProcess • 19 — Integration
ProcessName	Root	String	The name of the process.
ProcessDefId	Root	Integer	A process ID to enable its variable value setting action.
WorkItemStateId	Root	Integer	The state ID of the workitem.
VariantId	Root	Integer	The variant ID for process variant support.
CurrentDateTime	Root	Date	The current date and time of the server.
Process InstanceId	Root	String	The process instance ID of the workitem is returned.
WorkItemId	Root	Integer	The workitem ID of the workitem is returned.
ProcessVersion	Root	Integer	A process version that owns the fetched workitem.

Name	Parent Tag	Parameter Type	Description
ProcessedBy	Root	String	A user Id that processed the workitem.
ActivityName	Root	String	Activity name on which the workitem currently depends.
ActivityId	Root	Integer	Activity ID on which the workitem currently depends.
EntryDateTime	Root	Date	Date and time of the workitem entering the activity on which it currently exists.
AssignmentType	Root	String	The current assignment type for the workitem. Types are as follows: • S — Shared • F — Fixed
PriorityLevel	Root	Integer	The priority level of the fteched workitem. The priority level are as follows: • 1 — Low • 2 — Medium • 3 — High • 4 — Very high
ValidTill	Root	Date	The date and time by when the workitem are valid on the activity.
QueueId	Root	Integer	The ID of the queue that owns the workitem.
AssignedTo	Root	String	The name of the user for the assigned workitem.
WorkItemState	Root	String	Following states of the fetched workitem: • 1 — Not started • 2 — Running • 3 — Suspended • 4 — Terminated • 5 — Aborted • 6 — Completed

Name	Parent Tag	Parameter Type	Description
LockStatus	Root	String	Following is the Lock status of the fetched workitem: • Yes — Locked • No — Unlocked
LockedTime	Root	Date	The date and time at which the user locked the workitem.
QueueName	Root	String	The name of the queue to which the workitem belongs.
QueueType	Root	String	The type of the queue in which the workitem currently lies that are as follows: • I - Introduction • F - FIFO • D - (WIP) Dynamic • N - (WIP) No Assignment • S - (WIP) Permanent WIP stands for work in progress.
Introduction DateTime	Root	Date	The date and time when the workitem is introduced or started in the process.
ReferredBy	Root	Integer	User Name for the referred workitem.
CheckList CompleteFlag	Root	String	Indicates all the following ToDo items defined in the process for the workitem are completed or not. • Y - Complete • N - Incomplete
InstrumentStatus	Root	String	The exception status of the fetched workitem is as follows: • E - In exception • N - No exception raised
VariantId	Root	Integer	The variant ID to fetch the variable list.

Name	Parent Tag	Parameter Type	Description
URN	Root	String	Name of workitem as shown in UI
CacheTime	Root	Date	Indicates the date and time when the definition for external interfaces like ToDo, Exception, and Documents is cached.
LastModifiedTime	Root	Date	Last modified time of workitem.
IntroducedBy	Root	String	User name for the introduced workitem.
WorkitemData	Root	WFWorkitem DataResponse	Data of the workitem.
ExternalData	WorkItemData	ExternalData	Contains the user's input on the external interfaces associated with the WorkItem like To-Dos and Exceptions.
ToDoListInterface	ExternalData	List<WFToDo ListStat>	Contains the user's responses for To-Do items.
ToDoIndex	ToDoListInterface	Integer	Index of the To-Do item.
ToDoHistory	ToDoListInterface	String	User specified value for the To-Do item.
ActionDateTime	ToDoHistory	LocalDateTime	Date and time of the updated value for the given To-Do item.
ToDoRemark	ToDoHistory	String	User comment on the given To-Do item.
ActivityName	ToDoHistory	String	Activity name of the updated comment or value for the To-Do.
UserName	ToDoHistory	String	The user name who comments or specifies the value for the given To-Do.
PersonalName	ToDoHistory	String	Personal name of the user who comments or specifies the value for the given To-Do.

Name	Parent Tag	Parameter Type	Description
FamilyName	ToDoHistory	String	Family name of the user who comments or specified the value for the given To-Do.
ExceptionInterface	ExternalData	List<WF ExceptionStat>	Contains the user's responses for exception items.
ExceptionDefIndex	ExceptionInterface	Integer	Exception ID associated with the activity on which the workitem currently exist.
ActionIndex	ExceptionHistory	Integer	ID of the raising or clearing action on the exception.
ExcpSeqId	ExceptionHistory	Integer	It's a exception sequence Id.
ActivityName	ExceptionHistory	String	Activity for the raised or cleared exception.
UserName	ExceptionHistory	String	Name of the User who raised or cleared the exception.
ActionDateTime	ExceptionHistory	LocalDateTime	Date-Time for the raised or cleared exception.
ExceptionName	ExceptionHistory	String	Name of the exception.
CommentData	WorkItemData	List<WF CommentInfo>	Comments for the referred, reassigned, or rejected workitems during audit.
Comment	CommentData	String	Comments specified for the workitem.
CommentBy	CommentData	Integer	ID of the user who specified the comments.
CommentByName	CommentData	String	Name of the user who specified the comments corresponding to the ID.
CommentBy PersonalName	CommentData	Integer	Name of the user ID who specified the comment.
CommentTo	CommentData	String	ID of the user who received the comment.

Name	Parent Tag	Parameter Type	Description
CommentToName	CommentData	String	Name of the user ID who received the comment.
CommentTo PersonalName	CommentData	String	Name of the user ID who received the comment.
CommentType	CommentData	Integer	Type of the comments having one of the following three values: • 1 - Indicates comments for the referred workitems • 2 - Indicates comments for the reassigned workitems • 3 - Indicates comments provided during the audit rejection
Comment DateTime	CommentData	LocalDate Time	Indicates the date and time of the published comment.
ActivityName	CommentData	String	Name of the activity workItem when the user published the comment.
AccessDateTime	Documents	Date	Access Date and Time of the document
Comments	Documents	String	Any comment associated with the document.
CreationDateTime	Documents	Date	Date and Time of creation of the document.
DocumentLock	Documents	String	Indicates whether any user has locked the document or not. • Y – Yes • N – No
FamilyName	OwnerInfo	String	The Family name of the author.
PersonalName	OwnerInfo	String	The personal name of the owner of the document.
Name	OwnerInfo	String	Name of the owner of the document.

Name	Parent Tag	Parameter Type	Description
Type	OwnerInfo	String	The tag represents the type of owner.
Index	OwnerInfo	Integer	User Index of the owner.
Checkout ByUserInfo	Documents	PersonInfo	User name of the checked out document.
FtsFlag	Documents	String	FTS flag of a document.
ExpiryDateTime	Documents	Date	Expiry date and time of the document.
Location	Documents	String	location of the document.
PullPrintFlag	Documents	String	Pull the Print flag associated with the document. Its valid values are as follows: • Y – Pull print generated • P – Pending • F – Failed • I – Insoview generated
ThumbNailFlag	Documents	String	Specifies when Thumbnail is generated for the document.
DocStatus	Documents	String	Status of the document.
ParentFolderIndex	Documents	Integer	Folder Index of the document where the document resides.
DocOrderNo	Documents	Integer	The order in which the document is added to the folder.
FiledDateTime	Documents	Date	Date and Time when the document is filed in its Parent Folder.
ReferenceFlag	Documents	String	Reference flag of the document.
OwnerType	Documents	String	Represents the type of owner.
LoginUserRights	Documents	String	Rights string of the signed-in user.
FiledByUser	Documents	String	Username for the filed document.

Name	Parent Tag	Parameter Type	Description
IsSecureFlag	Documents	String	Indicates that the folder is protected by IRM Policy.
AnnotationFlag	Documents	String	Flag that specifies whether annotations are present in the document.
LinkDocFlag	Documents	String	Flag that specifies whether the document is present with the linked document or not.
Enablelog	Documents	String	Specifies whether the log is generated for the added document or not. • Y – Log is generated • N – Log not generated
ActualDocument Size	Documents	Integer	Size of the document.
OwnerIndex	Documents	Integer	Index of the document owner.
Owner	Documents	String	Name of the document owner.
RevisedDateTime	Documents	Date	Revised date and time of the document.
Document VersionNo	Documents	String	The version number of the document.
DocumentName	Documents	String	The name of the document.
CreatedByApp Name	Documents	String	The name of the application through which the document gets created in the DMS.
IsIndex	Documents	String	Image server index for retrieving the document image.
DocumentType	Documents	String	Document type with which the document is added in the workitem.
CheckoutStatus	Documents	String	Indicates whether the document is currently checked out or not.

Name	Parent Tag	Parameter Type	Description
NoOfPages	Documents	String	The number of pages in the document.
DocumentSize	Documents	String	The size of the document.
DocumentIndex	Documents	String	The DMS ID of the document is included in the workitem.
DataDefIndex	DataDefinition	Integer	Index of the Data Class associated with the document.
Attribute	WorkItemResponse	Map	Map containing the key-value pair of all the attributes (name versus definition) of the workitem.
Name	Attribute	String	The name of the workitem attribute variable is associated with the activity on which the workitem currently lies.
Value	Attribute	String	The current value of the attribute variable.

Name	Parent Tag	Parameter Type	Description
Type	Attribute	String	Indicates the type of the data attribute. Also, it's a numeric value where each digit corresponds to some information about the attribute. The first digit indicates the type of the variable whether it is System, Queue, or External Table. • 1 - Queue • 2 - External • 3 - System The second digit indicates the access permissions on the attribute on the current activity. • 1 - Read • 2 - Modify • 3 - Both Read and Modify The third and fourth digits indicate the data type of the variable. • 3 - Integer • 4 - Long • 6 - Float • 8 - Date • 10 - Text
Length	Attribute	Integer	The length of the value of the attribute.

Error codes and description

The following table describes the error codes that can occur while executing the *WFGetWorkitemDataExt* API:

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_300	404	Not Found	No authorization.	Ensure that the user is associated with the queue.
NGONE_WFS_16	200	Success	The workitem is locked by another user.	Ensure that the workitem is present on the queue along with the user is associated and it is not locked by another user.
NGONE_WFS_400	404	Not Found	Operation Failed.	Recheck the request body.
NGONE_WFS_6	404	Not Found	The Workitem (ProcessinstanceId or WorkItemId) is invalid.	ProcessinstanceId or WorkItemId value passed in request body is incorrect.
NGONE_WFS_31	404	Not Found	Workitem expired.	Expired Workitem is pending for processing in expiry utility.
NGONE_WFS_810	404	Not Found	Invalid queue.	QueueId value passed in the request body is incorrect.
N/A	401	Unauthorized	Invalid session.	Regenerate the session and pass the valid session Id in the authorization header.
NGONE_WFS_18	404	Not Found	No More Records.	No more workitems are present in the requested queue.
NGONE_WFS_801	500	Internal Server Error	SQL Exception Occurred.	An SQL exception occurred which stopped the operation. Refer to the Error logs for further analysis.

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_3001	400	Bad Request	Invalid Input Parameters.	Refer to the detailed error message to correct input parameters which have invalid values. You can refer to the below sample for reference (FieldName: ErrorDescription), 1. DocSortOrder: DocSortOrder value must be A or D. 2. DataExtFlag: DataExtFlag value must be Y or N. 3. ProcessDefId: ProcessDefId must have value greater than equal to zero and positive integer. There can be multiple scenarios for this error code. To resolve the issue, refer to the detailed error message.
NGONE_WFS_503	503	Temporary	Service is either down or slow.	Service is either unavailable or slow.
NGONE_WFS_856	412	Precondition Failed	Archival cabinet is not set.	Verify whether archival cabinet is present and configured correctly.
NGONE_WFS_957	404	Not Found	WorkItem has already been processed.	The workitem user requests to open and does not exist on the archival cabinet.

Sample request and response

Request:

```
{
  "ArchiveSearch": "N",
  "ProcessInstanceId": "AZ-0000008583-process",
  "WorkItemId": 1,
  "QueueId": 10,
  "QueueType": "N",
  "ActivityType": 1,
  "DocOrderBy": 5,
  "DocSortOrder": "A",
  "ObjectPreferenceList": "W,D",
  "GenerateLog": "Y",
```

```
  "ZipBuffer": "N",
  "UserDefVarFlag": "Y",
  "DataExtFlag": "Y"
}
```

Response:

```
{
  "MainCode": 0,
  "CurrentDateTime": "2022-08-17 14:32:53",
  "ProcessInstanceId": "AZ-0000008583-process",
  "WorkItemId": 1,
  "ProcessName": "portal_process",
  "ProcessVersion": 1,
  "ProcessDefId": 2,
  "ProcessedBy": "ritika",
  "ActivityName": "Start Event_1",
  "ActivityId": 1,
  "EntryDateTime": "2022-08-12 07:25:36",
  "AssignmentType": "S",
  "PriorityLevel": 1,
  "QueueId": 10,
  "AssignedTo": "ritika",
  "WorkItemStateId": 2,
  "WorkItemState": "Running",
  "LockedTime": "2022-08-17T14:32:32.820+00:00",
  "QueueName": "portal_process_Start Event_1",
  "QueueType": "I",
  "VariantId": 0,
  "CheckListCompleteFlag": "N",
  "ActivityType": 1,
  "WorkItemResponse": {
    "documentResponse": [],
    "externalData": {
      "exceptionInterface": []
    },
    "commentData": [],
    "retrievedCount": 0,
    "count": 0,
    "Attribute": {
      "ApplicantAge": {
        "Length": 2,
        "Type": "133",
        "VarFieldId": 0,
        "VariableId": 1,
        "IsView": "N",
        "Value": "17",
        "Name": "ApplicantAge"
      },
      "ApplicantName": {
        "Length": 255,
```

```
        "Type": "1310",
        "VarFieldId": 0,
        "VariableId": 19,
        "IsView": "N",
        "Value": "John",
        "Name": "ApplicantName"
      },
      "itemindex": {
        "Length": 255,
        "Type": "3210",
        "VarFieldId": 0,
        "VariableId": 53,
        "IsView": "N",
        "Value": "11954",
        "Name": "itemindex"
      },
      "itemtype": {
        "Length": 255,
        "Type": "3210",
        "VarFieldId": 0,
        "VariableId": 54,
        "IsView": "N",
        "Value": "F",
        "Name": "itemtype"
      },
      "VAR_REC_3": {
        "Length": 255,
        "Type": "3210",
        "VarFieldId": 0,
        "VariableId": 55,
        "IsView": "N",
        "Value": "null",
        "Name": "VAR_REC_3"
      },
      "EventName": {
        "Length": 255,
        "Type": "3210",
        "VarFieldId": 0,
        "VariableId": 56,
        "IsView": "N",
        "Value": "null",
        "Name": "EventName"
      },
      "RoutingCount": {
        "Length": 255,
        "Type": "3210",
        "VarFieldId": 0,
        "VariableId": 57,
        "IsView": "N",
        "Value": "0",
        "Name": "RoutingCount"
```

```
    },
    "HoldStatus": {
      "Length": 255,
      "Type": "323",
      "VarFieldId": 0,
      "VariableId": 39,
      "IsView": "N",
      "Value": "null",
      "Name": "HoldStatus"
    },
    "CheckListCompleteFlag": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 50,
      "IsView": "N",
      "Value": "N",
      "Name": "CheckListCompleteFlag"
    },
    "InstrumentStatus": {
      "Length": 255,
      "Type": "3110",
      "VarFieldId": 0,
      "VariableId": 37,
      "IsView": "N",
      "Value": "N",
      "Name": "InstrumentStatus"
    },
    "SaveStage": {
      "Length": 255,
      "Type": "3110",
      "VarFieldId": 0,
      "VariableId": 34,
      "IsView": "N",
      "Value": "Start Event_1",
      "Name": "SaveStage"
    },
    "Status": {
      "Length": 255,
      "Type": "3110",
      "VarFieldId": 0,
      "VariableId": 42,
      "IsView": "N",
      "Value": "null",
      "Name": "Status"
    },
    "CalendarName": {
      "Length": 255,
      "Type": "3110",
      "VarFieldId": 0,
      "VariableId": 10001,
```

```
      "IsView": "N",
      "Value": "null",
      "Name": "CalendarName"
    },
    "ChildProcessInstanceId": {
      "Length": 255,
      "Type": "313110",
      "VarFieldId": 0,
      "VariableId": -1,
      "IsView": "N",
      "Value": "null",
      "Name": "ChildProcessInstanceId"
    },
    "ChildWorkItemId": {
      "Length": 255,
      "Type": "31313",
      "VarFieldId": 0,
      "VariableId": -2,
      "IsView": "N",
      "Value": "null",
      "Name": "ChildWorkItemId"
    },
    "ProcessInstanceState": {
      "Length": 255,
      "Type": "323",
      "VarFieldId": 0,
      "VariableId": 40,
      "IsView": "N",
      "Value": "1",
      "Name": "ProcessInstanceState"
    },
    "CreatedDateTime": {
      "Length": 255,
      "Type": "318",
      "VarFieldId": 0,
      "VariableId": 28,
      "IsView": "N",
      "Value": "2022-08-12 07: 25: 36",
      "Name": "CreatedDateTime"
    },
    "CreatedByName": {
      "Length": 255,
      "Type": "3110",
      "VarFieldId": 0,
      "VariableId": 32,
      "IsView": "N",
      "Value": "ritika",
      "Name": "CreatedByName"
    },
    "IntroductionDateTime": {
      "Length": 255,
```

```
        "Type": "318",
        "VarFieldId": 0,
        "VariableId": 35,
        "IsView": "N",
        "Value": "null",
        "Name": "IntroductionDateTime"
      },
      "IntroducedBy": {
        "Length": 255,
        "Type": "3110",
        "VarFieldId": 0,
        "VariableId": 36,
        "IsView": "N",
        "Value": "null",
        "Name": "IntroducedBy"
      },
      "IntroducedAt": {
        "Length": 255,
        "Type": "3110",
        "VarFieldId": 0,
        "VariableId": 10003,
        "IsView": "N",
        "Value": "Start Event_1",
        "Name": "IntroducedAt"
      },
      "PriorityLevel": {
        "Length": 255,
        "Type": "323",
        "VarFieldId": 0,
        "VariableId": 38,
        "IsView": "N",
        "Value": "1",
        "Name": "PriorityLevel"
      },
      "WorkItemState": {
        "Length": 255,
        "Type": "323",
        "VarFieldId": 0,
        "VariableId": 41,
        "IsView": "N",
        "Value": "2",
        "Name": "WorkItemState"
      },
      "ActivityId": {
        "Length": 255,
        "Type": "323",
        "VarFieldId": 0,
        "VariableId": 43,
        "IsView": "N",
        "Value": "1",
        "Name": "ActivityId"
```

```
    },
    "LockedByName": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 46,
      "IsView": "N",
      "Value": "ritika",
      "Name": "LockedByName"
    },
    "LockedTime": {
      "Length": 255,
      "Type": "328",
      "VarFieldId": 0,
      "VariableId": 47,
      "IsView": "N",
      "Value": "2022-08-17 14: 32: 32",
      "Name": "LockedTime"
    },
    "LockStatus": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 48,
      "IsView": "N",
      "Value": "Y",
      "Name": "LockStatus"
    },
    "ActivityName": {
      "Length": 255,
      "Type": "3110",
      "VarFieldId": 0,
      "VariableId": 49,
      "IsView": "N",
      "Value": "Start Event_1",
      "Name": "ActivityName"
    },
    "AssignmentType": {
      "Length": 255,
      "Type": "3110",
      "VarFieldId": 0,
      "VariableId": 51,
      "IsView": "N",
      "Value": "S",
      "Name": "AssignmentType"
    },
    "ProcessedBy": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 52,
```

```
      "IsView": "N",
      "Value": "ritika",
      "Name": "ProcessedBy"
    },
    "EntryDateTime": {
      "Length": 255,
      "Type": "318",
      "VarFieldId": 0,
      "VariableId": 29,
      "IsView": "N",
      "Value": "2022-08-12 07: 25: 36",
      "Name": "EntryDateTime"
    },
    "ValidTillDateTime": {
      "Length": 255,
      "Type": "318",
      "VarFieldId": 0,
      "VariableId": 30,
      "IsView": "N",
      "Value": "null",
      "Name": "ValidTillDateTime"
    },
    "WorkItemName": {
      "Length": 255,
      "Type": "3110",
      "VarFieldId": 0,
      "VariableId": 31,
      "IsView": "N",
      "Value": "null",
      "Name": "WorkItemName"
    },
    "PreviousStage": {
      "Length": 255,
      "Type": "3110",
      "VarFieldId": 0,
      "VariableId": 33,
      "IsView": "N",
      "Value": "Start Event_1",
      "Name": "PreviousStage"
    },
    "TurnAroundDateTime": {
      "Length": 255,
      "Type": "318",
      "VarFieldId": 0,
      "VariableId": 10002,
      "IsView": "N",
      "Value": "null",
      "Name": "TurnAroundDateTime"
    },
    "WorkItemId": {
      "Length": 255,
```

```
        "Type": "323",
        "VarFieldId": 0,
        "VariableId": 10004,
        "IsView": "N",
        "Value": "1",
        "Name": "WorkItemId"
      },
      "SecondaryDBFlag": {
        "Length": 255,
        "Type": "3210",
        "VarFieldId": 0,
        "VariableId": 10022,
        "IsView": "N",
        "Value": "N",
        "Name": "SecondaryDBFlag"
      },
      "URN": {
        "Length": 255,
        "Type": "3110",
        "VarFieldId": 0,
        "VariableId": 10023,
        "IsView": "N",
        "Value": "portalprocess-8583",
        "Name": "URN"
      },
      "ManualProcessingFlag": {
        "Length": 255,
        "Type": "3210",
        "VarFieldId": 0,
        "VariableId": 10024,
        "IsView": "N",
        "Value": "N",
        "Name": "ManualProcessingFlag"
      },
      "DBExErrCode": {
        "Length": 255,
        "Type": "323",
        "VarFieldId": 0,
        "VariableId": 10025,
        "IsView": "N",
        "Value": "null",
        "Name": "DBExErrCode"
      },
      "DBExErrDesc": {
        "Length": 255,
        "Type": "3210",
        "VarFieldId": 0,
        "VariableId": 10026,
        "IsView": "N",
        "Value": "null",
        "Name": "DBExErrDesc"
```

```
    },
    "Locale": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 10027,
      "IsView": "N",
      "Value": "en_US",
      "Name": "Locale"
    },
    "CurrentDateTime": {
      "Length": 255,
      "Type": "318",
      "VarFieldId": 0,
      "VariableId": 27,
      "IsView": "N",
      "Value": "2022-08-17 20: 02: 53.353",
      "Name": "CurrentDateTime"
    },
    "QueueName": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 45,
      "IsView": "N",
      "Value": "portal_process_Start Event_1",
      "Name": "QueueName"
    },
    "QueueType": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 44,
      "IsView": "N",
      "Value": "I",
      "Name": "QueueType"
    },
    "ApplicantType": {
      "Length": 50,
      "Type": "2310",
      "VarFieldId": 0,
      "VariableId": 58,
      "IsView": "N",
      "Value": "Adult",
      "Name": "ApplicantType"
    },
    "ApplicantAddress": {
      "Length": 50,
      "Type": "2310",
      "VarFieldId": 0,
      "VariableId": 59,
```

```
        "IsView": "N",
        "Value": "Pheonix, Arizona",
        "Name": "ApplicantAddress"
      },
      "Employer": {
        "Length": 50,
        "Type": "2310",
        "VarFieldId": 0,
        "VariableId": 60,
        "IsView": "N",
        "Value": "Nutanix",
        "Name": "Employer"
      }
    }
  },
  "LockStatus": "Y",
  "CacheTime": "2022-08-12 07: 08: 15",
  "InstrumentStatus": "N",
  "Var_Rec1": "11954",
  "LockResult": "Y",
  "SessionId": "-443031847",
  "UserName": "ritika",
  "URN": "portalprocess-8583"
}
```


WMCompleteWorkitem

The *WMCompleteWorkitem* API allows you to specify the workitem completed and ready to route to the next step. For more details, you can refer to the JSON request and response parameters provided in the next sub-sections.

- Service URL — *http://<GatewayIp: GatewayPort >/workitem-service-context/workitem/completeworkitem*
  Where, GatewayIp is the IP of the machine, and GatewayPort is the port of the machine where the workitem service is deployed.
- Request Media Type — application/json
- Response Media Type — application/json
- Method Type — POST

Header parameters description

The following table describes the header parameters of the *WMCompleteWorkitem* API, if OAuth is disabled:

Name	Parameter Scope	Type	Description	Mandatory
Authorization	HeaderParam	String	The session ID of the signed-in user.	Yes
Accept-Language	HeaderParam	Locale	Represents the natural language and locale that the client prefers. Example, en-US.	No
Tenant-Id	HeaderParam	String	The name of the cabinet.	Yes

If OAuth is enabled:

Name	Parameter scope	Type	Description	Mandatory
AppId	HeaderParam	String	Registered AppId of the product	No
Access Token	HeaderParam	String	AccessToken for the signed-in user	No
SessionValid	HeaderParam	String	SessionValid for the signed-in user	No

Request body parameters description

The following table describes the request body parameters of the *WMCompleteWorkitem* API:

Name	Parent Tag	Type	Description	Valid Values	Default Values	Mandatory
Process InstanceId	Root	String	Process Instance ID to mark complete corresponding with workitem.		NA	Yes
WorkItemId	Root	Integer	Workitem ID to mark complete.		NA	Yes

Name	Parent Tag	Type	Description	Valid Values	Default Values	Mandatory
ActivityId	Root	Integer	Integer Represents the activity ID of the process.		NA	Yes
AuditStatus	Root	Character	Indicates the audit status of the workitem. Specified in case an Audit Workitem is completed by an auditor.	A - Accepted R - Rejected	R	No
Comments	Root	String	Represents the comments added while completing the workitems.		NA	No

Response body parameters description

The following table describes the response body parameters of the *WMCompleteWorkitem* API:

Name	Parent Tag	Parameter Type	Description
URN	Root	String	URN corresponding to the workitem that is to be marked completed.
ProcessDefId	Root	Integer	The ID of the process in which the workitem is marked completed.
WorkItemId	Root	Integer	The ID of the workitem that is marked completed.
Process InstanceId	Root	String	Process instance ID corresponding to workitem which is marked as completed.
ActivityId	Root	Integer	Activity ID on which workitem is completed.
ActivityName	Root	String	Activity name on which workitem is completed.
Completion Time	Root	Integer	Completion date time of the workitem on the activity.
EntryDateTime	Root	Date	Entry date time of the workitem on the activity.

Error codes and description

The following table describes the error codes that can occur while executing the *WMCompleteWorkitem* API:

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS _5	409	Conflict	The activity instance ID was not valid.	ActivityID passed is incorrect or workitem is not present on the specified activities.
NGONE_WFS _2	400	Bad Request	Invalid Process Definition.	ProcessDefID value is not found for the workitem specified in the request. Ensure that the ProcessInstanceID value passed in the request body is correct.
NGONE_WFS _400	500	Internal Server Error	Operation Failed.	Recheck the request body.
NGONE_WFS _6	500	Internal Server Error	The workitem is invalid.	The requested workitem is not eligible to be completed by the user.
N/A	401	Unauthorized	Invalid Session.	Regenerate the session and pass the valid session Id in the authorization header.

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_3001	400	Bad Request	Invalid Input Parameters.	Refer to the detailed error message to correct input parameters which have invalid values. You can refer to the below sample for reference (FieldName: ErrorDescription), 1. WorkItemId: WorkItemId must have value greater than equal to zero and positive integer. 2.ActivityId : ActivityId must have value greater than equal to zero and positive integer. • There can be multiple scenarios for this error code. To resolve the issue, refer to the detailed error message.
NGONE_WFS_801	500	Internal Server Error	SQL Exception Occurred.	An SQL exception occurred which stopped the operation. Refer to the Error logs for further analysis.
NGONE_WFS_503	503	Temporary	Service is either down or slow.	Service is either unavailable or slow.

Sample request and response

Request:

```
{
  "ProcessInstanceId": "RJ-0000000007-process",
  "WorkItemId": 1,
  "ActivityId": 2
}
```

Response:

```
{
  "EntryDateTime": "2022-08-26 13:15:43",
  "CompletionTime": 21219,
  "ActivityName": "Workdesk_2",
  "ActivityId": 2,
  "ProcessInstanceId": "RJ-0000000007-process",
  "WorkItemId": 1,
  "ProcessDefId": 1015
}
```

WMUnlockWorkitem

The *WMUnlockWorkitem* unlocks a currently locked workitem or all the locked workitems for a given user. For more details, you can refer to the JSON request and response parameters provided in the next sub-sections.

- Service URL — *http://<GatewayIp: GatewayPort >/workitem-service-context/workitem/unlock*
  Where, GatewayIp is the IP of the machine, and GatewayPort is the port of the machine where the workitem service is deployed.
- Request Media Type — application/json
- Response Media Type — application/json
- Method Type — PUT

Header parameters description

The following table describes the header parameters of the *WMUnlockWorkitem* API, if OAuth is disabled:

Name	Parameter Scope	Type	Description	Mandatory
Authorization	HeaderParam	String	The session ID of the signed-in user.	Yes
Accept-Language	HeaderParam	Locale	Represents the natural language and locake that the client prefers. Example, en-US	No
Tenant-Id	HeaderParam	String	Name of the cabinet.	Yes

If OAuth is enabled:

Name	Parameter scope	Type	Description	Mandatory
AppId	HeaderParam	String	Registered AppId of the product	No
Access Token	HeaderParam	String	AccessToken for the signed-in user	No
SessionValid	HeaderParam	String	SessionValid for the signed-in user	No

Request body parameters description

The following table describes the request body parameters of the *WMUnlockWorkitem* API:

Name	Parent Tag	Type	Description	Default Value	Valid Value	Mandatory
WorkItemId	Root	Integer	Workitem ID to unlock the workitem.	NA	NA	Yes

Name	Parent Tag	Type	Description	Default Value	Valid Value	Mandatory
IsAdmin	Root	String	Refers to whether the logged-in user is admin or not.	Y-Yes N-No	N	No
IsDisconnect	Root	String	API is also called from DisconnectCabinet.	Y – Yes N - No	N	Yes
Process InstanceId	Root	String	Process instance ID to unlock the corresponding workitem.	NA	NA	Yes
UnlockOption	Root	String	• Execution mode - Workitem mode or User mode. • In User mode, all WIs locked by the user get unlocked. • In workitem mode, a single workitem gets unlocked.	W- Workitem U- User	W	No
ActivityId	Root	Integer	Id of the Activity.	NA	NA	Yes

Response body parameters description

The following table describes the response body parameters of the *WMUnlockWorkitem* API:

Name	Parent Tag	Parameter Type	Description
mainCode	Root	Integer	The API returned success or error code.

Error codes and description

The following table describes the error codes that can occur while executing the *WMUnlockWorkitem* API:

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_400	409	CONFLICT	Operation failed	Recheck the request body.
NGONE_WFS_17	409	CONFLICT	The workitem is not locked by the user.	Ensure that the passed workitem is locked by the signed-in user.
NGONE_WFS_6	409	CONFLICT	The workitem is invalid	The workitem passed in the request body is incorrect. Recheck the request.
N/A	401	Unauthorized	Invalid Session	Regenerate the session and pass the valid session Id in the authorization header.
NGONE_WFS_801	500	Internal Server Error	SQL Exception Occurred.	An SQL exception occurred which stopped the operation Refer to the Error logs for further analysis.
NGONE_WFS_3001	400	Bad Request	Invalid Input Parameters.	Refer to the detailed error message to correct input parameters which have invalid values. You can refer to the below sample for reference (FieldName: ErrorDescription), 1. IsDisconnect: IsDisconnect must have value Y or N. 2. UnlockOption: UnlockOption must have values U or W. 3. ActivityId: ActivityId must have value greater than equal to zero and positive integer. • There can be multiple scenarios for this error code. To resolve the issue, refer to the detailed error message.
NGONE_WFS_503	503	Temporary	Service is either down or slow.	Service is either unavailable or slow.

Sample request and response

Request:

```
{
  "ProcessInstanceId": "RJ-0000000007-process",
  "WorkItemId": 2,
  "UnlockOption": "W",
  "ActivityId": 3,
  "IsDisconnect": "N",
  "isAdmin": "N"
}
```

Response:

```
{
  "mainCode": 0
}
```

WFSetAttributes

The WFSetAttributes API allows you to assign values to attributes of a workitem. It is also used to specify values for multiple workitems at a time. For more details, you can refer to the JSON request and response parameters provided in the next sub-sections.

- Service URL — *http://<GatewayIp:GatewayPort>/workitem-service-context/workitem/setattributes*
  Where, GatewayIp is the IP of the machine, and GatewayPort is the port of the machine where the workitem service is deployed.
- Request Media Type - application/json
- Response Media Type - application/json
- Method Type - PUT

Header parameters description

The following table describes the header parameters of the *WFSetAttributes* API, if OAuth is disabled:

Name	Parameter Scope	Type	Description	Mandatory
Authorization	HeaderParam	String	The session ID of the signed-in user.	Yes
Accept-Language	HeaderParam	Locale	Represents the natural language and locale that the client prefers. Example, en-US.	No
Tenant-Id	HeaderParam	String	The name of the cabinet.	Yes

If OAuth is enabled:

Name	Parameter scope	Type	Description	Mandatory
AppId	HeaderParam	String	Registered AppId of the product	No
Access Token	HeaderParam	String	AccessToken for the signed-in user	No
SessionValid	HeaderParam	String	SessionValid for the signed-in user	No

Request body parameters description

The following table describes the request body parameters of the *WFSetAttributes* API:

Name	Parent Type	Type	Description	Valid Values	Default Values	Mandatory
Process InstanceID	Root	String	The ID of the valid Process instance which is in a locked state by the session id	NA	NA	Yes
WorkItemID	Root	Integer	Workitem ID for which attribute values must be set.	NA	NA	Yes
ActivityID	Root	Integer	Integer Current activity id in which workitem is present.	N/A	N/A	Yes
ActivityType	Root	Integer	Current activity Type in which workitem is present.	NA	0	No
Complete	Root	String	Indicates whether the workitem is completed or initiated.	D-Done I-Initiate	N	No
Comments	Root	String	Comments to assign values to the attributes of a workitem.	NA	NA	No
IsValidation Required	Root	String	Indicates whether the data is validated before trying to update process variables.	Y or N	N	No

Name	Parent Type	Type	Description	Valid Values	Default Values	Mandatory
Attributes	Root	Map<String, Object>	Contains the keys as attribute names or user-defined names along with corresponding values.	NA	N	No

Name	Parent Type	Type	Description	Valid Values	Default Values	Mandatory
Insertion OrderId	Attributes	Long	InsertionOrderId passes in the SetAttributes API call and the value of InsertionOrderID defines the operation type that must be performed on the Complex array elements.	Following are the 3 cases: • The positive value passed in Insertion OrderId tag: Defines the insertion Order Id corresponding to the attributes passed that must be updated. • 0 value passed in Insertion OrderId tag: Defines the values passed for complex array and must be inserted. • The negative value passed in Insertion OrderId tag: Defines the Insertion OrderId for which the user needs to delete the row.	N	No

Name	Parent Type	Type	Description	Valid Values	Default Values	Mandatory
HashId	Attributes	Long	In case of creating a new record in a complex array, Sent the Unique HashId for each row of the complex array. HashId. New Insertion OrderId generated for the row is returned in response means, corresponding to the HashId passed in the request.	NA	N	No

Response body parameters description

The following table describes the response body parameters of the *WFSetAttributes* API:

Name	Parent Tag	Parameter Type	Description
Insertion OrderIdValue	Root	Map<String, String>	InsertionOrderId passes in the SetAttributes API call and the value of InsertionOrderID defines the type of performed operation on Complex array elements.
Start Process Response	Root	WFStartProcess Response	When the complete flag is “I”.
Complete Process Response	Root	WFComplete WorkItem Response	When the complete flag is “D”.

Name	Parent Tag	Parameter Type	Description
ProcessDefId	WFComplete WorkItem Response	Integer	The ID of the process definition corresponding to the workitem.
WorkItemId	WFComplete WorkItem Response	Integer	Id of the workitem.
Process InstanceId	WFComplete WorkItem Response	String	Process instance ID of the workitem.
ActivityId	WFComplete WorkItem Response	Integer	Activity ID on which the workitem is completed.
ActivityName	WFComplete WorkItem Response	String	Activity name on which workitem is completed.
Completion Time	WFComplete WorkItem Response	Integer	Completion time of the workitem on the activity.
WorkItemId	WFStart Process Response	Integer	Refers to the workitem id.
URN	WFStart Process Response	String	Refers to the URN no of the process Instance.

Error codes and descriptions

The following table describes the error codes that can occur while executing the *WFSetAttributes* API:

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_6	404	Not Found	The WorkItemId is invalid.	The workitem is not locked by the user. Ensure to execute the WFGetWorkitemDataExt API to lock the workitem before executing the WMUnlockWorkitem API.
NGONE_WFS_22	401	Unauthorized	The WorkItem is invalid.	The Workitem is locked by some other user. And this operation cannot be performed unless the workitem is unlocked by another user. Hence, it must be locked by the signed-in user.
NGONE_WFS_400	400	Bad Request	Requested Operation Failed.	Recheck the request body.
NGONE_WFS_8	400	Bad Request	The attribute assignment is failed.	Recheck the Complex Variable Definition. It can be due to the column used in complex relation is also used in mapping.
NGONE_WFS_4017	400	Bad Request	The Variable Name Is Invalid or Readonly cannot be modified.	Edit rights are missing for the specified attribute on the activity in which this workitem is present.
N/A	401	Unauthorized	Invalid Session.	Regenerate the session and pass the valid session Id in the authorization header.

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_3001	400	Bad Request	Invalid Input Parameters.	Refer to the detailed error message to correct the input parameters which have invalid values. You can refer to the below sample for reference (FieldName: ErrorDescription), 1.Complete: Complete value must be I or Y. 2.PDAFlag: PdaFlag value must be Y or N. 3. ActivityId: ActivityId must have value greater than equal to zero and positive integer. • There can be multiple scenarios for this error code. To resolve the issue, refer to the detailed error message.
NGONE_WFS_802	400	Bad Request	Invalid Parameter.	Refer to the detailed error message to correct parameters which have invalid values.
NGONE_WFS_801	500	Internal Server Error	SQL exception occurred.	An SQL exception occurred which stopped the operation. Refer to the Error logs for further analysis.
NGONE_WFS_503	503	Temporary	Service is either down or slow.	Service is either unavailable or slow.

Sample request and response

Below is the sample request and response body for *WFSetAttributes* API as per the following scenarios.

Updating queue and external variables

Request:

```
{
  "ActivityId": 1,
  "ProcessInstanceId": "WF-0000001062-uploadprocess",
  "ProcessDefId": 2004,
  "WorkitemId": 1,
  "Attributes": {
    "Age": 26,
    "Salary": 25000,
    "Name": "Claire",
    "Address": "Pheonix, Arizona",
    "LastName": "Bowman"
  }
}
```

Response:

```
{
  "InsertionOrderIdValue": {},
  "WorkitemUnholded": "N"
}
```

Updating nested complex array variables

Request:

```
{
  "ActivityId": 1,
  "ProcessInstanceId": "WF-0000001098-
  uploadprocess",
  "ProcessDefId": 2004,
```

```
  "WorkitemId": 1,
  "parentLogId": 0,
  "Attributes": {
  "age": 10,
  "Salary": 888,
  "name": "DivyanshuKUMAR",
  "ext_age": 23,
  "ext_name": "Divyanshu",
  "complex_address": {
  "user_city": "NOIDA",
  "user_pincode": 801503,
  "user_state": "UP"
  },
  "CompanyVar": {
  "InsertionOrderId": 0,
  "HashId": 1,
  "CompName": "NEWGEN",
  "CLocation": "NOIDA",
  "EmployeeV": [
  {
   "InsertionOrderId": 0,
   "HashId": 2,
   "EName": "ALICE",
   "EAge": 23,
   "ESal": 222},
   {
   "InsertionOrderId": 0,
   "HashId": 3,
   "EName": "BEN",
   "EAge": 22,
   "ESal": 234
   }
   ]
   }
   }
   }
```

Response:

```
{
  "InsertionOrderIdValue": {
    "1": "21",
    "2": "33",
    "3": "34"
  },
  "WorkitemUnholded": "N"
}
```

Updating nested complex array variables and completing workitem

This section contains the request and response body to update nested complex array variables and completing the workitem in a single request.

Request:

```
{
  "ActivityId": 1,
  "ProcessInstanceId": "WF-0000001098- uploadprocess",
  "ProcessDefId": 2004,
  "WorkitemId": 1,
  "Attributes": {
    "Age": 32,
    "Salary": 25000,
    "Name": "Ben",
    "Zone": 25,
    "LastName": "Beckman",
    "Address": {
      "City": "Dresden",
      "Zip": 1136,
      "Country": "Germany"
    },
    "Company": {
      "InsertionOrderId": 0,
      "HashId": 1,
      "CompName": "Newgen Software Technologies",
      "CompLocation": "Dresden",
      "Employee": [
        {
          "InsertionOrderId": 0,
          "HashId": 2,
          "EName": "Alex",
          "EAge": 28,
          "ESal": 5500
        },
        {
          "InsertionOrderId": 0,
          "HashId": 3,
          "EName": "Jake",
          "EAge": 22,
          "ESal": 4800
        }
      ]
    }
```

```
  }
}
```

Response:

```
{
  "InsertionOrderIdValue": {
    "1": "10075",
    "2": "10050",
    "3": "10051"
  },
  "WorkitemUnholded": "N",
  "WFCompleteWorkItemResponse": {
    "EntryDateTime": "2022-04-13 12:44:14",
    "CompletionTime": 435200,
    "ActivityName": "Workdesk_2",
    "ActivityId": 2,
    "ProcessInstanceId": "WF-0000002101-uploadprocess",
    "WorkItemId": 1,
    "ProcessDefId": 2004
  }
}
```

WFFetchAttributes

The WFFetchAttributes API call returns the attribute of a specified workitem depending upon the filter criterion specified.

- Service URL — *http://<GatewayIP:GatewayPort>/workitem-service-context/workitem/fetchattributes*
  Where, GatewayIp is the IP of the machine, and GatewayPort is the port of the machine where the workitem service is deployed.
- Request Media Type — application/json
- Response Media Type — application/json
- Method Type — POST

Header parameters description

The following table describes the header parameters of the *WFFetchAttributes* API, if OAuth is disabled:

Name	Parameter Scope	Type	Description	Mandatory
Authorization	HeaderParam	String	The session ID of the signed-in user.	Yes
Accept-Language	HeaderParam	Locale	Represents the natural language and locale that the client prefers. Example, en-US.	No
Tenant-Id	HeaderParam	String	Name of the cabinet.	Yes

If OAuth is enabled:

Name	Parameter scope	Type	Description	Mandatory
AppId	HeaderParam	String	Registered AppId of the product	No
Access Token	HeaderParam	String	AccessToken for the signed-in user	No
SessionValid	HeaderParam	String	SessionValid for the signed-in user	No

Request body parameters description

The following table describes the request body parameters of the *WFFetchAttributes* API:

Name	Parent Tag	Type	Description	Valid Values	Default Values	Mandatory
Process InstanceId	Root	String	Process instance ID to fetch the workitem attributes. Fetching the attributes of a workitem requires the ID of the Process Instance.	NA	NA	Yes
WorkItemId	Root	Integer	Workitem ID to fetch the workitem attributes	NA	NA	Yes
ActivityID	Root	Integer	Activity ID to fetch the attributes.	NA	NA	Yes
ProcessDefId	Root	Integer	Process definition ID corresponding to the workitem to fetch the attributes.	NA	NA	Yes
Attribute Name	Root	String	Name of fetched attributes.	NA	NA	Yes
NoOfRecords ToFetch	Root	Integer	Contains the maximum number of records to return.	NA	NA	No
LastValue	BatchInfo	String	Contains the last value of field for the last record in previous batch on which sorting in applied. By Default, sorting is applied on InsertionOrderId.	NA	NA	No

Name	Parent Tag	Type	Description	Valid Values	Default Values	Mandatory
LastInsertionOrderIdValue	BatchInfo	Integer	Contains the insertionorderid of the last record returned in the previous batch.	NA	NA	No
SortOrder	BatchInfo	String	Indicates the columns sorting order.	NA	A- Ascending D- Descending	No
OrderBy	BatchInfo	String	Indicates that the columns through which the returned result gets sorted.	NA	NA	No
BatchInfo	Root	String	Batching support (JSONObject in String Format).	NA	NA	No
NoOfRecordsToFetch	BatchInfo	Integer	Contains the maximum number of records to return.	NA	NA	No
FilterXML	BatchInfo	List <Filter>	List of filters to apply on search.	NA	NA	No
Filter	FilterXML	Filter	Filter to apply on search.	NA	NA	No
VarName	Filter	String	Name of the attribute on which filter is applied.	NA	NA	No

Name	Parent Tag	Type	Description	Valid Values	Default Values	Mandatory
Operator	Filter	Integer	Operator to define the relation between the attribute name and its value. For Example: name='Ram'; = is an operator).	1 – Less Than 2 – Less Than Equal To 3 – Equal To 4 – Not Equal To 5 – Greater Than 6 – Greater Than Equal To 7 - Like	NA	No
VarValue	Filter	String	The corresponding value of the attribute on which filter is applied.	NA	NA	No
Logical Operator	Filter	Integer	To join the multiple filter Condition.	0 - The last filter gets applied. Hence, there is no adding string. 1 - AND 2 - OR	N	No

Response body parameters description

The following table describes the response body parameters of the *WFFetchAttributes* API:

Name	Parent Tag	Parameter Type	Description
Attributes	Root	LinkedHashMap	List of the attributes fetched along with their details.
Name	Attributes	String	Name of the data attribute.
Type	Attributes	String	Type of the data attribute. This is a numeric value, where each digit corresponds to some information about the attribute. • 1st digit indicates the Type of the Variable, whether System, Queue or External Table. 1 - Queue 2 - External 3 - System • 2nd Digit indicates the access permissions on the attribute, on the current activity. 1 - Read 2 - Modify 3 – Both Read and Modify • 3rd and 4th Digits indicate the Data Type of the variable. 3 - Integer 4 - Long 6 - Float 8 - Date 10 - Text
Length	Attributes	Integer	Length of the attribute.

Name	Parent Tag	Parameter Type	Description
Value	Attributes	String	The current value of the attribute.
VariableId	Attributes	Integer	VariableId of the attributes.
VarFieldId	Attributes	Integer	VarfieldId is the Id of a field or attribute that is, a part of a complex structure. For a primitive variable, it is '0'.

Error codes and description

The following table describes the error codes that can occur while executing the *WFFetchAttributes* API:

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_6	404	Not Found	The WorkItemId is invalid.	The workitem passed in the request body is incorrect. Recheck the request.
NGONE_WFS_400	404	Not Found	The Requested Operation Failed.	Recheck the request body.
N/A	401	Unauthorized	Invalid Session.	Regenerate the session and pass the valid session Id in the authorization header.

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_3001	400	Bad Request	Invalid Input Parameters.	Refer to the detailed error message to correct input parameters which have invalid values. You can refer to the below sample for reference (FieldName: ErrorDescription). 1.NoOfRecordsToFetch: NoOfRecordsToFetch must have value greater than equal to zero and positive integer. 2. ActivityType: ActivityType must have value greater than equal to zero and positive integer. • There can be multiple scenarios for this error code. To resolve the issue, refer to the detailed error message.
NGONE_WFS_801	500	Internal Server Error	SQL Exception Occurred.	An SQL exception occurred which stopped the operation. Refer to the Error logs for further analysis.
NGONE_WFS_503	503	Temporary	Service is either down or slow.	Service is either unavailable or slow.

Sample request and response

Below is the sample request and response body for *WFFetchAttributes* API as per the following scenarios.

Fetching attributes without batch information

Request:

```
{
  "NoOfRecordsToFetch": 5,
  "ProcessDefId": 1010,
  "ActivityID": 1,
  "WorkItemId": 1,
  "ProcessInstanceId": "Process-0000000000001-27July2022",
  "AttributeName": "Department"
}
```

Response:

```
{
  "Attributes": {
    "QueueName": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 45,
      "IsView": "N",
      "Value": "MSTesting_Scanning",
      "Name": "QueueName"
    },
    "QueueType": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 44,
      "IsView": "N",
      "Value": "I",
      "Name": "QueueType"
    },
    "Department": [
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "2",
        "MaxInsertionOrderId": "2",
        "MinInsertionOrderId": "1",
        "Child": {
          "Employees": [
            {
              "Length": 0,
              "Type": "2311",
              "VarFieldId": 4,
              "VariableId": 58,
              "IsView": "N",
```

```
            "Value": "",
            "InsertionOrderId": "1",
            "MaxInsertionOrderId": "1",
            "MinInsertionOrderId": "1",
            "Child": {
              "EmployeeDOJ": {
                "Length": 8,
                "Type": "238",
                "VarFieldId": 7,
                "VariableId": 58,
                "IsView": "N",
                "Value": "2022-12-12 00: 00: 00",
                "Name": "EmployeeDOJ"
              },
              "EmployeeId": {
                "Length": 4,
                "Type": "234",
                "VarFieldId": 6,
                "VariableId": 58,
                "IsView": "N",
                "Value": "1",
                "Name": "EmployeeId"
              },
              "EmployeeName": {
                "Length": 50,
                "Type": "2310",
                "VarFieldId": 5,
                "VariableId": 58,
                "IsView": "N",
                "Value": "Phil",
                "Name": "EmployeeName"
              }
            },
            "Name": "Employees"
          }
        ],
        "DeptName": {
          "Length": 50,
          "Type": "2310",
          "VarFieldId": 2,
          "VariableId": 58,
          "IsView": "N",
          "Value": "PRDP",
          "Name": "DeptName"
        },
        "DeptId": {
          "Length": 4,
          "Type": "234",
          "VarFieldId": 3,
          "VariableId": 58,
          "IsView": "N",
```

```
            "Value": "1",
            "Name": "DeptId"
          }
        },
        "Name": "Department"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "2",
        "MaxInsertionOrderId": "2",
        "MinInsertionOrderId": "1",
        "Child": {
          "Employees": [
            {
              "Length": 0,
              "Type": "2311",
              "VarFieldId": 4,
              "VariableId": 58,
              "IsView": "N",
              "Value": "",
              "InsertionOrderId": "2",
              "MaxInsertionOrderId": "2",
              "MinInsertionOrderId": "2",
              "Child": {
                "EmployeeDOJ": {
                  "Length": 8,
                  "Type": "238",
                  "VarFieldId": 7,
                  "VariableId": 58,
                  "IsView": "N",
                  "Value": "2022-01-12 00: 00: 00",
                  "Name": "EmployeeDOJ"
                },
                "EmployeeId": {
                  "Length": 4,
                  "Type": "234",
                  "VarFieldId": 6,
                  "VariableId": 58,
                  "IsView": "N",
                  "Value": "2",
                  "Name": "EmployeeId"
                },
                "EmployeeName": {
                  "Length": 50,
                  "Type": "2310",
                  "VarFieldId": 5,
```

```
                  "VariableId": 58,
                  "IsView": "N",
                  "Value": "Alice",
                  "Name": "EmployeeName"
                }
              },
              "Name": "Employees"
            }
          ],
          "DeptName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "InnovationHub",
            "Name": "DeptName"
          },
          "DeptId": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "2",
            "Name": "DeptId"
          }
        },
        "Name": "Department"
      }
    ]
  }
}
```


Fetching attributes with batch information

Request:


```
{
  "NoOfRecordsToFetch": 10,
  "ProcessDefId": 1010,
  "ActivityID": 1,
  "WorkItemId": 1,
  "ProcessInstanceId": "Process-0000000000001-27July2022",
  "AttributeName": "Department",
  "BatchInfo": "{\"Department\":{\"SortOrder\":\"A\",
\"OrderBy\":1,\"NoOfRecordsToFetch\":5,\"FilterXML\":{\"Filter\":[{\"VarName\":
```

```
\"DeptName\",\"Operator\":3,\"VarValue\":\"AP6\",\"LogicalOperator\":2},
{\"VarName\":\"DeptId\",\"Operator\":3,\"VarValue\":\"8\",
\"LogicalOperator\":0}]}}}"
}
```

Response:


```
{
  "Attributes": {
    "QueueName": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 45,
      "IsView": "N",
      "Value": "MSTesting_Scanning",
      "Name": "QueueName"
    },
    "QueueType": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 44,
      "IsView": "N",
      "Value": "I",
      "Name": "QueueType"
    },
    "Department": [
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "11",
        "MaxInsertionOrderId": "11",
        "MinInsertionOrderId": "11",
        "Child": {
          "Employees": [
            {
              "Length": 0,
              "Type": "2311",
              "VarFieldId": 4,
              "VariableId": 58,
              "IsView": "N",
              "Value": "",
              "InsertionOrderId": "11",
              "MaxInsertionOrderId": "11",
              "MinInsertionOrderId": "11",
              "Child": {
```

```
            "EmployeeDOJ": {
              "Length": 8,
              "Type": "238",
              "VarFieldId": 7,
              "VariableId": 58,
              "IsView": "N",
              "Value": "2022-07-12 00: 00: 00",
              "Name": "EmployeeDOJ"
            },
            "EmployeeId": {
              "Length": 4,
              "Type": "234",
              "VarFieldId": 6,
              "VariableId": 58,
              "IsView": "N",
              "Value": "8",
              "Name": "EmployeeId"
            },
            "EmployeeName": {
              "Length": 50,
              "Type": "2310",
              "VarFieldId": 5,
              "VariableId": 58,
              "IsView": "N",
              "Value": "Phil",
              "Name": "EmployeeName"
            }
          },
          "Name": "Employees"
        }
      ],
      "DeptName": {
        "Length": 50,
        "Type": "2310",
        "VarFieldId": 2,
        "VariableId": 58,
        "IsView": "N",
        "Value": "AP6",
        "Name": "DeptName"
      },
      "DeptId": {
        "Length": 4,
        "Type": "234",
        "VarFieldId": 3,
        "VariableId": 58,
        "IsView": "N",
        "Value": "8",
        "Name": "DeptId"
      }
    },
    "Name": "Department"
```

```
        }
      ]
    }
}
```

Fetching attributes with batching and sorting

The following are the sample with batching and sorting scenarios:

- Batching and sorting (fetching first five records using company name)
- Batching and sorting (fetching next five records sorted using company name)
- Batching and sorting (fetching previous five records sorted using company name)

Batching and sorting (fetching first five records using company name)

Request:

```
{
  "NoOfRecordsToFetch": 5,
  "ProcessDefId": 1021,
  "ActivityID": 1,
  "WorkItemId": 1,
  "ProcessInstanceId": "Nested-0000000001-process",
  "AttributeName": "CompanyMV",
  "BatchInfo": "{\"CompanyMV\":{\"SortOrder\":\"A\",\"OrderBy\":\"CompanyName\",
\"NoOfRecordsToFetch\":5}}"
}
```

Response:

```
 {
  "Attributes": {
    "QueueName": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 45,
      "IsView": "N",
      "Value": "NestedComplex_Start Event_1",
      "Name": "QueueName"
    },
    "QueueType": {
      "Length": 255,
      "Type": "3210",
```

```
      "VarFieldId": 0,
      "VariableId": 44,
      "IsView": "N",
      "Value": "I",
      "Name": "QueueType"
    },
    "CompanyMV": [
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "5",
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Accenture",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1.30",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "3",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
```

```
            "IsView": "N",
            "Value": "201305",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Bangalore",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "2020-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "5",
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "BSNL",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
```

```
            "Value": "1.20",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "7",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "112233",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Agra",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1964-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "5",
```

```
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "HCL",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "2.20",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "9",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "220023",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Maharashtra",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
```

```
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1998-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "5",
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Infogain",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1.30",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "2",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
```

```
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "110087",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Delhi",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "2021-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "5",
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Infosys",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
```

```
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1.90",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "5",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "102033",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Bangalore",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1977-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      }
    ]
  },
  "RetrievedCount": 44,
  "Count": 44,
```

```
  "CorrelationId": "5d36e8cd-97b2-488e-9a1c-76b8e7ea4e63"
}
```

Batching and sorting (fetching next five records sorted using company name)

Request:

```
{
  "NoOfRecordsToFetch": 5,
  "ProcessDefId": 1021,
  "ActivityID": 1,
  "WorkItemId": 1,
  "ProcessInstanceId": "Nested-0000000001-process",
  "AttributeName": "CompanyMV",
  "BatchInfo": "{\"CompanyMV\":{\"SortOrder\":\"A\",\"OrderBy\":\"CompanyName\",
\"NoOfRecordsToFetch\":5,\"LastValue\":\"Infosys\",
\"LastInsertionOrderIdValue\":5}}"
}
```

Response:

```
{
  "Attributes": {
    "QueueName": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 45,
      "IsView": "N",
      "Value": "NestedComplex_Start Event_1",
      "Name": "QueueName"
    },
    "QueueType": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 44,
      "IsView": "N",
      "Value": "I",
      "Name": "QueueType"
    },
    "CompanyMV": [
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "6",
```

```
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "LIC",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "0.80",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "10",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "112234",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Lonavala",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
```

```
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1983-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "6",
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Newgen",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1.20",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
```

```
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "201301",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Noida",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "2022-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "6",
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Newgen",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
```

```
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1.20",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "8",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "201301",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Mumbai",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1974-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
```

```
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "6",
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "TCS",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1.40",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "4",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "220331",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Hyderabad",
            "Name": "BranchLocation"
```

```
        },
        "DateOfOpening": {
          "Length": 8,
          "Type": "238",
          "VarFieldId": 3,
          "VariableId": 58,
          "IsView": "N",
          "Value": "1955-01-01 00:00:00",
          "Name": "DateOfOpening"
        }
      },
      "Name": "CompanyMV"
    },
    {
      "Length": 0,
      "Type": "2311",
      "VarFieldId": 1,
      "VariableId": 58,
      "IsView": "N",
      "Value": "",
      "InsertionOrderId": "6",
      "MaxInsertionOrderId": "10",
      "MinInsertionOrderId": "1",
      "Child": {
        "CompanyName": {
          "Length": 50,
          "Type": "2310",
          "VarFieldId": 14,
          "VariableId": 58,
          "IsView": "N",
          "Value": "Wipro",
          "Name": "CompanyName"
        },
        "BranchIndex": {
          "Length": 15,
          "Type": "236",
          "VarFieldId": 4,
          "VariableId": 58,
          "IsView": "N",
          "Value": "1.20",
          "Precision": 2,
          "Name": "BranchIndex"
        },
        "CompanyId": {
          "Length": 2,
          "Type": "233",
          "VarFieldId": 2,
          "VariableId": 58,
          "IsView": "N",
          "Value": "6",
          "Name": "CompanyId"
```

```
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "201301",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Noida",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1972-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      }
    ]
  },
  "RetrievedCount": 44,
  "Count": 44,
  "CorrelationId": "515fb876-6d36-42e5-a589-36754fb51a26"
}
```


Batching and sorting (fetching previous five records sorted using company name)

Request:


```
{
  "NoOfRecordsToFetch": 5,
  "ProcessDefId": 1021,
  "ActivityID": 1,
  "WorkItemId": 1,
  "ProcessInstanceId": "Nested-0000000001-process",
  "AttributeName": "CompanyMV",
  "BatchInfo": "{\"CompanyMV\":{\"SortOrder\":\"D\",\"OrderBy\":\"CompanyName\",
```

```
\"NoOfRecordsToFetch\":5,\"LastValue\":\"Wipro\",
\"LastInsertionOrderIdValue\":5}}"
}
```

Response:


```
{
  "Attributes": {
    "QueueName": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 45,
      "IsView": "N",
      "Value": "NestedComplex_Start Event_1",
      "Name": "QueueName"
    },
    "QueueType": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 44,
      "IsView": "N",
      "Value": "I",
      "Name": "QueueType"
    },
    "CompanyMV": [
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "5",
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "TCS",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
```

```
            "VariableId": 58,
            "IsView": "N",
            "Value": "1.40",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "4",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "220331",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Hyderabad",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1955-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
```

```
        "Value": "",
        "InsertionOrderId": "5",
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Newgen",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1.20",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "8",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "201301",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Mumbai",
            "Name": "BranchLocation"
          },
```

```
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1974-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "5",
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Newgen",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1.20",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1",
            "Name": "CompanyId"
          },
```

```
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "201301",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Noida",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "2022-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "5",
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "LIC",
            "Name": "CompanyName"
          },
          "BranchIndex": {
```

```
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "0.80",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "10",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "112234",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Lonavala",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1983-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
```

```
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "5",
        "MaxInsertionOrderId": "10",
        "MinInsertionOrderId": "1",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Infosys",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1.90",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "5",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "102033",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
```

```
            "Value": "Bangalore",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1977-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      }
    ]
  },
  "RetrievedCount": 44,
  "Count": 44,
  "CorrelationId": "a660bf9e-f7eb-4a5b-b031-a4fe6d20ee79"
}
```

Fetching attributes with searching

The following are the sample with searching scenarios:

- Search by attribute name
- Search by attribute name (company name) and sorted by Date of opening

Sample for searching by searching name

Request:

```
{
  "NoOfRecordsToFetch": 5,
  "ProcessDefId": 1021,
  "ActivityID": 1,
  "WorkItemId": 1,
  "ProcessInstanceId": "Nested-0000000001-process",
  "AttributeName": "CompanyMV",
  "BatchInfo": "{\"CompanyMV\":{\"FilterXML\":{\"Filter\":[{\"VarName\":
\"CompanyName\",\"VarValue\":\"*wipro*\",\"LogicalOperator\":0,\"Operator\":7}]},
\"SortOrder\":\"A\",\"NoOfRecordsToFetch\":5}}"
}
```

Response:

```
{
  "Attributes": {
    "QueueName": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 45,
      "IsView": "N",
      "Value": "NestedComplex_Start Event_1",
      "Name": "QueueName"
    },
    "QueueType": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 44,
      "IsView": "N",
      "Value": "I",
      "Name": "QueueType"
    },
```

```
"CompanyMV": [
  {
    "Length": 0,
    "Type": "2311",
    "VarFieldId": 1,
    "VariableId": 58,
    "IsView": "N",
    "Value": "",
    "InsertionOrderId": "6",
    "MaxInsertionOrderId": "6",
    "MinInsertionOrderId": "6",
    "Child": {
      "CompanyName": {
        "Length": 50,
        "Type": "2310",
        "VarFieldId": 14,
        "VariableId": 58,
        "IsView": "N",
        "Value": "Wipro",
        "Name": "CompanyName"
      },
      "BranchIndex": {
        "Length": 15,
        "Type": "236",
        "VarFieldId": 4,
        "VariableId": 58,
        "IsView": "N",
        "Value": "1.20",
        "Precision": 2,
        "Name": "BranchIndex"
      },
      "CompanyId": {
        "Length": 2,
        "Type": "233",
        "VarFieldId": 2,
        "VariableId": 58,
        "IsView": "N",
        "Value": "6",
        "Name": "CompanyId"
      },
      "BranchPin": {
        "Length": 4,
        "Type": "234",
        "VarFieldId": 6,
        "VariableId": 58,
        "IsView": "N",
        "Value": "201301",
        "Name": "BranchPin"
      },
      "BranchLocation": {
        "Length": 50,
```

```
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "Noida",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1972-01-01 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      }
    ]
  },
  "RetrievedCount": 44,
  "Count": 44,
  "CorrelationId": "14f6c43c-e234-48c1-9232-9de9da1011fb"
}
```


Sample for searching (search by attribute name (company name) and sorted date of opening

Request:


```
{
  "NoOfRecordsToFetch": 5,
  "ProcessDefId": 1022,
  "ActivityID": 1,
  "WorkItemId": 1,
  "ProcessInstanceId": "Sort-0000000000003-process",
  "AttributeName": "CompanyMV",
  "BatchInfo": "{\"CompanyMV\":{\"FilterXML\":{\"Filter\":[{\"VarName\":
\"CompanyName\",\"VarValue\":\"*np*\",\"LogicalOperator\":0,\"Operator\":7}]},
\"SortOrder\":\"A\",\"OrderBy\":\"DateOfOpening\",\"NoOfRecordsToFetch\":5}}"
}
```


Response:


```
 {
  "Attributes": {
    "QueueName": {
      "Length": 255,
```

```
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 45,
      "IsView": "N",
      "Value": "NestedComplexNew_Start Event_1",
      "Name": "QueueName"
    },
    "QueueType": {
      "Length": 255,
      "Type": "3210",
      "VarFieldId": 0,
      "VariableId": 44,
      "IsView": "N",
      "Value": "I",
      "Name": "QueueType"
    },
    "CompanyMV": [
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "39",
        "MaxInsertionOrderId": "40",
        "MinInsertionOrderId": "30",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "npcl",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "6634.34",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
```

```
            "VariableId": 58,
            "IsView": "N",
            "Value": "6364",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "8364",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "dantewada",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1969-07-16 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "39",
        "MaxInsertionOrderId": "40",
        "MinInsertionOrderId": "30",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
```

```
            "IsView": "N",
            "Value": "np",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "1234.23",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "345",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "23453",
            "Name": "BranchPin"
          },
          "BranchLocation": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "delhi",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "2018-05-14 00:00:00",
            "Name": "DateOfOpening"
          }
        },
```

```
        "Name": "CompanyMV"
      },
      {
        "Length": 0,
        "Type": "2311",
        "VarFieldId": 1,
        "VariableId": 58,
        "IsView": "N",
        "Value": "",
        "InsertionOrderId": "39",
        "MaxInsertionOrderId": "40",
        "MinInsertionOrderId": "30",
        "Child": {
          "CompanyName": {
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 14,
            "VariableId": 58,
            "IsView": "N",
            "Value": "npc",
            "Name": "CompanyName"
          },
          "BranchIndex": {
            "Length": 15,
            "Type": "236",
            "VarFieldId": 4,
            "VariableId": 58,
            "IsView": "N",
            "Value": "2345.23",
            "Precision": 2,
            "Name": "BranchIndex"
          },
          "CompanyId": {
            "Length": 2,
            "Type": "233",
            "VarFieldId": 2,
            "VariableId": 58,
            "IsView": "N",
            "Value": "234",
            "Name": "CompanyId"
          },
          "BranchPin": {
            "Length": 4,
            "Type": "234",
            "VarFieldId": 6,
            "VariableId": 58,
            "IsView": "N",
            "Value": "23542",
            "Name": "BranchPin"
          },
          "BranchLocation": {
```

```
            "Length": 50,
            "Type": "2310",
            "VarFieldId": 5,
            "VariableId": 58,
            "IsView": "N",
            "Value": "dholpur",
            "Name": "BranchLocation"
          },
          "DateOfOpening": {
            "Length": 8,
            "Type": "238",
            "VarFieldId": 3,
            "VariableId": 58,
            "IsView": "N",
            "Value": "2022-10-17 00:00:00",
            "Name": "DateOfOpening"
          }
        },
        "Name": "CompanyMV"
      }
    ]
  },
  "RetrievedCount": 44,
  "Count": 44,
  "CorrelationId": "8b663cdc-43b7-4676-b6bd-f6957ea994d1"
}
```


InvokeWebService

The *InvokeWebService* API allows you to invoke the RESTful and SOAP web services.

- Service URL — *http://<GatewayIp: GatewayPort >/web-service-context/webservice/invoke*
  Where, GatewayIp is the IP of the machine, and GatewayPort is the port of the machine where the workitem service is deployed.
- Request Media Type —application/json
- Response Media Type — application/json
- Method Type — POST

Header parameters description

The following table describes the header parameters of the *InvokeWebService* API, if OAuth is disabled:

Name	Parameter Scope	Type	Description	Mandatory
Accept-Language	HeaderParam	Locale	Represents the natural language and locale that the client prefers. Example, en-US.	No
Tenant-Id	HeaderParam	String	Name of the cabinet.	Yes
Authorization	HeaderParam	String	The session ID of the signed-in user.	Yes

If OAuth is enabled:

Name	Parameter scope	Type	Description	Mandatory
AppId	HeaderParam	String	Registered AppId of the product	No
Access Token	HeaderParam	String	AccessToken for the signed-in user	No
SessionValid	HeaderParam	String	SessionValid for the signed-in user	No

Request body parameters description

The following table describes the request body parameters of the *InvokeWebService* API:

Name	Parent Tag	Type	Description	Default Value	Valid Value
ISRestService	Root	String	Indicates whether the called web-service is a REST or SOAP service.	NA	• Y – REST • N – SOAP

Name	Parent Tag	Type	Description	Default Value	Valid Value
ISIFORM	Root	String	Indicates when the web-service invoker is called from the portal application. This is a mandatory parameter.	Y	Y—Portal Application
ExtMethod Index	Root	Integer	The ID of the external method for web service generated in NewgenONE. This can be obtained from ExtMethodDefTable.	NA	• 34 – For SOAP service • Method Index – For REST service
ResourceURL	Root	String	Complete URL of the REST service resource.	NA	NA
RequestMedia Type	Root	String	Indicates the format type of request data.	N	• J- JSON • X- XML • P- Plaintext • N- None
Response MediaType	Root	String	Indicates the format type of response data	N	• J- JSON • X- XML • P- Plaintext • N- None
Operation Type	Root	String	Indicates the HTTP method for the requested webservice.	NA	• POST • PUT • GET • DELETE
Authentication Type	Root	String	Authentication type required through webservice for secure access.	NA	• B- Basic Authentication • NA- No Authentication
Authentication Details	Root	String	Pass an XML containing authentication details such as UserName and Password. For example, <Authentication Details><UserName> badmin</ UserName><Password>System882#</ Password></Authentication Details>	NA	

Name	Parent Tag	Type	Description	Default Value	Valid Value
MethodName	Root	String	Method Name that gets exposed by the web service for invocation. For SOAP services, this tag is mandatory and must hold the value of the method name used in the catalog definition. For REST services, this tag is non-mandatory.	NA	
Invocation Type	Root	String	Indicates if the webservice type is REST or SOAP. This is a mandatory parameter.	NA	• R- REST • S- SOAP
TimeOut Interval	Root	Integer	Defines the connection timeout and read timeout for SOAP webservice invocation.	NA	
BRMSREST	Root	String	Indicates in case the requested webservice is a BRMS REST service.	NA	• Y- Yes • N- No
InParams	Root	Map <String, WFIn Params**>	Map containing the value of Input parameters accepted through the webservice and their values along with parameter properties such as "paramType", "paramScope", "isArray" as shown in the sample request. For RESTful Services, you need to add a 'Child' tag in nested structures to signify the child element for the parent structure.	NA	
paramScope	InParams	String	Indicates where to send the InParam element.	NA	• C- Request Body • H- HeaderParam • Q- QueryParam • P- PathVariable
retryCount	Root	Integer	A number of times webservice execution gets retired in case of failure.	0	

Name	Parent Tag	Type	Description	Default Value	Valid Value
WSDL Location	Root	String	A complete WSDL address of the SOAP service resource. This is not applicable to RESTful applications.	NA	
ServiceName	Root	String	Name of the webservice which gets invoked. This is a mandatory parameter.	NA	
OutParams	Root	TreeMap <String, WFOutputParam>	A map contains the definition of output parameters accepted by the webservice and their mapping with columns from DMD tables. For both SOAP and REST services, you need to add a "child" tag in nested structures to signify the child element for the parent structure.	NA	
isArray	WFOutputParam*	Boolean	Indicates when this OutParam element is an array type.	False	• True • False
isMapped	WFOutputParam	Boolean	Indicates when this outparam element is mapped with a column of the DMD table.	False	• True • False
value	WFOutputParam*	String	Name of the DMD table column to which the OutParam is mapped.	NA	

WFOutParam* is a logical JSON entity containing the definition for output parameters such as paramType, isMapped, isArray, child, and the value containing the DMD table mapped fields. For more information, refer to the above-explained parameters. WFInParams** is a logical JSON entity containing definitions for input parameters such as paramType, isMapped, paramScope, isArray, child, and value. For more information, refer to the above-explained parameters.

Response body parameters description

The following table describes the response body parameters of the *InvokeWebService* API:

Name	Parent Tag	Parameter Type	Description
ResponseCode	Root	Integer	Returned the HTTP response status code.

Name	Parent Tag	Parameter Type	Description
ResponseDescription	Root	String	Description of a response.
Response	Root	Map<String,Object>	Response comprising the output of Web-service that contains the catalog variables and their corresponding values.

Error codes and description

The following table describes the error codes that can occur while executing the *InvokeWebService* API:

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_400	500	Internal Server Error	Operation Failed.	Recheck the request body.
NGONE_WFS_2001	400	Bad Request	Required request header missing.	Header information is missing. Recheck the header.
NGONE_WFS_415	415	Unsupported Media Type	Content type not supported.	The response media type passed in the request is incorrect. Recheck the web service definition.
NGONE_WFS_841	404	Not Found	SOAP WSDL not found.	WSDL URL is not found. Recheck the URL from a web browser.
NGONE_WFS_842	502	Bad Gateway	Proxy access denied.	Network access to webservice is denied. Recheck the network settings.
NGONE_WFS_843	404	Not Found	Error in Parsing a request.	Internal error occurred in parsing a request. It is due to the temporary problem with the external web service.
N/A	401	Unauthorized	Invalid Session.	Regenerate the session and pass the valid session Id in the authorization header.

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_801	500	Internal Server Error	SQL exception occurred.	An SQL exception occurred which stopped the operation. Refer to the Error logs for further analysis.
NGONE_WFS_503	503	Temporary	Service is either down or slow.	Service is either unavailable or slow.

Sample request and response

Request:

```
{
  "MappingFileFlag": "N",
  "debug": "false",
  "InvocationType": "R",
  "ProxySettings": {
    "ProxyEnabled": "false"
  },
  "ResponseMediaType": "J",
  "IsRestService": "Y",
  "AppServerInfo": {
    "JNDIServerPort": "8080",
    "EngineName": "ibps_maaz",
    "JNDIServerName": "127.0.0.1"
  },
  "ExtMethodIndex": "2003",
  "TimeOutInterval": "0",
  "ServiceName": "updateEmployeesGlobal",
  "InParams": {
    "employeeWrapper": {
      "paramType": "11",
      "paramScope": "C",
      "isArray": false,
      "child": {
        "employeeDept": {
          "paramType": "10",
          "paramScope": "C",
          "isArray": false,
          "value": "Microservice"
        },
        "age": {
          "paramType": "3",
          "paramScope": "C",
          "isArray": false,
```

```
          "value": "26"
        }
      }
    }
  },
  "OperationType": "POST",
  "MethodName": "updateEmployeesGlobal",
  "WSDLLocation": "WSDL",
  "RequestMediaType": "J",
  "ResourceURL": "http://localhost:8080/ws-invoker-rest/webservice/rest/update-
employees",
  "WorkitemId": "",
  "OutParams": {
    "employeeWrapper": {
      "paramType": "11",
      "isArray": false,
      "complexAttribute": "Part2Table",
      "child": {
        "employeeDept": {
          "paramType": "10",
          "isMapped": "true",
          "isArray": false,
          "value": "Part2Table.Field1"
        },
        "employees": {
          "paramType": "11",
          "isArray": true,
          "complexAttribute": "Part2Table.Part2ChildGrid",
          "child": {
            "employeeName": {
              "paramType": "10",
              "isMapped": "true",
              "isArray": false,
              "value": "Part2Table.Part2ChildGrid.Grid1"
            },
            "employeeId": {
              "paramType": "3",
              "isMapped": "true",
              "isArray": false,
              "value": "Part2Table.Part2ChildGrid.Grid2"
            },
            "location": {
              "paramType": "11",
              "isArray": true,
              "complexAttribute":
"Part2Table.Part2ChildGrid.Part2ChildNestedGrid",
              "child": {
                "country": {
                  "paramType": "10",
                  "isMapped": "true",
                  "isArray": false,
```

```
                  "value": "Part2Table.Part2ChildGrid.Part2ChildNestedGrid.Nest1"
                },
                "pin": {
                  "paramType": "4",
                  "isMapped": "true",
                  "isArray": false,
                  "value": "Part2Table.Part2ChildGrid.Part2ChildNestedGrid.Nest3"
                },
                "city": {
                  "paramType": "10",
                  "isMapped": "false",
                  "isArray": false
                },
                "state": {
                  "paramType": "10",
                  "isMapped": "false",
                  "isArray": false
                }
              }
            }
          }
        },
        "age": {
          "paramType": "3",
          "isMapped": "true",
          "isArray": false,
          "value": "Part2Table.Field2"
        }
      }
    }
  },
  "AuthenticationType": "NA"
}
```


Response:


```
{
  "soapMessageResponse": null,
  "RequestInput": null,
  "ResponseCode": 200,
  "ResponseDescription": "Ok : The request has succeeded",
  "ResponseOutput": null,
  "Response": {
    "UserDefVarFlag": "Y",
    "ActivityId": null,
    "Attributes": {
      "employeeWrapper": {
        "employeeDept": "Genesis",
        "employees": [
          {
            "employeeName": "Phil",
```

```
            "employeeId": 1,
            "location": [
              {
                "country": "U.S.A.",
                "pin": 116517,
                "city": "L.A.",
                "state": "California"
              },
              {
                "country": "U.S.A.",
                "pin": 116518,
                "city": "Tempe",
                "state": "Arizona"
              }
            ]
          },
          {
            "employeeName": "George",
            "employeeId": 2,
            "location": [
              {
                "country": "Germany",
                "pin": 110017,
                "city": "Dresden",
                "state": "Dresden"
              },
              {
                "country": "Germany",
                "pin": 110018,
                "city": "Berlin",
                "state": "Berlin"
              }
            ]
          }
        ],
        "age": 32
      },
      "Fault": 200,
      "FaultDesc": "Ok : The request has succeeded"
    },
    "OmniService": "Y",
    "ProcessInstanceId": "",
    "WorkitemId": null
  }
}
```

WFSearchWorkitemList

This API allows you to search workitems based on specified criteria. It allows you to search for workitems across various versions of a process or processes. In addition, allowing you to perform quick search and advanced search operations.

- URL — *http://<GatewayIp:GatewayPort >/workitem-service-context/workitem/searchworkitems*
  Where, GatewayIp is the IP of the machine, and GatewayPort is the port of the machine where the workitem service is deployed.
- Request Media Type — application/json
- Response Media Type — application/json
- Method Type — POST

Header parameters description

The following table describes the header parameters of the *WFSearchWorkitemList* service API, if OAuth is disabled:

Name	Parameter Scope	Type	Description	Mandatory
Authorization	HeaderParam	String	The session ID of the signed-in user.	Yes
Accept-Language	HeaderParam	Locale	Represents the specific region or language. Example: en-US	No
Tenant-Id	HeaderParam	String	Cabinet name.	Yes

If OAuth is enabled:

Name	Parameter scope	Type	Description	Mandatory
AppId	HeaderParam	String	Registered AppId of the product	No
Access Token	HeaderParam	String	AccessToken for the signed-in user	No

Name	Parameter scope	Type	Description	Mandatory
SessionValid	HeaderParam	String	SessionValid for the signed-in user	No

Request body parameters description

The following table describes the request body parameters of the *WFSearchWorkitemList service* API:

Name	Parent Tag	Type	Description	Default Value	Valid Value
Enable Multilingual	Root	Character	Enables the multilingual results in the queue list output.	N	• Y-Enabled • N-Disabled
DataFlag	Root	Character	Indicates whether the search results defined in the process are returned with a response or not.	N	• Y - Search • results are returned • N - Search • results are • not returned
OpenMode	Root	String	Decides whether a user is a business admin or a user desktop user. If this flag value is PM then the EnableMultiLingual flag value is ignored.	WD	• WD -WebDesktop • PM - Process Manager

Name	Parent Tag	Type	Description	Default Value	Valid Value
ShowAll WorkItems Flag	Root	Character	Displays all the eligible workitems satisfying the search criteria including the parent copies of distributed and referred workitems.	Y	• Y - Yes • N – No
UserDefVar Flag	Root	Character	Pass 'N' in case of a quick search by the user. Else, pass 'Y' in all other cases.	N	• Y - All scenarios except for a quick search • N - Quick search scenario
BatchInfo	Root	BatchInfo	Parent tag to specify batching information to fetch the next or previous batch of workitems.	NA	NA

Name	Parent Tag	Type	Description	Default Value	Valid Value
OrderBy	BatchInfo	Integer	An integer value indicates the order by field based on which the returned workitem list gets sorted. The returned list of workitems can be sorted on fields like Priority, Activity Name, Locked By, and so on.	2	• 1 - Priority Level • 2 - Process InstanceId • 3 - ActivityName • 4 - LockedBy Name • 5 - Introduced By • 6 - Instrument Status • 7 -CheckList Complete Flag • 8 - Lock Status • 9 - Workitem State • 10 - Entry DateTime • 11 - ValidTill • 12 - Locked Time • 13 – Introduction DateTime • 14 – Queue Name • 15 – Process Name • 16 – Assigned User • 17 - Status • 18 - Created DateTime

Name	Parent Tag	Type	Description	Default Value	Valid Value
SortOrder	BatchInfo	Character	Sorts the workitems in ascending or descending order.	A	• A - Ascending • D - Descending
LastValue	BatchInfo	Object	Value of the Order By field of the last record retrieved in the previous batch.	NA	NA
Process InstanceId	BatchInfo	String	The Process Instance ID of the last fetched workitem from the previous batch.	NA	NA
WorkItemId	BatchInfo	Integer	The Workitem ID of the last fetched workitem from the previous batch.	NA	NA
NoOfRecords ToFetch	BatchInfo	Integer	Number of records to fetch per batch.	Server Batch Size	NA
ReferredBy	Root	Integer	The user name of the currently referred workitem.	-1	NA
Filter	Root	Filter	Contains the parameters that can be used as filtering criteria for searching workitems.	-1	NA
Process DefinitionId	Filter	Integer	Indicates the Process definition ID for which you need to perform the search operation.	0	NA

Name	Parent Tag	Type	Description	Default Value	Valid Value
SearchOn Previous Version	Filter	Character	Flag Y indicates whether the search results must include workitems from previous versions of the process in case search operation is performed using Process name.	N	• Y - Search results must include workitems from previous versions of the process. • N - Search for the latest version
SortOn Version	Filter	Character	Indicates when to sort the workitems based on the versions.	NA	• Y – Sorted Workitems based on versions in an ascending order • N - N or in case no value is specified then the workitems are sorted based on versions in a descending order

Name	Parent Tag	Type	Description	Default Value	Valid Value
Process Name	Filter	String	Indicates the process name of the fetched workitems. It is also used to search across all versions of the specified process. If you want to search only on the current version of a process, then specify the process definition ID in the ProcessDefinitionId Tag.	NA	NA
State	Filter	Integer	Determines the search case required for execution.	2	• 1 - To search on Created DateTime • 2 - To search process data from transactional tables • 6 - To search history tables • 7 - To search on Introduction DateTime

Name	Parent Tag	Type	Description	Default Value	Valid Value
URNFlag	Filter	Character	Indicates whether the search operation is performed on URN and returns URN in the response.	N	• Y – To perform search on URN and to retrieve URN Name in response for each workitem • N - If URN is not required in response
URNName	Filter	String	For searching on URN Name, add * before and after the URN number. For example, *21*	NA	NA
Process Instance Name	Filter	String	For searching on Process Instance ID, you can search workitem by providing any of the following: • Actual Process Instance ID. For example, Account Opening 000000000000000001 • Using Asterisk character. For example, *21*	NA	NA

Name	Parent Tag	Type	Description	Default Value	Valid Value
LockedBy User	Filter	String	Using the user name, you can search for workitems that are currently locked by the given user.	NA	NA
IntroducedBy UserId	Filter	Integer	Using the user ID, you can search for the workitems initiated by the given user in the process.	NA	NA
ExcludeExit Workitems	Filter	Character	To exclude exited Workitems from the search list.	N	• Y - To exclude exit workitems • N – To include exit workitems
Archive Search	Filter	Character	Search for the workitems in the archival cabinet.	N	• Y - To search on archival cabinet • N - To search on active cabinet
Search Attributes	Filter	Search Attributes	Required for searching the workitems using the search variables.	NA	NA
Advanced Search	Search Attributes	Character	Indicates the advanced search option.	N	• Y - For using an advanced search feature • N - In cases other than advanced search
Conditions are applicable while searching using search variables and in an advanced search scenario.					

Name	Parent Tag	Type	Description	Default Value	Valid Value
Conditions	Search Attributes	List <Condition>	Conditions are applicable in performing search operation using search variables and in advanced search scenarios.	NA	NA
Operator	Conditions	Integer	The operator is used for preparing the condition.	NA	• * 1 for '<' • * 2 for '<=' • * 3 for '=' • * 4 for '!=' • * 5 for '>' • * 6 for '>=' • * 7 for 'Like' • * 8 for 'Not Like' • * 9 for 'Null' • * 10 for 'Not Null'
Join Condition	Conditions	String	Join condition in case the search criteria comprise multiple parameters.	NA	AND or OR
Field	Conditions	Field	Refers to the search variable, its details, and the value to apply to filter the workitems.	NA	NA
Open Bracket	Conditions	Boolean	Open bracket to group multiple conditions together that is, mainly while using 'Or' join conditions.	NA	True or False

Name	Parent Tag	Type	Description	Default Value	Valid Value
Close Bracket	Conditions	Boolean	Closing bracket signifies the end of grouped conditions.	NA	True or False
Name	Field	String	Name of the search variable to filter the workitems.	NA	NA
Value	Field	Object	Value of the search variable basis which workitems are filtered.	NA	NA
Type	Field	Integer	Type of the search variable basis which workitems are filtered.	NA	• * 3 for Integer • * 4 for Long • * 6 for float • * 8 for Date • * 10 for String • * 12 for Boolean • * 15 for Short • Date • * 16 for Time • * 17 for Duration • * 18 for NText
Process VariantId	Filter	Integer	ProcessVariantID of the process.	0	NA
QueueId	Filter	Integer	Using the queue ID, you can search for the workitems present in the given queue.	0	NA

Name	Parent Tag	Type	Description	Default Value	Valid Value
Queue Name	Filter	String	Using the queue name, you can search for workitems present in the following queue. You can either specify the Queue ID in QueueID Tag or the Name in this tag.	NA	NA
ActivityId	Filter	Integer	Using the Activity ID, you can search for workitems that are present in the given activity.	0	NA
ActivityName	Filter	String	Using the Activity Name, you can search for the workitems that are present in the given activity.	NA	NA
QueryList is applicable in Quick Search scenarios.					
QueryList	Filter	List <QueryInfo>	Contains the variable details or value for quick search case on the filtered workitems.	NA	NA
VarName	QueryList	String	Name of the quick search variable for filtering workitems.	NA	NA

Name	Parent Tag	Type	Description	Default Value	Valid Value
VarType	QueryList	Integer	The data type of the quick search variable for filtering workitems.	NA	• * 3 for Integer • * 4 for Long • * 6 for float • * 8 for Date • * 10 for String • * 12 for Boolean • * 15 for Short Date • * 16 for Time • * 17 for Duration • * 18 for NText
Length	QueryList	Integer	Column length of the mapped quick search variable for filtering workitems.	NA	NA
Operator	QueryList	Integer	Operator to define the condition.	NA	• * 1 for '<' • * 2 for '<=' • * 3 for '=' • * 4 for '!=' • * 5 for '>' • * 6 for '>=' • * 7 for 'Like' • * 8 for 'Not Like' • * 9 for 'Null' • * 10 for 'Not Null'
VarValue	QueryList	Object	Value of the quick search variable basis which workitems are filtered.	NA	NA

Name	Parent Tag	Type	Description	Default Value	Valid Value
Join Condition	QueryList	String	Join condition used in case the search criteria comprise the multiple parameters.	NA	NA
ProcessInstance State	Filter	Integer	Represents the state of workitem for the entire process, it is 1 for workitems that are not introduced, 2 for workitems that are introduced but have not exited yet, and 6 for completed workitems.	0	• 1 - Not Started • 2 - Running • 3 - Suspended • 4 - Terminated • 5 - Aborted • 6 - Completed
HoldType	Filter	String	Passed in the filter to retrieve the workitems which are either on a Hold Activity or are temporarily on hold on a user workstep.	N	• N - No hold condition • A - Search workitmes available on Hold Activity • T - Search only Temporary hold workitmes • B - Search all workitems which are on Hold Activity, or which are on Temporarily Hold

Name	Parent Tag	Type	Description	Default Value	Valid Value
Priority	Filter	Integer	To filter the workitems in search results based on their priority levels.	0	• 1- Low • 2 -Medium • 3 -High • 4 -Very High
DelayFlag	Filter	Character	To fetch the delayed workitems.	B	• Y - Fetch workitems currently in Delay • N - Fetch workitems are not in Delay • B - Fetch all workitems irrespective of the delay attribute
Fixed Assigned ToUser	Filter	String	Using the user name, you can search for workitems that are fixed assigned to a given user in the process.	NA	NA
System Assigned ToUser	Filter	String	Using the user name, you can search for workitems that are temporarily assigned to a given user in the process.	NA	NA
StateDate Range	Filter	StateDate Range	The date range corresponds to the state specified in the State tag.	NA	NA

Name	Parent Tag	Type	Description	Default Value	Valid Value
StateDate RangeStart Date	StateDate Range	Date	Start date of the date range corresponding to the state specified in the State tag.	NA	NA
StateDate RangeEnd Date	StateDate Range	Date	End date of the date range corresponding to the state specified in the State tag.	NA	NA
ValidTill DateRange	Filter	ValidTillDate Range	Date Range to fetch valid workitems on the activity.	NA	NA
ValidTillRange StartDate	ValidTillDate Range	Date	Start date of the date range to fetch the valid workitems on the activity.	NA	NA
ValidTillRange EndDate	ValidTillDate Range	Date	End date of the date range to fetch valid workitems on the current activity.	NA	NA
LockStatus	Filter	Character	Flag to filter the workitems in search results based on the current lock status.	B	• Y – To fetch locked workitems • N – To fetch unlocked workitems

Name	Parent Tag	Type	Description	Default Value	Valid Value
Exception Status	Filter	Character	Flag to filter the workitems in search results based on the exception status.	B	• E - To fetch only the workitems on which any exception is raised • N - To fetch the workitems irrespective of exception status
For searching using Criteria Management feature.					
IsCriteria ReportCase	Root	Character	Flag indicating to search for workitems using the filters defined in criteria created from 'Criteria Management' feature.	N	Y or N
CriteriaId	Root	Integer	Id of the criteria using which the workitems are searched.	NA	NA
FilterId	Root	Integer	Id of the filter that is, defined in criteria using which the workitems are searched.	NA	NA
GetWICount	Root	Character	Flag indicating to retrieve count of workitems satisfying each filter defined under given criteria.	N	Y or N

Name	Parent Tag	Type	Description	Default Value	Valid Value
IsCount Enabled	Root	Character	Flag to retrieve total count of workitems satisfying the search conditions or criteria.	N	Y or N

Response body parameters description

The following table describes the response body parameters of the *WFSearchWorkitemList* service API:

Name	Parent Tag	Type	Description
TotalCount	Root	Integer	It's the total number of present workitems. In case the present workitems number is greater than the batch size so batch size plus one is returned.
RetrievedCount	Root	Integer	It's the total number of retrieved workitems. If a count is greater than the batch size so workitems equivalent to batch size is returned.
QueryQueueId	Root	Integer	Signifies the queue ID of the search queue that is, the query queue.
WorkItemList	Root	List<WorkItemInfo>	Details of the workitems retrieved as they satisfied the search criteria.
URN	WorkItemList	String	Represents the display name of the workitem.
RegistrationNo	WorkItemList	String	Name or Process Instance ID of the Workitem.

Name	Parent Tag	Type	Description
QueueName	WorkItemList	String	Name of the Queue containing the workitem currently.
ProcessName	WorkItemList	String	Name of the Process that contains the workitem.
ProcessVersionNo	WorkItemList	Integer	Version No. of the Process in which the workitem is present.
ActivityName	WorkItemList	String	Activity Name of the current workitem.
WorkItemState	WorkItemList	String	The name of the Workitem State is as follows: • Not Started • Running • Suspended • Terminated • Aborted • Completed
CheckListStatus	WorkItemList	Character	Indicates whether all the To Dos defined for the Process have been worked upon for the workitem. • Y – Checklist Complete • N – Pending
AssignedTo	WorkItemList	String	Name of the user to whom the workitem is currently assigned.
AssignedToPersonalName	WorkItemList	String	Personal name of the user to whom the workitem is currently assigned.
AssignedToFamilyName	WorkItemList	String	Family name of the user to whom the workitem is currently assigned.
PendingSince	WorkItemList	Date	Signifies the datetime since the workitem was routed to the activity it is currently on.

Name	Parent Tag	Type	Description
ValidTill	WorkItemList	Date	Date and time validity of the workitem for current activity.
WorkItemId	WorkItemList	Integer	Workitem ID that is 1 for a parent workitem. It increases sequentially for the subsequent child workitems (if any).
PriorityLevel	WorkItemList	Integer	It is the priority level set for the workitem. • 1 – Low • 2 – Medium • 3 – High • 4 – Very High
ParentWorkItemId	WorkItemList	Integer	Workitem ID of the Parent Workitem, in case of distributed workitems.
ProcessDefinitionId	WorkItemList	Integer	The process ID of the workitem.
ActivityId	WorkItemList	Integer	Activity ID for the workitem.
InstrumentStatus	WorkItemList	Character	Exception Status of the workitem, • E – In Exception • N – Normal
LockStatus	WorkItemList	Character	Lock Status of the workitem, • Y – Locked • N – Not Locked
LockedByUserName	WorkItemList	String	User ID of the currently locked workitem.
LockedByPersonalName	WorkItemList	String	User Name of the currently locked workitem.
LockedByFamilyName	WorkItemList	String	The family name of the user for the currently locked workitem.
CreatedByUserName	WorkItemList	String	Name of the User who created the workitem.

Name	Parent Tag	Type	Description
CreationDateTime	WorkItemList	Date	DateTime on which the workitem is created.
LockedTime	WorkItemList	Date	DateTime on which the workitem is locked.
IntroductionDateTime	WorkItemList	Date	DateTime on which the workitem is initiated or introduced in the process.
IntroducedBy	WorkItemList	String	User ID of the user who introduced the workitem in the process.
IntroducedByPersonalName	WorkItemList	String	Name of the user who introduced the workitem in the process.
IntroducedByFamilyName	WorkItemList	String	Family name of the user who introduced the workitem in the process.
AssignmentType	WorkItemList	Character	Current Assignment Type for the workitem. • S – Shared • F – Fixed
ProcessInstanceState	WorkItemList	Integer	State of the Process Instance. • 1 - Not Started • 2 - Running • 3 - Suspended • 4 - Terminated • 5 - Aborted • 6 - Completed
QueueType	WorkItemList	Integer	Type of the Queue that the workitem currently belongs, like introduction, No Assignment or Dynamic.
Status	WorkItemList	String	The Value of the System-Modifiable Status Variable.
QueueId	WorkItemList	Integer	Queue ID of the workitem.
TurnAroundTime	WorkItemList	Date	Turn around time for the workitem.

Name	Parent Tag	Type	Description
ReferredBy	WorkItemList	String	User Name that refers to the workitem. • In case the workitem is currently referred.
ReferredTo	WorkItemList	String	User Name to whom the workitem is referred. • In case the workitem is currently referred.
TurnAroundDateTime	WorkItemList	Date	Expected workitem delay for the given workitem.
ProcessedBy	WorkItemList	String	User Id of the user who processed the workitem.
ProcessedByPersonalName	WorkItemList	String	Name of the user who processed the workitem.
ProcessedByFamilyName	WorkItemList	String	Family name of the user who processed the workitem.
ActivityType	WorkItemList	Integer	Activity type that signifies the type of workstep.
WorkItemStateId	WorkItemList	Integer	The ID of the State of the workitem, • 1 - Not Started • 2 - Running • 3 - Suspended • 4 - Terminated • 5 - Aborted • 6 - Completed
VAR_REC_1	WorkItemList	String	VAR_REC_1 stores the folder index of the folder corresponding to the workitem created in OmniDocs (DMS).
WorkItemCount	Root	Integer	Total count of workitems satisfying the search conditions or criteria.
LastModifiedOn	Root	Date	Last modified date and time for given criteria.

Name	Parent Tag	Type	Description
FilterInfo	Root	FilterInfo	Comprises of filter details and the count of workitems satisfying the given filters.
FilterId	FilterInfo	Integer	Filter ID defined in the criteria to retrieve the workitem count.
FilterAlias	FilterInfo	String	Name of the filter defined in the criteria using to retrieve the workitem count.
WICount	FilterInfo	Integer	Count of workitems satisfying the filter defined under a particular criteria.
EntityName	FilterInfo	Integer	Represents name of the criteria as per locale in case it's configured using the multilingual definition feature.

Error codes and description

The following table describes the error codes that can occur while executing the *WFSearchWorkitemList* service API:

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
N/A	401	Unauthorized	Invalid Session	Regenerate the session and pass the valid session Id in the authorization header.
NGONE_WFS_300	401	Unauthorized	User is not associated on the given QueueId	The user does not have rights on the queue under consideration, please give relevant rights to the user and try again.

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
3001	400	Bad Request	Invalid input parameters	Refer to the detailed error message to correct input parameters which have invalid values. You can refer to the below sample for reference (FieldName: ErrorDescription), 1. OpenMode - Provide a valid value for 'OpenMode' that is, either PM or WD. 2. OrderBy - Provide a valid value for 'OrderBy' field, it must be a positive integer.
NGONE_WFS_503	503	Temporary	Service is either down or slow	Service is either unavailable or slow.
NGONE_WFS_830	401	Unauthorized	No rights	Either user has no rights on the current object (process) or user is searching using the variable which is not declared as a search variable.
NGONE_WFS_1009	400	Bad Request	Invalid parameter	Criteria ID is invalid.
NGONE_WFS_803	401	Unauthorized	Invalid parameter	No rights on the current object (criteria).
NGONE_WFS_856	412	Precondition Failed	Archival cabinet is not set.	Verify whether the archival cabinet exist and configured correctly.

Sample request and response

Below is the sample request and response body for *WFSearchWorkitemList* API as per the following scenarios.

Searching workitems with two search variables

This section contains the request and response body to search workitems in a process using two search variables.

Request:

```
{
  "EnableMultilingual": "Y",
  "OpenMode": "WD",
  "IsCriteriaReportCase": "Y",
  "GetWICount": "Y",
  "CriteriaId": 2,
  "DataFlag": "Y",
  "BatchInfo": {
    "OrderBy": 2,
    "SortOrder": "D",
    "LastValue": null,
    "ProcessInstanceId": null,
    "WorkItemId": null,
    "NoOfRecordsToFetch": 20
  },
  "UserDefVarFlag": "Y",
  "ShowAllWorkItemsFlag": "Y",
  "IsCountEnabled": "N"
}
```

Response:

```
{
  "FilterInfo": [
    {
      "FilterId": 2,
      "FilterAlias": "First_Filter",
      "WICount": 12,
      "EntityName": null
    },
```

```
    {
      "FilterId": 3,
      "FilterAlias": "Second_Filter",
      "WICount": 2,
      "EntityName": null
    },
    {
      "FilterId": 4,
      "FilterAlias": "Third_Filter",
      "WICount": 2,
      "EntityName": null
    }
  ],
  "LastModifiedOn": "2022-12-30 12:16:22"
}
```

Searching workitems using quick search

This section contains the request and response body to search workitems using the quick search on a queue variable.

Request:

```
{
   "EnableMultilingual": "Y",
   "OpenMode": "PM",
   "Filter": {
       "ProcessName": "SearchWIList",
       "ProcessDefinitionId": 4,
       "SearchOnPreviousVersion": "Y",
       "URNFlag": "Y",
       "URNName": null,
       "QueryList": [
           {
               "VarName": "Q_Age",
               "VarType": 3,
               "Length": 1024,
               "Operator": 3,
               "VarValue": 26,
               "JoinCondition": "And"
           }
       ]
   },
   "DataFlag": "Y",
   "BatchInfo": {
       "OrderBy": 2,
       "SortOrder": "A",
```

```
        "LastValue": null,
        "ProcessInstanceId": null,
        "WorkItemId": null,
        "NoOfRecordsToFetch": 20
    },
    "UserDefVarFlag": "N",
    "ShowAllWorkItemsFlag": "Y"
    }
```

Response:


```
{
    "TotalCount": 1,
    "RetrievedCount": 1,
    "QueryQueueId": 14,
    "WorkItemList": [
        {
            "URN": "SearchTesting-1",
            "RegistrationNo": "SearchTesting-000000000000000001",
            "QueueName": "SearchWIList_Scanning",
            "ProcessName": "SearchWIList",
            "ProcessVersionNo": 1,
            "ActivityName": "Scanning",
            "WorkItemState": "RUNNING",
            "CheckListStatus": "N",
            "PendingSince": "2022-10-06 12:13:14.007",
            "WorkItemId": 1,
            "PriorityLevel": 1,
            "ProcessDefinitionId": 4,
            "ActivityId": 1,
            "InstrumentStatus": "N",
            "LockStatus": "N",
            "CreatedByUserName": "badmin",
            "CreationDateTime": "2022-10-06 12:13:14.007",
            "AssignmentType": "S",
            "ProcessInstanceState": 1,
            "QueueType": "I",
            "QueueId": 15,
            "ProcessedBy": "badmin",
            "ProcessedByPersonalName": "New User",
            "ActivityType": 1,
            "WorkItemStateId": 2,
            "VAR_REC_1": "78",
            "VAR_REC_2": "F",
            "QueueData": [
                {
                    "Name": "Q_Age",
                    "Value": 26
                },
                {
                    "Name": "Q_DateOfJoining",
```

```
                    "Value": "2022-09-12 00:00:00.0"
                },
                {
                    "Name": "Q_Id",
                    "Value": 10
                },
                {
                    "Name": "Q_Name",
                    "Value": "Jason"
                },
                {
                    "Name": "Q_Salary",
                    "Value": 15000.00
                },
                {
                    "Name": "Ext_Date_Of_Birth",
                    "Value": "1996-06-19 00:00:00.0"
                },
                {
                    "Name": "Ext_Type",
                    "Value": "Adult"
                }
            ]
        }
    ]
}
```

Searching workitems using quick search on URN

This section contains the request and response body to search workitems using the quick search feature on URN.

Request:

```
{
    "EnableMultilingual": "Y",
    "OpenMode": "PM",
    "Filter": {
        "ProcessInstanceName" : "SearchTesting-0000000000000001",
        "ProcessDefinitionId": 4,
        "URNFlag": "N",
        "AdvancedSearch": "N"
    },
    "DataFlag": "Y",
    "BatchInfo": {
```

```
        "OrderBy": 2,
        "SortOrder": "A",
        "LastValue": null,
        "ProcessInstanceId": null,
        "WorkItemId": null,
        "NoOfRecordsToFetch": 20
    },
    "UserDefVarFlag": "Y",
    "ShowAllWorkItemsFlag": "Y",
    }
```

Response:

```
{
    "TotalCount": 1,
    "RetrievedCount": 1,
    "QueryQueueId": 14,
    "WorkItemList": [
        {
            "URN": "SearchTesting-1",
            "RegistrationNo": "SearchTesting-000000000000000001",
            "QueueName": "SearchWIList_Scanning",
            "ProcessName": "SearchWIList",
            "ProcessVersionNo": 1,
            "ActivityName": "Scanning",
            "WorkItemState": "RUNNING",
            "CheckListStatus": "N",
            "PendingSince": "2022-10-06 12:13:14.007",
            "WorkItemId": 1,
            "PriorityLevel": 1,
            "ProcessDefinitionId": 4,
            "ActivityId": 1,
            "InstrumentStatus": "N",
            "LockStatus": "N",
            "CreatedByUserName": "badmin",
            "CreationDateTime": "2022-10-06 12:13:14.007",
            "AssignmentType": "S",
            "ProcessInstanceState": 1,
            "QueueType": "I",
            "QueueId": 15,
            "ProcessedBy": "badmin",
            "ProcessedByPersonalName": "New User",
            "ActivityType": 1,
            "WorkItemStateId": 2,
            "VAR_REC_1": "78",
            "VAR_REC_2": "F",
            "QueueData": [
                {
                    "Name": "Q_Age",
                    "Value": 26
                },
```

```
                {
                    "Name": "Q_DateOfJoining",
                    "Value": "2022-09-12 00:00:00.0"
                },
                {
                    "Name": "Q_Id",
                    "Value": 10
                },
                {
                    "Name": "Q_Name",
                    "Value": "Jason"
                },
                {
                    "Name": "Q_Salary",
                    "Value": 15000.00
                },
                {
                    "Name": "Ext_Date_Of_Birth",
                    "Value": "1996-06-19 00:00:00.0"
                },
                {
                    "Name": "Ext_Type",
                    "Value": "Adult"
                }
            ]
        }
    ]
}
```

Searching workitems with quick search on ProcessInstanceId

This section contains the request and response body to search workitems using the quick search on the ProcessInstanceID.

Request:

```
{
    "EnableMultilingual": "Y",
    "OpenMode": "PM",
    "Filter": {
        "ProcessInstanceName" : "SearchTesting-0000000000000001",
        "ProcessDefinitionId": 4,
        "URNFlag": "N",
        "AdvancedSearch": "N"
    },
```

```
    "DataFlag": "Y",
    "BatchInfo": {
        "OrderBy": 2,
        "SortOrder": "A",
        "LastValue": null,
        "ProcessInstanceId": null,
        "WorkItemId": null,
        "NoOfRecordsToFetch": 20
    },
    "UserDefVarFlag": "Y",
    "ShowAllWorkItemsFlag": "Y",
    }
```

Response:

```
{
    "TotalCount": 1,
    "RetrievedCount": 1,
    "QueryQueueId": 14,
    "WorkItemList": [
        {
            "URN": "SearchTesting-1",
            "RegistrationNo": "SearchTesting-000000000000000001",
            "QueueName": "SearchWIList_Scanning",
            "ProcessName": "SearchWIList",
            "ProcessVersionNo": 1,
            "ActivityName": "Scanning",
            "WorkItemState": "RUNNING",
            "CheckListStatus": "N",
            "PendingSince": "2022-10-06 12:13:14.007",
            "WorkItemId": 1,
            "PriorityLevel": 1,
            "ProcessDefinitionId": 4,
            "ActivityId": 1,
            "InstrumentStatus": "N",
            "LockStatus": "N",
            "CreatedByUserName": "badmin",
            "CreationDateTime": "2022-10-06 12:13:14.007",
            "AssignmentType": "S",
            "ProcessInstanceState": 1,
            "QueueType": "I",
            "QueueId": 15,
            "ProcessedBy": "badmin",
            "ProcessedByPersonalName": "New User",
            "ActivityType": 1,
            "WorkItemStateId": 2,
            "VAR_REC_1": "78",
            "VAR_REC_2": "F",
            "QueueData": [
                {
                    "Name": "Q_Age",
```

```
                    "Value": 26
                },
                {
                    "Name": "Q_DateOfJoining",
                    "Value": "2022-09-12 00:00:00.0"
                },
                {
                    "Name": "Q_Id",
                    "Value": 10
                },
                {
                    "Name": "Q_Name",
                    "Value": "Jason"
                },
                {
                    "Name": "Q_Salary",
                    "Value": 15000.00
                },
                {
                    "Name": "Ext_Date_Of_Birth",
                    "Value": "1996-06-19 00:00:00.0"
                },
                {
                    "Name": "Ext_Type",
                    "Value": "Adult"
                }
            ]
        }
    ]
}
```

Searching workitems using advanced search

Request:

```
{
    "EnableMultilingual": "Y",
    "OpenMode": "PM",
    "Filter": {
        "ProcessName": "SearchWIList",
        "LocalProcessName": "SearchWIList",
        "ProcessDefinitionId": 4,
        "ArchiveSearch": "N",
        "SearchOnPreviousVersion": "N",
        "SortOnVersion": "N",
        "URNFlag": "Y",
        "URNName": null,
```

```
        "ExcludeExitWorkitems": "N",
        "State": 2,
        "HoldType": "1",
        "LockedByUser": null,
        "LockedByUserId": 0,
        "AssignedToUser": null,
        "AssignedToUserId": 0,
        "IntroducedByUser": null,
        "IntroducedByUserId": 0,
        "SearchAttributes": {
            "EntrySetting": "Y",
            "AdvancedSearch": "Y",
            "Conditions": [
                {
                    "Operator": 6,
                    "JoinCondition": "And",
                    "Field": {
                        "Name": "Q_Age",
                        "Value": 22,
                        "Type": 3
                    }
                },
                {
                    "Operator": 2,
                    "JoinCondition": "And",
                    "Field": {
                        "Name": "Q_Age",
                        "Value": 26,
                        "Type": 3
                    }
                }
            ]
        }
    },
    "DataFlag": "Y",
    "BatchInfo": {
        "OrderBy": 2,
        "SortOrder": "A",
        "LastValue": null,
        "ProcessInstanceId": null,
        "WorkItemId": null,
        "NoOfRecordsToFetch": 20
    },
    "UserDefVarFlag": "Y",
    "ShowAllWorkItemsFlag": "Y"
    }
```

Response:


```
{
   "TotalCount": 3,
   "RetrievedCount": 3,
   "QueryQueueId": 14,
   "WorkItemList": [
       {
           "URN": "SearchTesting-1",
           "RegistrationNo": "SearchTesting-000000000000000001",
           "QueueName": "SearchWIList_Scanning",
           "ProcessName": "SearchWIList",
           "ProcessVersionNo": 1,
           "ActivityName": "Scanning",
           "WorkItemState": "RUNNING",
           "CheckListStatus": "N",
           "PendingSince": "2022-10-06 12:13:14.007",
           "WorkItemId": 1,
           "PriorityLevel": 1,
           "ProcessDefinitionId": 4,
           "ActivityId": 1,
           "InstrumentStatus": "N",
           "LockStatus": "N",
           "CreatedByUserName": "badmin",
           "CreationDateTime": "2022-10-06 12:13:14.007",
           "AssignmentType": "S",
           "ProcessInstanceState": 1,
           "QueueType": "I",
           "QueueId": 15,
           "ProcessedBy": "badmin",
           "ProcessedByPersonalName": "New User",
           "ActivityType": 1,
           "WorkItemStateId": 2,
           "VAR_REC_1": "78",
           "VAR_REC_2": "F",
           "QueueData": [
               {
                   "Name": "Q_Age",
                   "Value": 26
               },
               {
                   "Name": "Q_DateOfJoining",
                   "Value": "2022-09-12 00:00:00.0"
               },
               {
                   "Name": "Q_Id",
                   "Value": 10
               },
               {
                   "Name": "Q_Name",
                   "Value": "Jason"
```

```
                },
                {
                    "Name": "Q_Salary",
                    "Value": 15000.00
                },
                {
                    "Name": "Ext_Date_Of_Birth",
                    "Value": "1996-06-19 00:00:00.0"
                },
                {
                    "Name": "Ext_Type",
                    "Value": "Adult"
                }
            ]
        },
        {
            "URN": "SearchTesting-4",
            "RegistrationNo": "SearchTesting-000000000000000004",
            "QueueName": "SearchWIList_Scanning",
            "ProcessName": "SearchWIList",
            "ProcessVersionNo": 1,
            "ActivityName": "Scanning",
            "WorkItemState": "RUNNING",
            "CheckListStatus": "N",
            "PendingSince": "2022-10-20 02:33:34.753",
            "WorkItemId": 1,
            "PriorityLevel": 1,
            "ProcessDefinitionId": 4,
            "ActivityId": 1,
            "InstrumentStatus": "N",
            "LockStatus": "N",
            "CreatedByUserName": "badmin",
            "CreationDateTime": "2022-10-20 02:33:34.753",
            "AssignmentType": "S",
            "ProcessInstanceState": 1,
            "QueueType": "I",
            "QueueId": 15,
            "ProcessedBy": "badmin",
            "ProcessedByPersonalName": "New User",
            "ActivityType": 1,
            "WorkItemStateId": 2,
            "VAR_REC_1": "81",
            "VAR_REC_2": "F",
            "QueueData": [
                {
                    "Name": "Q_Age",
                    "Value": 23
                },
                {
                    "Name": "Q_DateOfJoining",
                    "Value": "2021-09-12 00:00:00.0"
```

```
                },
                {
                    "Name": "Q_Id",
                    "Value": 44
                },
                {
                    "Name": "Q_Name",
                    "Value": "Justin"
                },
                {
                    "Name": "Q_Salary",
                    "Value": 80000.00
                },
                {
                    "Name": "Ext_Date_Of_Birth",
                    "Value": "1989-06-01 00:00:00.0"
                },
                {
                    "Name": "Ext_Type",
                    "Value": "Adult"
                }
            ]
        }
    ]
}
```

Retrieving next batch of workitems

This section contains the request and response body to retrieve the next batch of workitems that satisfy the search criteria.

Request:

```
{
    "EnableMultilingual": "Y",
    "OpenMode": "PM",
    "Filter": {
        "ProcessName": "SearchWIList",
        "LocalProcessName": "SearchWIList",
        "SearchOnPreviousVersion": "N",
        "ProcessDefinitionId": 4,
        "SortOnVersion": "N",
        "URNFlag": "Y",
        "URNName": null,
        "ExcludeExitWorkitems": "N",
        "State": 1,
        "HoldType": "1",
```

```
        "LockedByUser": null,
        "LockedByUserId": 0,
        "AssignedToUser": null,
        "AssignedToUserId": 0,
        "IntroducedByUser": null,
        "IntroducedByUserId": 0,
        "SearchAttributes": {
            "AdvancedSearch": "N"
        }
    },
    "DataFlag": "Y",
    "BatchInfo": {
        "OrderBy": 2,
        "SortOrder": "A",
        "LastValue": "SearchTesting-000000000000000001",
        "ProcessInstanceId": "SearchTesting-000000000000000001",
        "WorkItemId": 1,
        "NoOfRecordsToFetch": 20
    },
    "UserDefVarFlag": "Y",
    "ShowAllWorkItemsFlag": "Y"
}
```


Response:


```
{
    "TotalCount": 2,
    "RetrievedCount": 2,
    "QueryQueueId": 14,
    "WorkItemList": [
        {
            "URN": "SearchTesting-2",
            "RegistrationNo": "SearchTesting-000000000000000002",
            "QueueName": "SearchWIList_Scanning",
            "ProcessName": "SearchWIList",
            "ProcessVersionNo": 1,
            "ActivityName": "Scanning",
            "WorkItemState": "RUNNING",
            "CheckListStatus": "N",
            "PendingSince": "2022-10-06 12:13:14.007",
            "WorkItemId": 1,
            "PriorityLevel": 1,
            "ProcessDefinitionId": 4,
            "ActivityId": 1,
            "InstrumentStatus": "N",
            "LockStatus": "N",
            "CreatedByUserName": "badmin",
            "CreationDateTime": "2022-10-06 12:13:14.007",
            "AssignmentType": "S",
            "ProcessInstanceState": 1,
            "QueueType": "I",
```

```
            "QueueId": 15,
            "ProcessedBy": "badmin",
            "ProcessedByPersonalName": "New User",
            "ActivityType": 1,
            "WorkItemStateId": 2,
            "VAR_REC_1": "78",
            "VAR_REC_2": "F",
            "QueueData": [
                {
                    "Name": "Q_Age",
                    "Value": 26
                },
                {
                    "Name": "Q_DateOfJoining",
                    "Value": "2022-09-12 00:00:00.0"
                },
                {
                    "Name": "Q_Id",
                    "Value": 10
                },
                {
                    "Name": "Q_Name",
                    "Value": "Jason"
                },
                {
                    "Name": "Q_Salary",
                    "Value": 15000.00
                },
                {
                    "Name": "Ext_Date_Of_Birth",
                    "Value": "1996-06-19 00:00:00.0"
                },
                {
                    "Name": "Ext_Type",
                    "Value": "Adult"
                }
            ]
        },
        {
            "URN": "SearchTesting-3",
            "RegistrationNo": "SearchTesting-000000000000000003",
            "QueueName": "SearchWIList_Scanning",
            "ProcessName": "SearchWIList",
            "ProcessVersionNo": 1,
            "ActivityName": "Scanning",
            "WorkItemState": "RUNNING",
            "CheckListStatus": "N",
            "PendingSince": "2022-10-20 02:33:34.753",
            "WorkItemId": 1,
            "PriorityLevel": 1,
            "ProcessDefinitionId": 4,
```

```
            "ActivityId": 1,
            "InstrumentStatus": "N",
            "LockStatus": "N",
            "CreatedByUserName": "badmin",
            "CreationDateTime": "2022-10-20 02:33:34.753",
            "AssignmentType": "S",
            "ProcessInstanceState": 1,
            "QueueType": "I",
            "QueueId": 15,
            "ProcessedBy": "badmin",
            "ProcessedByPersonalName": "New User",
            "ActivityType": 1,
            "WorkItemStateId": 2,
            "VAR_REC_1": "81",
            "VAR_REC_2": "F",
            "QueueData": [
                {
                    "Name": "Q_Age",
                    "Value": 23
                },
                {
                    "Name": "Q_DateOfJoining",
                    "Value": "2021-09-12 00:00:00.0"
                },
                {
                    "Name": "Q_Id",
                    "Value": 44
                },
                {
                    "Name": "Q_Name",
                    "Value": "Justin"
                },
                {
                    "Name": "Q_Salary",
                    "Value": 80000.00
                },
                {
                    "Name": "Ext_Date_Of_Birth",
                    "Value": "1989-06-01 00:00:00.0"
                },
                {
                    "Name": "Ext_Type",
                    "Value": "Adult"
                }
            ]
        }
    ]
}
```

Searching workitems using filters and criteria management

This section contains the request and response body to search workitems using the defined filters and utilising the criteria management feature.

Request:


```
{
  "EnableMultilingual": "Y",
  "OpenMode": "WD",
  "IsCriteriaReportCase": "Y",
  "GetWICount": "N",
  "CriteriaId": 2,
  "FilterId": 2,
  "DataFlag": "Y",
  "BatchInfo": {
    "OrderBy": 2,
    "SortOrder": "D",
    "LastValue": null,
    "ProcessInstanceId": null,
    "WorkItemId": null,
    "NoOfRecordsToFetch": 20
  },
  "UserDefVarFlag": "Y",
  "ShowAllWorkItemsFlag": "Y",
  "IsCountEnabled": "Y"
}
```


Response:


```
{
  "WorkItemList": [
    {
      "RegistrationNo": "SearchWIList-00000000000000000000000042",
      "QueueName": "SearchWIList_Start",
      "ProcessName": "SearchWIList",
      "ProcessVersionNo": 1,
      "ActivityName": "Start",
      "WorkItemState": "RUNNING",
      "CheckListStatus": "N",
      "PendingSince": "2023-01-05 11:54:39",
      "WorkItemId": 1,
      "PriorityLevel": 1,
      "ParentWorkItemId": 0,
```

```
      "ProcessDefinitionId": 16,
      "ActivityId": 1,
      "InstrumentStatus": "N",
      "LockStatus": "N",
      "CreatedByUserName": "Jason",
      "CreationDateTime": "2023-01-05 11:54:39",
      "AssignmentType": "S",
      "ProcessInstanceState": 1,
      "QueueType": "I",
      "QueueId": 65,
      "ProcessedBy": "Jason",
      "ProcessedByPersonalName": "Jason",
      "ProcessedByFamilyName": "Thomas",
      "ActivityType": 1,
      "WorkItemStateId": 2,
      "VAR_REC_1": "2416",
      "VAR_REC_2": "F",
      "QueueData": [
        {
          "Name": "VAR_REC_2",
          "Value": "F"
        },
        {
          "Name": "Age",
          "Value": "16"
        },
        {
          "Name": "Salary",
          "Value": "20000"
        },
        {
          "Name": "Category",
          "Value": "Minor"
        },
        {
          "Name": "DateOfJoining",
          "Value": "2022-09-13 00:00:00"
        },
        {
          "Name": "DueAmount",
          "Value": "15000"
        }
      ]
    },
    {
      "RegistrationNo": "SearchWIList-00000000000000000000000041",
      "QueueName": "SearchWIList_Start",
      "ProcessName": "SearchWIList",
      "ProcessVersionNo": 1,
      "ActivityName": "Start",
      "WorkItemState": "NOTSTARTED",
```

```
      "CheckListStatus": "N",
      "PendingSince": "2023-01-05 11:54:38",
      "WorkItemId": 1,
      "PriorityLevel": 1,
      "ParentWorkItemId": 0,
      "ProcessDefinitionId": 16,
      "ActivityId": 1,
      "InstrumentStatus": "N",
      "LockStatus": "N",
      "CreatedByUserName": "Peter",
      "CreationDateTime": "2023-01-05 11:54:38",
      "AssignmentType": "S",
      "ProcessInstanceState": 1,
      "QueueType": "I",
      "QueueId": 65,
      "ProcessedBy": "Peter",
      "ProcessedByPersonalName": "Peter",
      "ProcessedByFamilyName": "Jordan",
      "ActivityType": 1,
      "WorkItemStateId": 1,
      "VAR_REC_1": "2415",
      "VAR_REC_2": "F",
      "QueueData": [
        {
          "Name": "VAR_REC_2",
          "Value": "F"
        },
        {
          "Name": "Age",
          "Value": "16"
        },
        {
          "Name": "Salary",
          "Value": "20000"
        },
        {
          "Name": "Category",
          "Value": "Minor"
        },
        {
          "Name": "DateOfJoining",
          "Value": "2022-09-13 00:00:00"
        },
        {
          "Name": "DueAmount",
          "Value": "15000"
        }
      ]
    }
  ],
  "TotalCount": 2,
```

```
  "RetrievedCount": 2,
  "WorkItemCount": 2,
  "LastModifiedOn": "2022-12-30 12:16:22"
}
```


Retrieving workitem count after applying defined filters

This section contains the request and response body to retrieve workitem count after applying defined filters under a single criteria.

Request:


```
{
  "EnableMultilingual": "Y",
  "OpenMode": "WD",
  "IsCriteriaReportCase": "Y",
  "GetWICount": "Y",
  "CriteriaId": 2,
  "DataFlag": "Y",
  "BatchInfo": {
    "OrderBy": 2,
    "SortOrder": "D",
    "LastValue": null,
    "ProcessInstanceId": null,
    "WorkItemId": null,
    "NoOfRecordsToFetch": 20
  },
  "UserDefVarFlag": "Y",
  "ShowAllWorkItemsFlag": "Y",
  "IsCountEnabled": "N"
}
```


Response:


```
{
  "FilterInfo": [
    {
      "FilterId": 2,
      "FilterAlias": "First_Filter",
      "WICount": 12,
      "EntityName": null
    },
    {
      "FilterId": 3,
```

```
      "FilterAlias": "Second_Filter",
      "WICount": 2,
      "EntityName": null
    },
    {
      "FilterId": 4,
      "FilterAlias": "Third_Filter",
      "WICount": 2,
      "EntityName": null
    }
  ],
  "LastModifiedOn": "2022-12-30 12:16:22"
}
```


WFFetchWorkList

This *WFFetchWorklist* API allows you to fetch the list of workitems present in a specific queue based on queue, user, and group filters or alias applied to the queue. Through this API, you can also perform search operations on specified queue, and fetch the workitems in MyQueue.

- Service URL — *http://<GatewayIp:GatewayPort>/workitem-service-context/workitem/fetchworklist*
  Where, GatewayIP is the IP of the machine, and GatewayPort is the port of the machine where the workitem service is deployed.
  Example, http: //192.168.148.91:9090/workitem-service context/workitem/fetchworklist
- Request Media Type — application/json
- Response Media Type — application/json
- Method Type — POST

Header parameters description

The following table describes the header parameters of the *WFFetchWorklist* service API, if OAuth is disabled:

Name	Parameter Scope	Type	Description	Mandatory
Accept-Language	HeaderParam	Locale	Indicates the natural language and locale that the client prefers. Example: en-US	No
Tenant-Id	HeaderParam	String	Name of the cabinet.	Yes
Authorization	HeaderParam	String	It is the signed-in user's session id.	Yes

If OAuth is enabled:

Name	Parameter scope	Type	Description	Mandatory
AppId	HeaderParam	String	Registered AppId of the product	No
Access Token	HeaderParam	String	AccessToken for the signed-in user	No
SessionValid	HeaderParam	String	SessionValid for the signed-in user	No

Request body parameters description

The following table describes the request body parameters of the *WFFetchWorkList* service API:

Name	Parent Tag	Type	Description	Default Value	Valid Value
DataFlag	Root	String	Indicates whether the workitem data is returned or not.	N	• Y– Yes and it returns the workitem data as well • N – No and it does not return the Workitem Data
FetchLocked Flag	Root	String	Indicate whether the locked workitems can be fetched from the queue.	N	• Y- Yes • N-No
ReturnCount Flag	Root	String	Indicates whether only the count is to be returned or both count and workitem list is to be returned.	N	• Y-Return count and List of workItems present in the requested queue. • N-Return only count of workItems present in the queue
MyQueue Flag	Root	String	Indicates whether the workitems are fetched from the My queue of the user.	N	• Y- Yes • N-No

Name	Parent Tag	Type	Description	Default Value	Valid Value
ProcessAlias	Root	String	Indicate whether Process specific alias with values is returned while fetching workitems present in MyQueue. If there are workitems of different processes in MyQueue then process specific alias is not returned.	N	• Y- Yes • N-No
ProcessDefId	Root	Integer	Process definition ID is the Process ID for the fetched workitems.	0	Zero or positive Integer values
ClientOrder Flag	Root	String	Indicates whether the client has clicked on some field or Alias for sorting.	N	• Y-Yes • N-No
Starting RecordNo	Root	Integer	Indicates the starting index from of the fetched workitems.	0	Zero or positive Integer values
Reminder Flag	Root	String	Indicates when ReminderFlag is to be returned in response.	N	• Y-Yes, append the reminderFlag in response • N-No
PagingFlag	Root	String	A flag is Y if paging is enabled through webdesktop.ini.	N	• Y-Yes • N-No

Name	Parent Tag	Type	Description	Default Value	Valid Value
QueueType	Root	String	Queue Type for the ID specified in the QueueId Tag.	NA	• I– Introduction • F – FIFO • D–(WIP)Dynamic • N – (WIP) No Assignment • S–(WIP) Permanent • U – User • Queue • Q – Query • Queue
BatchInfo	Root	WFBatch Info	Contains the details of parameters used in batching a list of workitems.	NA	NA
NoOfRecords ToFetch	BatchInfo	Integer	Batch size information in terms of no. of workitems fetched per batch.	NA	Positive Integer Values

Name	Parent Tag	Type	Description	Default Value	Valid Value
OrderBy	BatchInfo	Integer	An integer value indicates the order by field to sort the return workitem list. The returned list of workitems can be sorted into fields like Priority, Activity Name, Locked By, and so on. The list can also be sorted on the Mapped Process Variables or Aliases, for which the values are specified starting from 101.	2	• 1 - PriorityLevel • 2- ProcessInstanceId • 3 - ActivityName • 4 - LockedByName • 5 - IntroducedBy • 6 - Instrument Status • 7 - CheckList CompleteFlag • 8 - LockStatus • 9 - WorkitemState • 10 - EntryDateTime • 11 - ValidTill • 12 - LockedTime • 13 - Introduction DateTime • 17 - Status • 18 - CreatedDate Time • Above 100 – Alias Basically, continue adding 100 to the order in which the Aliases are returned for the Queue. For example, if the 1st Alias returned is LoanAmount, 2nd is LoanType, then to sort on Loan Amount, specify 101. Similarly, to sort on LoanType specify 102, and more.

Name	Parent Tag	Type	Description	Default Value	Valid Value
SortOrder	BatchInfo	String	Workitems sort order.	A	• A-Ascending • D-Descending
LastValue	BatchInfo	String	Value of the Sort Order By field for the last workitem fetched in the previous batch. This tag requires no specification while fetching the 1st batch.	NA	NA
LastProcess Instance	BatchInfo	String	Process Instance ID of the last fetched workitem in the previous batch. This tag requires no specification while fetching the 1st batch.	NA	NA
Last WorkItem	BatchInfo	Integer	Workitem ID of the last fetched workitem in the previous batch. This tag requires no specification while fetching the 1st batch.	NA	Positive Integer Values
Filter	Filter	WFFilter	Contains the parameters used for preparing filter to sort the list of workitem	NA	NA

Name	Parent Tag	Type	Description	Default Value	Valid Value
Attribute Name	Root	String	Signifies the Type is specified as a value between 0-255. This is the name of the Queue variable for which the workitem list value gets filtered.	NA	NA
Dynamic Queue Flag	Filter	String	Indicates that the dynamically assigned workitems are to be fetched from the dynamic queue.	NA	N-No Y-Yes
QueueId	Filter	Integer	Queue ID corresponding to which workitems are to be fetched.	0	Zero or Positive Integer Values
Type	Filter	Integer	Indicates whether the filter String is prepared or if you need to prepare the filter using FilterJSON.	0	• 256 - FilterStringis prepared SQL query, Any other • Integer - make Filter using FilterJSON field
Comparison	Filter	Integer	Indicates whether the Integer equivalent of comparison operators is used in filters to sort the fetched workitems.	0	• 0-Equal To • 1 - less than • 2 - Less than • equal to • 3 - Equal to • 4 - Not equal to • 5 - Greater than, • 6 - Greater than equal to • 7 - Like • 8 - Not like • 9 - Null, Not null

Name	Parent Tag	Type	Description	Default Value	Valid Value
FilterString	Filter	String	If the Type is specified as other than 256, then this contains the Attribute Value specified in the AttributeName tag for which you can apply a filter on the workitem list. If the Type is specified as 256, then this contains the actual SQL where Clause string, which appends to the SQL string to fetch the workitems meeting the criteria.	NA	NA
FilterJSON	Filter	String	Contains the values of attributes to apply a filter on the fetched workitem list.	NA	NA
StateDate Range	FilterJSON	StateDate Range	Date Range corresponding to the state specified in the State tag.	NA	NA
StateDate RangeStart Date	StartDate Range	Date	Start date of the date range corresponding to the state specified in the State tag.	NA	NA
StateDate RangeEnd Date	StartDate Range	Date	End date of the date range corresponding to the state specified in the State tag.	NA	NA

Name	Parent Tag	Type	Description	Default Value	Valid Value
Process DefinitionID	FilterJSON	Integer	Process Definition ID Value is used in the filter to fetch the workitems of the process definition ID.	NA	NA
Process Name	FilterJSON	String	Process name filter to be applied on workitems in worklist.	NA	NA
ActivityId	FilterJSON	Integer	Activity ID filter to be applied on workitems in worklist.	NA	NA
ActivityName	FilterJSON	String	Activity name filter to be applied on workitems in worklist.	NA	NA
Process Instance Name	FilterJSON	String	Process Instance ID of the workitem on which filter is applied.	NA	NA
Priority	FilterJSON	String	Priority level Value to filter the workitem list.	NA	NA
Exception Status	FilterJSON	String	Value of Instrument Status to apply a filter on workitem list.	NA	NA
LockStatus	FilterJSON	String	To filter the response based on the current lock status of workitems.	NA	• Y- To fetch the locked as well as unlocked workitems • N- To fetch only the unlocked workitems in Response

Name	Parent Tag	Type	Description	Default Value	Valid Value
Introduced ByUser	FilterJSON	String	Introduced By User Value is used in the filter to fetch workitems introduced by a specific user.	NA	NA
LockedBy User	FilterJSON	String	Locked By User Value is used in the filter to fetch workitems locked by a specific user.	NA	NA
AssignedTo User	FilterJSON	String	Assigned To User Value is used in the filter to fetch workitems locked by a specific user.	NA	NA
ReferBy Name	FilterJSON	String	Referred by name is used to make a filter and to fetch the workitems list.	NA	NA

Name	Parent Tag	Type	Description	Default Value	Valid Value
Condition	FilterJSON	List <Condition>	To fetch the workitems based on certain predefined filters. The filters are same that are used in "Set filter option in webdesktop on UI". Various conditions on different parameters of queue can be applied. For example, PriorityLevel=Low, LockStatus=N, InstrumentStatus=N. This list contains the definition of such conditions.	NA	NA
Operator	Condition	Integer	Integer equivalent of comparison character.	NA	• 1 - Less than • 2 - Less than • equal to • 3 - Equal to • 4 - Not equal to • 5 - Greater than • 6 - Greater than equal to • 7-Like • 8-Not like • 9-Null • 10-Not null
JoinCondition	Condition	String	Join conditions between multiple conditions applied on a queue.	NA	And, Or, Not

Name	Parent Tag	Type	Description	Default Value	Valid Value
Alias	Condition	Alias	Contains the name, DataType, and value of the parameter to apply a filter on the queue to fetch the list of workitems.	NA	NA
Name	Alias	String	Name of the parameter based on which you can filter the workitems list present in the queue. For example - LockStatus, InstrumentStatus, and so on.	NA	NA
Type	Alias	Integer	Integer equivalent of the DataType of the parameter used to filter workitems present on the queue.	NA	• 8 - Date • 15 - short Date • 3 - Integer • 4 - Long • 10 - String
Value	Alias	String	Value of the parameter to apply a filter on the list of workitems present on the queue.	NA	NA

Response body parameters description

The following table describes the response body parameters of the *WFFetchWorkList* service API:

Name	Parent Tag	Parameter Type	Description
UserName	Root	String	User Name for the called API.

Name	Parent Tag	Parameter Type	Description
QueueId	Root	Integer	Queue ID of requested workitems.
RetrievedCount	Root	Integer	No. of workitems retrieved in the current Call. This is dependent on the batch size indicated in the request.
Count	Root	Integer	The total No. of workitems that meet the specified criteria.
WorkitemCount	Root	Integer	Count of the Workitem.
AliasProcessDefId	Root	Integer	If ProcessAlias=Y in webdesktop.ini then AliasProcessDefId is returned with the process definition ID of the process in Myqueue with Alias defined. If MyQueue has more than 1 process workitem then AliasProcessDefid is not returned.
Instruments	Root	WFInstrument	List of Workitems.
ProcessInstanceId	Instruments	String	The process instance ID of the workitem is returned.
WorkItemName	Instruments	String	Name of the workitem.
RouteId	Instruments	Integer	Process Definition ID of the fetched Workitem.
RouteName	Instruments	String	The process name of the fetched workitem.
WorkStageId	Instruments	Integer	Activity ID of the fetched workitem.

Name	Parent Tag	Parameter Type	Description
ActivityName	Instruments	String	Activity Name of the fetched workitem.
PriorityLevel	Instruments	Integer	The Priority Level of the fetched workitem. 1 – Low 2 – Medium 3 – High 4 – Very High
InstrumentStatus	Instruments	String	Exception Status of the fetched workitem. E – In Exception N – No Exception raised
LockStatus	Instruments	String	Lock status of the fetched workitem. Yes – Locked No – Unlocked
LockedByUser Name	Instruments	String	User Name for the locked workitem.
LockedByPersonal Name	Instruments	String	Personal Name of the User for the locked workitem.
LockedByFamily Name	Instruments	String	Family Name of the User for the locked workitem.
ExpiryDateTime	Instruments	Date	Date and time validity of the fetched workitem for current activity.
CreatedByUser Name	Instruments	string	Name of the user for the fetched workitem.
CreatedByPersonal Name	Instruments	String	Personal Name of the user for the fetched workitem.
CreatedByFamily Name	Instruments	String	Family Name of the user for the fetched workitem.

Name	Parent Tag	Parameter Type	Description
CreationDateTime	Instruments	Date	Date-Time when the Document is added to the workitem.
WorkitemState	Instruments	String	State of the fetched workitem. 1 – Not Started 2 – Running 3 – Suspended 4 – Terminated 5 – Aborted 6 – Completed
CheckListComplete Flag	Instruments	String	Indicates all the Defined Process ToDos of the completed workitems. Y – Complete N – Incomplete
EntryDateTime	Instruments	Date	The date and Time for the workitem entered the current activity.
LockedTime	Instruments	Date	Date and Time of the workitem locked by the user.
IntroductionDate Time	Instruments	Date	Date and Time for the workitem introduced or started in the process.
IntroducedBy	Instruments	String	Name of the user for the fetched workitem introduced.
IntroducedByPersonal Name	Instruments	String	Personal Name of the user for the fetched workitem introduced.
IntroducedByFamily Name	Instruments	String	Family Name of the user for the fetched workitem introduced.
AssignedTo	Instruments	String	User name for the currently assigned workitem.

Name	Parent Tag	Parameter Type	Description
AssignedToPersonal Name	Instruments	String	Personal Name of the user for the currently assigned workitem.
AssignedToFamily Name	Instruments	String	Family Name of the user for the currently assigned workitem.
WorkItemId	Instruments	Integer	Workitem ID of the returned workitem.
QueueName	Instruments	String	Queue Name of the workitem.
AssignmentType	Instruments	String	Current Assignment Type for the workitem. S – Shared F – Fixed
ProcessInstanceState	Instruments	Integer	The state of the Process Instance corresponding to the workitem. 1 – Not Started 2 – Running 3 – Suspended 4 – Terminated 5 – Aborted 6 – Completed
QueueType	Instruments	String	The Type of Queue for the current workitem. I – Introduction F – FIFO D – (WIP) Dynamic N – (WIP) No Assignment S – (WIP) Permanent
Status	Instruments	String	Status of the Exception. R – Raised C – Cleared
QueueId	Instruments	Integer	Queue ID of the workitem.

Name	Parent Tag	Parameter Type	Description
Turnaroundtime	Instruments	Date	The expected Turn-Around Time of the workitem in the workstep.
Referredby	Instruments	String	Name of the user for referred workitem.
Referredto	Instruments	Integer	User ID to whom workitem is referred.
Q_DivertedByUserId	Instruments	Integer	User ID for referred workitem.
ActivityType	Instruments	Integer	Type of Activity is as follows: 1 - Work Introduction 2 - Work Exit 3 - Discard 4 - Hold 5 - Distribute 6 - Collect 7 - Decision 18 - SubProcess 10 - Custom 19 – Integration 11 - Query
ProcessedBy	Instruments	String	Name of the User for the processed workitem.
ProcessedByPersonal Name	Instruments	String	Personal Name of the User for the processed workitem.
ProcessedByFamily Name	Instruments	String	Family Name of the User for the processed workitem.
URN	Instruments	String	URN generated corresponding to the workitem.

Name	Parent Tag	Parameter Type	Description
SecondaryDBFlag	Instruments	String	Indicates when the process variable data is present in the secondary cabinet for the fetched workitem. Y - Yes N - No
CalendarName	Instruments	String	Name of the calendar attached with the fetched workitem.
TATRemaining	Instruments	Long	Total remaining working time to complete the workitem as per the turnaround time defined on the current activity.
TATConsumed	Instruments	Long	Total working time consumed after the entry of the workitem on current activity.

Error codes and description

The following table describes the error codes that can occur while executing the *WFFetchWorkList* service API:

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
N/A	401	Unauthorized	Invalid Session	Regenerate the session and pass the valid session ID in the authorization header.

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_18	404	Not Found	No More Records	There are no more workitems present in the requested queue.
NGONE_WFS_400	404	Not Found	Operation Failed	Recheck the request body.
NGONE_WFS_852	404	Not Found	Alias Not Found.	The requested alias is incorrect.
NGONE_WFS_15	404	Not Found	Invalid Filter	The filter applied to fetch the list of workitems is incorrect resulting in zero fetched workitems.
NGONE_WFS_802	404	Not Found	Invalid Parameters	Invalid parameters passed for filter preparation.
NGONE_WFS_801	500	Internal Server Error	SQL exception occurred	An SQL exception stopped the operation. Refer to the error logs for further analysis.
NGONE_WFS_3001	400	Bad Request	Invalid input parameters	Refer to the detailed error message to correct input parameters that have invalid values. Example – (FieldName: ErrorDescription) SystemAssigned: SystemAssignedWI must have the value Y or N. QueueType: QueueType value can be D, F, I, M, N, S, and U.

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_986	404	Not Found	Invalid Queue	Pass the valid QueueID in the request body.
NGONE_WFS_503	503	Temporary	Service is either down or slow	Service is either unavailable or slow.

Sample request and response

Below is the sample request and response body for *WFetchWorkList* API as per the following scenarios.

Fetching worklist using queue

This section contains the request and response body to fetch worklist on clicking queue.

Request:

```
{
  "DataFlag": "Y",
  "FetchLockedFlag": "N",
  "Filter": {
    "DynamicQueueFlag": "Y",
    "QueueId": 2033,
    "Type": 256,
    "Comparison": 0
  },
  "BatchInfo": {
    "NoOfRecordsToFetch": 20,
    "OrderBy": 2,
    "SortOrder": "A",
    "LastValue": "",
    "LastProcessInstance": "",
    "LastWorkItem": ""
  },
  "QueueType": "I",
  "ClientOrderFlag": "N",
  "UseStdParser": "Y"
}
```

Response:


```
{
  "Instruments": [
    {
      "ProcessInstanceId": "practice1-0000001031-process",
      "WorkItemName": "practice1-0000001031-process",
      "RouteId": 1028,
      "RouteName": "todo_Documents1",
      "WorkStageId": 1,
      "ActivityName": "Start Event_1",
      "PriorityLevel": 1,
      "InstrumentStatus": "N",
      "LockStatus": "N",
      "CreatedByUserName": "Adam",
      "CreatedByPersonalName": "Adam",
      "CreatedByFamilyName": "Parker",
      "CreationDateTime": "2022-06-06 14:50:45.893",
      "WorkitemState": "RUNNING",
      "CheckListCompleteFlag": "N",
      "EntryDateTime": "2022-06-06 14:50:45.893",
      "WorkItemId": 1,
      "QueueName": "todo_Documents1_Start Event_1",
      "AssignmentType": "S",
      "ProcessInstanceState": 1,
      "QueueType": "I",
      "QueueId": 2033,
      "ActivityType": 1,
      "ProcessedBy": "Adam",
      "ProcessedByPersonalName": "Adam",
      "ProcessedByFamilyName": "Parker",
      "URN": "ToDo_Document1-1031",
      "SecondaryDBFlag": "N"
    },
    {
      "ProcessInstanceId": "practice1-0000001032-process",
      "WorkItemName": "practice1-0000001032-process",
      "RouteId": 1028,
      "RouteName": "todo_Documents1",
      "WorkStageId": 1,
      "ActivityName": "Start Event_1",
      "PriorityLevel": 1,
      "InstrumentStatus": "N",
      "LockStatus": "N",
      "CreatedByUserName": "Adam",
      "CreatedByPersonalName": "Adam",
      "CreatedByFamilyName": "Parker",
      "CreationDateTime": "2022-06-06 15:01:56.417",
      "WorkitemState": "RUNNING",
      "CheckListCompleteFlag": "N",
      "EntryDateTime": "2022-06-06 15:01:56.417",
```

```
      "WorkItemId": 1,
      "QueueName": "todo_Documents1_Start Event_1",
      "AssignmentType": "S",
      "ProcessInstanceState": 1,
      "QueueType": "I",
      "QueueId": 2033,
      "ActivityType": 1,
      "ProcessedBy": "Adam",
      "ProcessedByPersonalName": "Adam",
      "ProcessedByFamilyName": "Parker",
      "URN": "ToDo_Document1-1032",
      "SecondaryDBFlag": "N"
    },
    {
      "ProcessInstanceId": "practice1-0000001033-process",
      "WorkItemName": "practice1-0000001033-process",
      "RouteId": 1028,
      "RouteName": "todo_Documents1",
      "WorkStageId": 1,
      "ActivityName": "Start Event_1",
      "PriorityLevel": 1,
      "InstrumentStatus": "N",
      "LockStatus": "N",
      "CreatedByUserName": "Adam",
      "CreatedByPersonalName": "Adam",
      "CreatedByFamilyName": "Parker",
      "CreationDateTime": "2022-06-06 15:03:16.51",
      "WorkitemState": "RUNNING",
      "CheckListCompleteFlag": "N",
      "EntryDateTime": "2022-06-06 15:03:16.51",
      "WorkItemId": 1,
      "QueueName": "todo_Documents1_Start Event_1",
      "AssignmentType": "S",
      "ProcessInstanceState": 1,
      "QueueType": "I",
      "QueueId": 2033,
      "ActivityType": 1,
      "ProcessedBy": "Adam",
      "ProcessedByPersonalName": "Adam",
      "ProcessedByFamilyName": "Parker",
      "URN": "ToDo_Document1-1033",
      "SecondaryDBFlag": "N"
    },
    {
      "ProcessInstanceId": "practice1-0000001034-process",
      "WorkItemName": "practice1-0000001034-process",
      "RouteId": 1028,
      "RouteName": "todo_Documents1",
      "WorkStageId": 1,
      "ActivityName": "Start Event_1",
      "PriorityLevel": 1,
```

```
      "InstrumentStatus": "N",
      "LockStatus": "N",
      "CreatedByUserName": "Adam",
      "CreatedByPersonalName": "Adam",
      "CreatedByFamilyName": "Parker",
      "CreationDateTime": "2022-06-21 13:18:05.57",
      "WorkitemState": "RUNNING",
      "CheckListCompleteFlag": "N",
      "EntryDateTime": "2022-06-21 13:18:05.57",
      "WorkItemId": 1,
      "QueueName": "todo_Documents1_Start Event_1",
      "AssignmentType": "S",
      "ProcessInstanceState": 1,
      "QueueType": "I",
      "QueueId": 2033,
      "ActivityType": 1,
      "ProcessedBy": "Adam",
      "ProcessedByPersonalName": "Adam",
      "ProcessedByFamilyName": "Parker",
      "URN": "ToDo_Document1-1034",
      "SecondaryDBFlag": "N"
    },
    {
      "ProcessInstanceId": "practice1-0000002036-process",
      "WorkItemName": "practice1-0000002036-process",
      "RouteId": 1028,
      "RouteName": "todo_Documents1",
      "WorkStageId": 1,
      "ActivityName": "Start Event_1",
      "PriorityLevel": 1,
      "InstrumentStatus": "E",
      "LockStatus": "N",
      "CreatedByUserName": "Adam",
      "CreatedByPersonalName": "Adam",
      "CreatedByFamilyName": "Parker",
      "CreationDateTime": "2022-08-31 12:23:47.41",
      "WorkitemState": "Running",
      "CheckListCompleteFlag": "Y",
      "EntryDateTime": "2022-08-31 12:23:47.41",
      "WorkItemId": 1,
      "QueueName": "todo_Documents1_Start Event_1",
      "AssignmentType": "S",
      "ProcessInstanceState": 1,
      "QueueType": "I",
      "QueueId": 2033,
      "ActivityType": 1,
      "ProcessedBy": "Adam",
      "ProcessedByPersonalName": "Adam",
      "ProcessedByFamilyName": "Parker",
      "URN": "ToDo_Document1-2036",
      "SecondaryDBFlag": "N"
```

```
    },
    {
      "ProcessInstanceId": "practice1-0000002041-process",
      "WorkItemName": "practice1-0000002041-process",
      "RouteId": 1028,
      "RouteName": "todo_Documents1",
      "WorkStageId": 1,
      "ActivityName": "Start Event_1",
      "PriorityLevel": 1,
      "InstrumentStatus": "N",
      "LockStatus": "N",
      "CreatedByUserName": "Adam",
      "CreatedByPersonalName": "Adam",
      "CreatedByFamilyName": "Parker",
      "CreationDateTime": "2022-08-31 12:54:07.787",
      "WorkitemState": "RUNNING",
      "CheckListCompleteFlag": "N",
      "EntryDateTime": "2022-08-31 12:54:07.787",
      "WorkItemId": 1,
      "QueueName": "todo_Documents1_Start Event_1",
      "AssignmentType": "S",
      "ProcessInstanceState": 1,
      "QueueType": "I",
      "QueueId": 2033,
      "ActivityType": 1,
      "ProcessedBy": "Adam",
      "ProcessedByPersonalName": "Adam",
      "ProcessedByFamilyName": "Parker",
      "URN": "ToDo_Document1-2041",
      "SecondaryDBFlag": "N"
    }
  ],
  "RetrievedCount": 6,
  "Count": 6,
  "WorkitemCount": 0,
  "ParkerasProcessDefId": 0,
  "LastModifiedOn": "2022-03-21 17:08:11.047",
  "UserName": "Adam",
  "QueueId": 2033,
“MainCode”:0
}
```


Fetching worklist using set filters

This section contains the request and response body to fetch worklist by setting filter on workitem parameters.

Request:


```
{
  "CountFlag": "Y",
  "DataFlag": "Y",
  "ZipBuffer": "N",
  "FetchLockedFlag": "N",
  "Filter": {
    "DynamicQueueFlag": "Y",
    "QueueId": "25",
    "Type": "256",
    "Comparison": "0",
    "FilterJSON": "<FilterXML><QueueID>25</QueueID><Type>256</
Type><ProcessName>microservicenewrj</ProcessName><ProcessDefId>4</
ProcessDefId><SearchAttributes></SearchAttributes></FilterXML>"
  },
  "BatchInfo": {
    "NoOfRecordsToFetch": "20",
    "OrderBy": "2",
    "SortOrder": "A",
    "LastValue": "",
    "LastProcessInstance": "",
    "LastWorkItem": ""
  },
  "UserIndex": "46",
  "QueueType": "N",
  "UseStdParser": "Y"
}
```


Response:


```
{
  "MainCode": 0,
  "UserName": "maaz",
  "QueueId": 25,
  "RetrievedCount": 6,
  "Count": 6,
  "LastModifiedOn": "2023-01-09 11:43:38.000",
  "Instruments": [
    {
      "instrumentStatus": "N",
      "ProcessInstanceId": "RJ-0000000001-process1",
      "WorkItemName": "RJ-0000000001-process1",
      "RouteId": 4,
      "RouteName": "microservicenewrj",
      "WorkStageId": 2,
      "ActivityName": "Workdesk_2",
      "PriorityLevel": 1,
      "InstrumentStatus": "N",
      "LockStatus": "N",
```

```
    "CreatedByUserName": "server",
    "CreatedByPersonalName": "New User",
    "CreatedByFamilyName": "",
    "CreationDateTime": "2023-01-03 08:30:58.000",
    "WorkitemState": "RUNNING",
    "CheckListCompleteFlag": "N",
    "EntryDateTime": "2023-01-09 11:41:39.000",
    "IntroductionDateTime": "2023-01-09 11:41:38.000",
    "IntroducedBy": "maaz",
    "IntroducedByPersonalName": "Maaz",
    "IntroducedByFamilyName": "Ali",
    "WorkItemId": 1,
    "QueueName": "microservicenewrj_SwimLane_1",
    "AssignmentType": "S",
    "ProcessInstanceState": 2,
    "QueueType": "N",
    "QueueId": 25,
    "ReferredTo": 0,
    "TurnAroundDateTime": "2023-01-10 11:41:39.000",
    "Q_DivertedByUserId": 0,
    "ActivityType": 10,
    "ProcessedBy": "maaz",
    "ProcessedByPersonalName": "Maaz",
    "ProcessedByFamilyName": "Ali",
    "URN": "Microserviceprocess-1",
    "SecondaryDBFlag": "N",
    "TATRemaining": -8636,
    "TATConsumed": 10076,
    "Data": {
      "QueueData": [
        {
          "Name": "DATEALIAS",
          "Value": "1999-01-08 18:30:00"
        },
        {
          "Name": "FLOATALIAS",
          "Value": "6"
        },
        {
          "Name": "INTALIAS",
          "Value": "34"
        },
        {
          "Name": "LONGALIAS"
        },
        {
          "Name": "STRALIAS",
          "Value": "57"
        }
      ]
    }
```

```
    },
    {
      "instrumentStatus": "N",
      "ProcessInstanceId": "RJ-0000000003-process1",
      "WorkItemName": "RJ-0000000003-process1",
      "RouteId": 4,
      "RouteName": "microservicenewrj",
      "WorkStageId": 2,
      "ActivityName": "Workdesk_2",
      "PriorityLevel": 1,
      "InstrumentStatus": "N",
      "LockStatus": "N",
      "CreatedByUserName": "varun",
      "CreatedByPersonalName": "New User",
      "CreatedByFamilyName": "",
      "CreationDateTime": "2023-01-04 08:34:43.000",
      "WorkitemState": "NOTSTARTED",
      "CheckListCompleteFlag": "N",
      "EntryDateTime": "2023-01-06 11:52:46.000",
      "IntroductionDateTime": "2023-01-04 08:38:21.000",
      "IntroducedBy": "varun",
      "IntroducedByPersonalName": "New User",
      "IntroducedByFamilyName": "",
      "WorkItemId": 1,
      "QueueName": "microservicenewrj_SwimLane_1",
      "AssignmentType": "S",
      "ProcessInstanceState": 2,
      "QueueType": "N",
      "QueueId": 25,
      "ReferredTo": 0,
      "TurnAroundDateTime": "2023-01-07 11:52:46.000",
      "Q_DivertedByUserId": 0,
      "ActivityType": 10,
      "ProcessedBy": "varun",
      "ProcessedByPersonalName": "New User",
      "ProcessedByFamilyName": "",
      "URN": "Microserviceprocess-3",
      "SecondaryDBFlag": "N",
      "TATRemaining": -12945,
      "TATConsumed": 14385,
      "Data": {
        "QueueData": [
          {
            "Name": "DATEALIAS",
            "Value": "1998-08-28 18:30:00"
          },
          {
            "Name": "FLOATALIAS",
            "Value": "4555"
          },
          {
```

```
            "Name": "INTALIAS",
            "Value": "45"
          },
          {
            "Name": "LONGALIAS"
          },
          {
            "Name": "STRALIAS",
            "Value": "10655"
          }
        ]
      }
    },
    {
      "instrumentStatus": "N",
      "ProcessInstanceId": "RJ-0000000005-process1",
      "WorkItemName": "RJ-0000000005-process1",
      "RouteId": 4,
      "RouteName": "microservicenewrj",
      "WorkStageId": 2,
      "ActivityName": "Workdesk_2",
      "PriorityLevel": 1,
      "InstrumentStatus": "N",
      "LockStatus": "N",
      "CreatedByUserName": "maaz",
      "CreatedByPersonalName": "Maaz",
      "CreatedByFamilyName": "Ali",
      "CreationDateTime": "2023-01-06 11:49:27.000",
      "WorkitemState": "NOTSTARTED",
      "CheckListCompleteFlag": "N",
      "EntryDateTime": "2023-01-06 11:52:47.000",
      "IntroductionDateTime": "2023-01-06 11:50:57.000",
      "IntroducedBy": "maaz",
      "IntroducedByPersonalName": "Maaz",
      "IntroducedByFamilyName": "Ali",
      "WorkItemId": 1,
      "QueueName": "microservicenewrj_SwimLane_1",
      "AssignmentType": "S",
      "ProcessInstanceState": 2,
      "QueueType": "N",
      "QueueId": 25,
      "ReferredTo": 0,
      "TurnAroundDateTime": "2023-01-07 11:52:47.000",
      "Q_DivertedByUserId": 0,
      "ActivityType": 10,
      "ProcessedBy": "maaz",
      "ProcessedByPersonalName": "Maaz",
      "ProcessedByFamilyName": "Ali",
      "URN": "Microserviceprocess-5",
      "SecondaryDBFlag": "N",
      "TATRemaining": -12945,
```

```
    "TATConsumed": 14385,
    "Data": {
      "QueueData": [
        {
          "Name": "DATEALIAS",
          "Value": "1990-05-21 18:30:00"
        },
        {
          "Name": "FLOATALIAS",
          "Value": "51"
        },
        {
          "Name": "INTALIAS",
          "Value": "15"
        },
        {
          "Name": "LONGALIAS"
        },
        {
          "Name": "STRALIAS",
          "Value": "51"
        }
      ]
    }
  },
  {
    "instrumentStatus": "N",
    "ProcessInstanceId": "RJ-0000000006-process1",
    "WorkItemName": "RJ-0000000006-process1",
    "RouteId": 4,
    "RouteName": "microservicenewrj",
    "WorkStageId": 2,
    "ActivityName": "Workdesk_2",
    "PriorityLevel": 1,
    "InstrumentStatus": "N",
    "LockStatus": "N",
    "CreatedByUserName": "abhi",
    "CreatedByPersonalName": "New User",
    "CreatedByFamilyName": "",
    "CreationDateTime": "2023-01-09 06:57:44.000",
    "WorkitemState": "RUNNING",
    "CheckListCompleteFlag": "N",
    "EntryDateTime": "2023-01-09 11:41:40.000",
    "IntroductionDateTime": "2023-01-09 11:41:39.000",
    "IntroducedBy": "maaz",
    "IntroducedByPersonalName": "Maaz",
    "IntroducedByFamilyName": "Ali",
    "WorkItemId": 1,
    "QueueName": "microservicenewrj_SwimLane_1",
    "AssignmentType": "S",
    "ProcessInstanceState": 2,
```

```
      "QueueType": "N",
      "QueueId": 25,
      "ReferredTo": 0,
      "TurnAroundDateTime": "2023-01-10 11:41:40.000",
      "Q_DivertedByUserId": 0,
      "ActivityType": 10,
      "ProcessedBy": "maaz",
      "ProcessedByPersonalName": "Maaz",
      "ProcessedByFamilyName": "Ali",
      "URN": "Microserviceprocess-6",
      "SecondaryDBFlag": "N",
      "TATRemaining": -8636,
      "TATConsumed": 10076,
      "Data": {
        "QueueData": [
          {
            "Name": "DATEALIAS",
            "Value": "2007-05-11 18:30:00"
          },
          {
            "Name": "FLOATALIAS",
            "Value": "23000"
          },
          {
            "Name": "INTALIAS",
            "Value": "121"
          },
          {
            "Name": "LONGALIAS"
          },
          {
            "Name": "STRALIAS",
            "Value": "Abhi"
          }
        ]
      }
    },
    {
      "instrumentStatus": "N",
      "ProcessInstanceId": "RJ-0000000009-process1",
      "WorkItemName": "RJ-0000000009-process1",
      "RouteId": 4,
      "RouteName": "microservicenewrj",
      "WorkStageId": 2,
      "ActivityName": "Workdesk_2",
      "PriorityLevel": 1,
      "InstrumentStatus": "N",
      "LockStatus": "N",
      "CreatedByUserName": "maaz",
      "CreatedByPersonalName": "Maaz",
      "CreatedByFamilyName": "Ali",
```

```
      "CreationDateTime": "2023-01-09 11:45:15.000",
      "WorkitemState": "NOTSTARTED",
      "CheckListCompleteFlag": "N",
      "EntryDateTime": "2023-01-09 11:45:59.000",
      "IntroductionDateTime": "2023-01-09 11:45:55.000",
      "IntroducedBy": "maaz",
      "IntroducedByPersonalName": "Maaz",
      "IntroducedByFamilyName": "Ali",
      "WorkItemId": 1,
      "QueueName": "microservicenewrj_SwimLane_1",
      "AssignmentType": "S",
      "ProcessInstanceState": 2,
      "QueueType": "N",
      "QueueId": 25,
      "ReferredTo": 0,
      "TurnAroundDateTime": "2023-01-10 11:45:59.000",
      "Q_DivertedByUserId": 0,
      "ActivityType": 10,
      "ProcessedBy": "maaz",
      "ProcessedByPersonalName": "Maaz",
      "ProcessedByFamilyName": "Ali",
      "URN": "Microserviceprocess-9",
      "SecondaryDBFlag": "N",
      "TATRemaining": -8631,
      "TATConsumed": 10071,
      "Data": {
        "QueueData": [
          {
            "Name": "DATEALIAS",
            "Value": "2090-08-17 18:30:00"
          },
          {
            "Name": "FLOATALIAS",
            "Value": "90"
          },
          {
            "Name": "INTALIAS",
            "Value": "90"
          },
          {
            "Name": "LONGALIAS"
          },
          {
            "Name": "STRALIAS",
            "Value": "90"
          }
        ]
      }
    },
    {
      "instrumentStatus": "N",
```

```
    "ProcessInstanceId": "RJ-0000000010-process1",
    "WorkItemName": "RJ-0000000010-process1",
    "RouteId": 4,
    "RouteName": "microservicenewrj",
    "WorkStageId": 2,
    "ActivityName": "Workdesk_2",
    "PriorityLevel": 1,
    "InstrumentStatus": "N",
    "LockStatus": "N",
    "CreatedByUserName": "maaz",
    "CreatedByPersonalName": "Maaz",
    "CreatedByFamilyName": "Ali",
    "CreationDateTime": "2023-01-09 11:47:53.000",
    "WorkitemState": "NOTSTARTED",
    "CheckListCompleteFlag": "N",
    "EntryDateTime": "2023-01-09 11:48:43.000",
    "IntroductionDateTime": "2023-01-09 11:48:42.000",
    "IntroducedBy": "maaz",
    "IntroducedByPersonalName": "Maaz",
    "IntroducedByFamilyName": "Ali",
    "WorkItemId": 1,
    "QueueName": "microservicenewrj_SwimLane_1",
    "AssignmentType": "S",
    "ProcessInstanceState": 2,
    "QueueType": "N",
    "QueueId": 25,
    "ReferredTo": 0,
    "TurnAroundDateTime": "2023-01-10 11:48:43.000",
    "Q_DivertedByUserId": 0,
    "ActivityType": 10,
    "ProcessedBy": "maaz",
    "ProcessedByPersonalName": "Maaz",
    "ProcessedByFamilyName": "Ali",
    "URN": "Microserviceprocess-10",
    "SecondaryDBFlag": "N",
    "TATRemaining": -8629,
    "TATConsumed": 10069,
    "Data": {
      "QueueData": [
        {
          "Name": "DATEALIAS",
          "Value": "2090-12-18 18:30:00"
        },
        {
          "Name": "FLOATALIAS",
          "Value": "90"
        },
        {
          "Name": "INTALIAS",
          "Value": "90"
        },
```

```
          {
            "Name": "LONGALIAS"
          },
          {
            "Name": "STRALIAS",
            "Value": "90"
          }
        ]
      }
    }
  ]
}
```

Fetching worklist using advanced search

This section contains the request and response body to fetch worklist using advanced search on queue.

Request:

```
{
  "DataFlag": "Y",
  "ZipBuffer": "N",
  "FetchLockedFlag": "N",
  "ReminderFlag": "Y",
  "ReturnCountFlag": "Y",
  "Filter": {
    "QueueId": 25,
    "Type": 256,
    "FilterJSON": "<FilterXML><QueueName>microservicenewrj_SwimLane_1</
QueueName><LocalQueueName>microservicenewrj_SwimLane_1</
LocalQueueName><QueueID>25</QueueID><Type>256</
Type><ProcessName>microservicenewrj</
ProcessName><LocalProcessName>microservicenewrj</
LocalProcessName><ProcessDefinitionID>4</
ProcessDefinitionID><ActivityName>Workdesk_2</ActivityName><ActivityId>2</
ActivityId><SearchPreviousVersion>N</SearchPreviousVersion><SortOnVersion>N</
SortOnVersion><URNFlag>Y</URNFlag><URNName></URNName><ExcludeExitWorkitems>N</
ExcludeExitWorkitems><State>2</State><HoldType>1</HoldType><ArchiveSearch>N</
ArchiveSearch><LockedByUser></LockedByUser><LockedByUserId>0</
LockedByUserId><AssignedToUser></AssignedToUser><AssignedToUserId>0</
AssignedToUserId><IntroducedByUser>Adam</IntroducedByUser><IntroducedByUserId>46</
IntroducedByUserId><IntroducedByUserName>Adam</
IntroducedByUserName><SearchAttributes><Condition Operator=\"6\"
JoinCondition=\"AND\"><Alias Name=\"dateAlias\" Type=\"8\">2090-12-18 18:30:00</
Alias></Condition><Condition Operator=\"2\" JoinCondition=\"AND\"><Alias
Name=\"dateAlias\" Type=\"8\">2090-12-19 18:29:59</Alias></Condition><Condition
```

```
Operator=\"3\" JoinCondition=\"AND\"><Alias Name=\"floatAlias\" Type=\"6\">90</
Alias></Condition><Condition Operator=\"3\" JoinCondition=\"AND\"><Alias
Name=\"intALias\" Type=\"3\">90</Alias></Condition><Condition Operator=\"7\"
JoinCondition=\"AND\"><Alias Name=\"strALias\" Type=\"10\">90</Alias></
Condition><AdvanceSearch>N</AdvanceSearch><TimeRequired>Y</TimeRequired></
SearchAttributes><PinToSearch>false</PinToSearch><SearchFlag>2</SearchFlag></
FilterXML>"
  },
  "BatchInfo": {
    "NoOfRecordsToFetch": "20",
    "OrderBy": "2",
    "SortOrder": "D",
    "LastValue": "",
    "LastProcessInstance": "",
    "LastWorkItem": ""
  },
  "UserIndex": "46",
  "QueueType": "",
  "IsAdvanceSearchOnQueue": "Y",
  "UseStdParser": "Y"
}
```


Response:


```
{
  "MainCode": 0,
  "UserName": "Adam",
  "QueueId": 25,
  "RetrievedCount": 1,
  "Count": 1,
  "WorkitemCount": 1,
  "LastModifiedOn": "2023-01-09 11:43:38",
  "ReminderFlag": "N",
  "Instruments": [
    {
      "instrumentStatus": "N",
      "ProcessInstanceId": "RJ-0000000010-process1",
      "WorkItemName": "RJ-0000000010-process1",
      "RouteId": 4,
      "RouteName": "microservicenewrj",
      "WorkStageId": 2,
      "ActivityName": "Workdesk_2",
      "PriorityLevel": 1,
      "InstrumentStatus": "N",
      "LockStatus": "N",
      "CreatedByUserName": "Adam",
      "CreatedByPersonalName": "Adam",
      "CreatedByFamilyName": "Richards",
      "CreationDateTime": "2023-01-09 11:47:53",
      "WorkitemState": "NOTSTARTED",
      "CheckListCompleteFlag": "N",
```

```
      "EntryDateTime": "2023-01-09 11:48:43",
      "IntroductionDateTime": "2023-01-09 11:48:42",
      "IntroducedBy": "Adam",
      "IntroducedByPersonalName": "Adam",
      "IntroducedByFamilyName": "Richards",
      "WorkItemId": 1,
      "QueueName": "microservicenewrj_SwimLane_1",
      "AssignmentType": "S",
      "ProcessInstanceState": 2,
      "QueueType": "N",
      "QueueId": 25,
      "ReferredTo": 0,
      "TurnAroundDateTime": "2023-01-10 11:48:43",
      "Q_DivertedByUserId": 0,
      "ActivityType": 10,
      "ProcessedBy": "Adam",
      "ProcessedByPersonalName": "Adam",
      "ProcessedByFamilyName": "Richards",
      "URN": "Microserviceprocess-10",
      "SecondaryDBFlag": "N",
      "TATRemaining": -8462,
      "TATConsumed": 9902,
      "Data": {
        "QueueData": [
          {
            "Name": "DATEALIAS",
            "Value": "2090-12-18 18:30:00"
          },
          {
            "Name": "FLOATALIAS",
            "Value": "90"
          },
          {
            "Name": "INTALIAS",
            "Value": "90"
          },
          {
            "Name": "LONGALIAS"
          },
          {
            "Name": "STRALIAS",
            "Value": "90"
          }
        ]
      }
    }
  ]
}
```

WFSetOAuthCredentials

This API is used to set OAuth credentials in WFAppDetails

- URL: *http://<GatewayIp: GatewayPort >/user-service-context/users/set-oauth*.
- Request Media Type: application/json
- Response Media Type: application/json
- Method Operation: POST

Header parameters description

The following table describes the header parameters of the *WFSetOAuthCredentials* service API:

Name	Parameter Scope	Type	Description	Mandatory
Accept-Language	HeaderParam	Locale	Indicates the natural language and locale that the client prefers. Example: en-US	No
Tenant-Id	HeaderParam	String	Name of the cabinet.	Yes
Authorization	HeaderParam	String	It is the signed-in user's session id.	Yes

Request body parameters description

The following table describes the request body parameters of the *WFSetOAuthCredentials* service API:

Name	ParentTag	Type	Description	ValidValues	Default Values	Mandatory
AppId	Root	String	ID for the registered application.	N/A	N/A	Yes
AppName	Root	String	Name of the App for the registered application.	N/A	N/A	Yes
OAuthEnabled	Root	String	Signifies whether OAuth is enabled or not in the cabinet.	Y-In case OAuth is enabled	N	Yes
ServerSidePublicKey	Root	String	Public key for server operations, used in RSA algorithim.	N/A	N/A	Yes
ClientSidePrivateKey	Root	String	Encrypted key for client-side	N/A	N/A	Yes
ClientSidePublicKeyRoot	Root	String	Public key for client side operations	N/A	N/A	Yes

Response body parameters description

The following table describes the response body parameters of the *WFSetOAuthCredentials* service API:

Name	Parent Tag	Parameter Type	Description
MinCode	Root	String	Sucess

Error-codes and description

The following table describes the error codes that can occur while executing the *WFSetOAuthCredentials* API:

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_801	500	Internal Server Error	An SQL exception occurred which stopped the operation. Refer to the Error logs for further analysis.	SQL exception occurred which stopped the operation. Refer to the Error logs for further analysis.
NGONE_WFS_503	503	Service is either down or slow.	Service is either down or slow.	N/A
NGONE_WFS_503	503	Redis server is down.	Redis cache server is down.	N/A
NGONE_WFS_503	503	Service is either unavailable or slow.	Service is either unavailable or slow.	Service is either unavailable or slow.

Sample response

Below is the sample response body for *WFSetOAuthCredentials* API:

Request:

```
{
"AppId": "e82edd37-77fb-4ab4-ad84-ba2899356cc3",
"AppName": "NewgenONEDeafult_Auth",
"OAuthEnabled": "Y",
"ServerSidePublicKey":
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAiwA8ENLj
376CtH6NLqA+MKf4q3b1rhnhkaF71C8qRkUwe
6KSUKxZrt9QaK4yAfsycJYJeTD29mxShqPcAIOkK8JC8sIcnE2vBMshG2aX/
kOvnQ7i1ulLz9y0NLQPiq4we0IBxr ILP6BAHGCxhcHmXGKQOvgmcIfX/
NOvaIftkuXk8Qw12sbWs8birpe3hrBKLT26tBhm4i
UFocd7AR04+QSQ7iKUTLks 81yRgZ/hOADssJ+zG6FQkcNPwszJ1v/
ge551xA6GaOxxc0W92Ev0c7syx7YCfp
6Kq2E9BLqEZtOuE7Mxj5l01/RkRgmCUYg1IxLw3Aw2YS2RFDpt8xEaSQIDAQAB",
```

```
"ClientSidePrivateKey":
"MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCGQCqq
AGVULXRPBwQHt5kRfelOV6jK+
E+VDcKSXdqncYG7aK3oQF21D5LD7/F
fnVruaFO0tHCZd5fStBsYFS71dNz3OgXhCbYHammYR5HA7ZGTjNOsW2LwhT7 oabjlujYtK/VPB/h/
koQTSCXr0lXK+/67le23KWpocd9/d486CG8QgPGx0 ryC0q8pwU4SMgnZK7QTm/
jKU6F2YYSG90mCDyK792AzD95yaPDwBswM56Aw
vXECZ8mWhsHH9DtlWz4VFrwbfkxD9nLFez+OJ9XGkGUlHEQXm6c
1Y3AruSMA7nI7daIKfOnsPsVYxhHhNkk5fQoxLZiI0+xyawwbIQbrA gMBAAECggEAEvy+S/
70Vdx757VxT3obDwtE1Qy21cYa01T34OWfSDFYjN 34QrHfzvzd4BJui/
xgYPmmZRdkGYB9m5JXhqdwVczqd5jgPfy2nwPg3WW56pRvE3RGd8k RDD18MTjKqssUYH0oG75f2
gMp5OQ+YMUF10ietX81+MLFxvHcD1YkjqtwH7ddO/FIpX+1g
KFUI+RZNFNIRZ6ay3uGQNv0QhvcXnPg3Ib6/
xUnDmxDn5puA5H3zAU0DZE3M+BqP5GQEJx C4Wu4V0dpjkBkRRiz5NZYGnyrNH/
h9gqz4Ol1UuEks0DWtCWc
JX8paNmzNRsZFT5MZ5DecM4 Fb//slo5b4/
EguQKBgQDG9+wmmOf1tTxgXR6QshE2eVPDFAXEBiP+bZ0D9RH
spUY39eJF4UGn2k12FXdJP tgDeVOOD1f8HGDdPp2Qbice/fmaCKDh15N/n/
QFOjWDNGjSqkxDWn8FekDoGu
M56UZq55IvYyHQP8+ Mbpp855GMjm08z+8H52TC/3fjAxSklQKBgQCsu1OA54UmdBeJbM14/CJXkVK/
EOhAJJhR6CTVeAOC
amFdb7jsho24HHLPSd0MQ7DJRCHToJWKOPNDhLs9EKjFMwwugJL8uu+uePAuGv1fK3k/+jBMI/
ldVFsBeCUpL128Hye9zUu4GHL6FkTuUCKwaL239OHsoPRoNq/
QLiSdfwKBgCsQtFtyUxqanLPa0+fLsvAQeL HLnqfcEGRcg1S88aE/8J/
iJ5SqedSD6qo5YTVIVecClrQz2/5kgg7BI8eEIBGUNTdC0NexGmrQSzfqbnS0nk+
irpkReFbOLEhQsYfyVtyCvOBm24XAmgKyWdvZH2bF2R0FgJ48VlHii6dmHky1AoGAU55DqE1r6dedZUukl
IMKZPXW5uPaauLMUipdWnHE0SeKht8E2Iuyo4QyGCMbvUovlN9EuNwuSEGVkgP5WHBuIzi5FvHrNimQyhe
FvewW6DF1ewE0zEdF+mZRv/5Xjv59FArJBFsT4sUDNAn/yJZGlLu/zAPiMtkgOCrxJYiQ9SUCgYA9DL6P
Z34h02SmFRtJJ27H2gRx3FvdEdgQ8udq/nx9HYfU6fQffF60AvQ/FCiLwPmf6lWba1gDgmoxAap
7Nr6MdTxm1Al66P7o9/0r4ctOnlbU0EQ1XJwjeCiTHszhbtSE57PgXE2fESa6uwWnO47oAnHw
r1s5QYvGNIGaBmBSVA==",

"ClientSidePublicKey":
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAhkAqqgB
lVC10TwcEB7eZEX3pTleoyvhPlQ3Ckl3ap3GBu2it6EBdtQ+ Sw+/
xX51a7mhTtLRwmXeX0rQbGBUu9XTc9zoF4Qm2B2ppmEe
RwO2Rk4zTrFti8IU+6Gm45bo2LSv1Twf4f5KEE0gl69JVyvv+u5
XttylqaHHff3ePOghvEIDxsdK8gtKvKcFOEjIJ2Su0E5v4ylOhd mGEhvdJgg8iu/dgMw/
ecmjw8AbMDOegML1xAmfJlobBx/ Q7ZVs+FRa8G35MQ/ZyxXs/jifVxpBlJRxEF5unNWNwK7kjAO5y
 O3WiCnzp7D7FWMYR4TZJOX0KMS2YiNPscmsMGyEG6wIDAQAB"
}
```


Response


```
{
    "MainCode": 0
}
```

WFGetOAuthCredentials

This API is used to get OAuth credentials in WFAppDetails

- URL: *http://<GatewayIp: GatewayPort >/user-service-context/users/set-oauth*.
- Request Media Type: application/json
- Response Media Type: application/json
- Method Operation: POST

Header parameter description

The following table describes the header parameters of the *WFGetOAuthCredentials* service API:

Name	Parameter Scope	Type	Description	Mandatory
Accept-Language	HeaderParam	Locale	Indicates the natural language and locale that the client prefers. Example: en-US	No
Tenant-Id	HeaderParam	String	Name of the cabinet.	Yes
Authorization	HeaderParam	String	It is the signed-in user's session id.	Yes

Request body parameter

There are no request parameter for *WFGetOAuthCredentials* API:

Response body parameter

The following table describes the response body parameters of the *WFGetOAuthCredentials* service API:

Name	Parent Tag	Parameter Type	Description
AppId	Root	String	ID for the registered application.
AppName	Root	String	Name of the App for the registered application.
OAuthEnabled	Root	String	Signifies whether OAuth is enabled or not in the cabinet.
CreatedDateTime	Root	String	Signifies the time at which OAuth is enabled on the cabinet.
RevisedDateTime	Root	String	Signifies the time at which credentials are updated.
ServerSidePublicKey	Root	String	The public key for server operations is used in the RSA algorithm.
ClientSidePrivateKey	Root	String	Encrypted key for client-side
ClientSidePublicKeyRoot	Root	String	Public key for client-side operations

Error-codes and description

The following table describes the error codes that can occur while executing the *WFGetOAuthCredentials* API:

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_801	500	Internal Server Error	SQL exception occurred which stopped the operation. Refer to the Error logs for further analysis.	SQL exception occurred which stopped the operation. Refer to the Error logs for further analysis.
NGONE_WFS_503	503	Service is either down or slow.	Service is either down or slow.	N/A
NGONE_WFS_503	503	Redis server is down.	Redis cahce server is down.	N/A

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
NGONE_WFS_503	503	Service is either unavailable or slow.	Service is either unavailable or slow.	Service is either unavailable or slow.
NGONE_WFS_18	200	Ok	No more records	No more records.

Sample response

Below is the sample response body for *WFGetOAuthCredentials* API:

Response:


```
{
"AppId": "e82edd37-77fb-4ab4-ad84-ba2899356cc3",
"AppName": "NewgenONEDeafult_Auth",
"OAuthEnabled": "Y",
"ServerSidePublicKey":
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAiwA8ENLj
376CtH6NLqA+MKf4q3b1rhnhkaF71C8qRkUwe
6KSUKxZrt9QaK4yAfsycJYJeTD29mxShqPcAIOkK8JC8sIcnE2vBMshG2aX/
kOvnQ7i1ulLz9y0NLQPiq4we0IBxr ILP6BAHGCxhcHmXGKQOvgmcIfX/
NOvaIftkuXk8Qw12sbWs8birpe3hrBKLT26tBhm4i
UFocd7AR04+QSQ7iKUTLks 81yRgZ/hOADssJ+zG6FQkcNPwszJ1v/
ge551xA6GaOxxc0W92Ev0c7syx7YCfp
6Kq2E9BLqEZtOuE7Mxj5l01/RkRgmCUYg1IxLw3Aw2YS2RFDpt8xEaSQIDAQAB",

"ClientSidePrivateKey":
"MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCGQCqq
AGVULXRPBwQHt5kRfelOV6jK+
E+VDcKSXdqncYG7aK3oQF21D5LD7/F
fnVruaFO0tHCZd5fStBsYFS71dNz3OgXhCbYHammYR5HA7ZGTjNOsW2LwhT7 oabjlujYtK/VPB/h/
koQTSCXr0lXK+/67le23KWpocd9/d486CG8QgPGx0 ryC0q8pwU4SMgnZK7QTm/
jKU6F2YYSG90mCDyK792AzD95yaPDwBswM56Aw
vXECZ8mWhsHH9DtlWz4VFrwbfkxD9nLFez+OJ9XGkGUlHEQXm6c
1Y3AruSMA7nI7daIKfOnsPsVYxhHhNkk5fQoxLZiI0+xyawwbIQbrA gMBAAECggEAEvy+S/
70Vdx757VxT3obDwtE1Qy21cYa01T34OWfSDFYjN 34QrHfzvzd4BJui/
xgYPmmZRdkGYB9m5JXhqdwVczqd5jgPfy2nwPg3WW56pRvE3RGd8k RDD18MTjKqssUYH0oG75f2
gMp5OQ+YMUF10ietX81+MLFxvHcD1YkjqtwH7ddO/FIpX+1g
KFUI+RZNFNIRZ6ay3uGQNv0QhvcXnPg3Ib6/
xUnDmxDn5puA5H3zAU0DZE3M+BqP5GQEJx C4Wu4V0dpjkBkRRiz5NZYGnyrNH/
h9gqz4Ol1UuEks0DWtCWc
JX8paNmzNRsZFT5MZ5DecM4 Fb//slo5b4/
EguQKBgQDG9+wmmOf1tTxgXR6QshE2eVPDFAXEBiP+bZ0D9RH
spUY39eJF4UGn2k12FXdJP tgDeVOOD1f8HGDdPp2Qbice/fmaCKDh15N/n/
```

```
QFOjWDNGjSqkxDWn8FekDoGu
M56UZq55IvYyHQP8+ Mbpp855GMjm08z+8H52TC/3fjAxSklQKBgQCsu1OA54UmdBeJbM14/CJXkVK/
EOhAJJhR6CTVeAOC
amFdb7jsho24HHLPSd0MQ7DJRCHToJWKOPNDhLs9EKjFMwwugJL8uu+uePAuGv1fK3k/+jBMI/
ldVFsBeCUpL128Hye9zUu4GHL6FkTuUCKwaL239OHsoPRoNq/
QLiSdfwKBgCsQtFtyUxqanLPa0+fLsvAQeL HLnqfcEGRcg1S88aE/8J/
iJ5SqedSD6qo5YTVIVecClrQz2/5kgg7BI8eEIBGUNTdC0NexGmrQSzfqbnS0nk+
irpkReFbOLEhQsYfyVtyCvOBm24XAmgKyWdvZH2bF2R0FgJ48VlHii6dmHky1AoGAU55DqE1r6dedZUukl
IMKZPXW5uPaauLMUipdWnHE0SeKht8E2Iuyo4QyGCMbvUovlN9EuNwuSEGVkgP5WHBuIzi5FvHrNimQyhe
FvewW6DF1ewE0zEdF+mZRv/5Xjv59FArJBFsT4sUDNAn/yJZGlLu/zAPiMtkgOCrxJYiQ9SUCgYA9DL6P
Z34h02SmFRtJJ27H2gRx3FvdEdgQ8udq/nx9HYfU6fQffF60AvQ/FCiLwPmf6lWba1gDgmoxAap
7Nr6MdTxm1Al66P7o9/0r4ctOnlbU0EQ1XJwjeCiTHszhbtSE57PgXE2fESa6uwWnO47oAnHw
r1s5QYvGNIGaBmBSVA==",

"ClientSidePublicKey":
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAhkAqqgB
lVC10TwcEB7eZEX3pTleoyvhPlQ3Ckl3ap3GBu2it6EBdtQ+ Sw+/
xX51a7mhTtLRwmXeX0rQbGBUu9XTc9zoF4Qm2B2ppmEe
RwO2Rk4zTrFti8IU+6Gm45bo2LSv1Twf4f5KEE0gl69JVyvv+u5
XttylqaHHff3ePOghvEIDxsdK8gtKvKcFOEjIJ2Su0E5v4ylOhd mGEhvdJgg8iu/dgMw/
ecmjw8AbMDOegML1xAmfJlobBx/ Q7ZVs+FRa8G35MQ/ZyxXs/jifVxpBlJRxEF5unNWNwK7kjAO5y
 O3WiCnzp7D7FWMYR4TZJOX0KMS2YiNPscmsMGyEG6wIDAQAB"
}
```

Implementing hook

This chapter explains how to implement hooks in the workitem service using the provided framework.

Following are the prerequisites to implement the hook:

- Spring Boot: Version 3.1.6
- Java: Version 17
- Gradle: Version 8.4

Introduction to Hook

Hooks are a valuable tool for extending and customizing the functionality of the workitem service. It provides the ability to intercept and modify the request as well as the response body of the service before and after the execution of workitem service endpoint.

Hook implementation components

Hook implementation requires Configuration Service and Hook Service. Both services must be in a running state and deployed on the cloud.

Configuration Service

Configuration Service is a non-modifiable product service responsible for managing the configuration settings related to hooks. It allows developers to specify which hooks must be enabled. The configuration service provides endpoints through which developers or end users can interact with the configuration and their current status.

Hook Service

The Hook service is a modifiable product service for handling the execution of hooks during the execution of the workitem endpoint. This is applicable in case the hook is enabled for a particular endpoint (API). You need to modify the HookImplementation Class to implement a new Hook.

Hence, during the implementation of hooks, you must utilitise both the configuration and hook service. The configuration service allows you to define and manage the settings for the hooks, while the hook service is responsible for executing the hooks based on the configured settings.

To ensure proper configuration and functionality of the hook service, it is necessary to make specific changes in the *application.properties* file. These changes must be applied to the hook service.

The required key modifications are as follows:

- server.port — Update the server port configuration according to your requirements and environment. This ensures the hook service is accessible on the desired port.
- Active profile — Specify the active profile for hook service on the environment. For example, development, staging, production, database: Microsoft SQL, PostgreSQL, and Oracle to enable the appropriate configurations.
- Kafka server (IP/PORT) — Provide the IP address and port details of the Kafka server in the *application.properties* or corresponding yml files. This makes the connection between the microservices and Kafka for efficient message communication.
- Database or cabinet details — In the *msconfig > datasourceinfo.json* file, provide the required details such as the database name, credentials, and connection URL. This enables the hook-service framework to establish a connection with the specified database or cabinet.

Configuring the hook implementation

This section provides the hook implementation steps in the below categories:

- Enabling the hook execution using the configuration service
- Defining the hook execution method and their execution using the hook service

Enabling the hook execution using configuration service

This section consists of the following endpoints to enable the hook execution:

- get-available-configuration
- update-configuration

get-available-configuration

The *get-available-configuration* API allows you to retrieve all configurations along with their corresponding configuration types. This endpoint provides access to configuration information without any request body.

- Service URL — *http://<GatewayIp: GatewayPort>/configuration-service-context/configuration-service/get-available-configuration*

  Where, GatewayIp is the IP of the machine, and GatewayPort is the port of the machine where the configuration service is deployed.
  Example — *http://192.168.149.129:8765/configuration-service/get-available-configuration*
- Request Method — GET
- Response Media Type — application/json
- Request Body — NA

Header parameters description

The following table describes the header parameters of the *get-available-configurations* API:

Name	Type	Description	Mandatory
Authorization	String	The session ID of the signed-in user.	Yes
Accept-Language	Locale	Represents the specific region or language. Example: en-US	No
Tenant-Id	String	The name of the cabinet.	Yes

If OAuth is enabled:

Name	Parameter scope	Type	Description	Mandatory
AppId	HeaderParam	String	Registered AppId of the product	No
Access Token	HeaderParam	String	AccessToken for the signed-in user	No
SessionValid	HeaderParam	String	SessionValid for the signed-in user	No

Response body parameters description

The following table describes the response body parameters of the *get-available-configuration* API:

Name	Parent Tag	Type	Description
ConfigType	Root	String	Specifies the type or category of the configuration.
ConfigurationName	ConfigType	String	Indicates the name or identifier of the configuration.
ConfigurationValue	ConfigType	String	Represents the actual value (Status) associated with the configuration.

Name	Parent Tag	Type	Description
PreHookMethodName	ConfigType	String	Specifies the name of the pre-hook method associated with the configuration. This parameter is applicable only for the hook configuration type.
PostHookMethodName	ConfigType	String	Specifies the name of the post-hook method associated with the configuration. This parameter is applicable only for the hook configuration type.

Sample response body

Use the configurations received from *get-available-configuration* endpoint to the update configuration endpoint. Hence, you need to use this endpoint to gather all the configurations.

```
{
  "Hook": [                //Here,Hook is the type of configuration
    {
      "ConfigurationName": "uploadworkitem",
      "ConfigurationValue": "Y",
      "PreHookMethodName": "preUploadWorkItem",
      "PostHookMethodName": "postUploadWorkItem"
    },
    {
      "ConfigurationName": "getworkitem",
      "ConfigurationValue": "N",
      "PreHookMethodName": "preGetWorkitem",
      "PostHookMethodName": "postGetWorkitem"
    }
  ]
}
```

Error codes and description

The following table describes the error codes that can occur while executing the *get-available-configuration* API:

Error Code	HTTP Status	HTTP Status Description	Error Code Description	Comments
N/A	401	Unauthorized	Invalid Session	Regenerate the session and pass the valid session ID in the authorization header.

update-configuration

The *update-configuration* API allows you to update the configuration values (status) associated with their respective configuration names and types. It allows you to modify the status of configurations by specifying the appropriate configuration name and type in the request. This endpoint allows you to update configuration values based on their specific requirements.

- Service URL — *http://<GatewayIp: GatewayPort>/configuration-service-context/configuration-service/update-configurations*

  Where, GatewayIp is the IP of the machine, and GatewayPort is the port of the machine where the configuration service is deployed.
  Example — *http://192.168.149.129:8765/configuration-service/update-configuration*
- Request Media Type — application/json
- Response Media Type — application/json
- Method Type — POST

Header parameters description

The following table describes the header parameters of the *update-configuration* API:

Name	Type	Description	Mandatory
Authorization	String	The session ID of the signed-in user.	Yes
Accept-Language	Locale	Represents the specific region or language. Example: en-US	No

Name	Type	Description	Mandatory
Tenant-Id	String	The name of the cabinet.	Yes

If OAuth is enabled:

Name	Parameter scope	Type	Description	Mandatory
AppId	HeaderParam	String	Registered AppId of the product	No
Access Token	HeaderParam	String	AccessToken for the signed-in user	No
SessionValid	HeaderParam	String	SessionValid for the signed-in user	No

Request body parameters description

The following table describes the request body parameters of the *update-configuration* API:

Name	Parent Tag	Type	Description	Default Values	Valid Values	Mandatory
ConfigType	Root	String	Allows you to enable or disable the hook. The configuration type must be set as hook. • The hook must match the configuration type received while fetching data from the get-available-configuration endpoint.	NA		Yes
ConfigurationName	Root	String	Refers to the workitem API name if the configuration type is hook. • Value must be matched with the configuration name received while fetching data from the get-available-configuration endpoint.	NA		Yes
ConfigurationValue	Root	String	Indicates the status of the hook that is, enabled or disabled.	N	Y – To enable the hook N – To disable the hook	Yes

Response body parameters description

The following table describes the response body parameters of the *update-configuration* API:

Name	Parent Tag	Type	Description
ConfigType	Root	String	Specifies the type or category of the configuration.
ConfigurationName	ConfigType	String	Indicates the name or identifier of the configuration.
ConfigurationValue	ConfigType	String	Represents the actual value (Status) associated with the configuration.

Name	Parent Tag	Type	Description
PreHookMethodName	ConfigType	String	Specifies the name of the pre-hook method associated with the configuration.
PostHookMethodName	ConfigType	String	Indicates the name of the post-hook method associated with the configuration.

Sample request and response body

Request:

```
{
     "ConfigType":"Hook",
    "ConfigurationName": "getworkitem",
    "ConfigurationValue":"Y"
}
```

Response:

```
{
  "Hook": [                //Here, Hook is the type of configuration
    {
      "ConfigurationName": "uploadworkitem",
      "ConfigurationValue": "Y",
      "PreHookMethodName": "preUploadWorkItem",
      "PostHookMethodName": "postUploadWorkItem"
    },
    {
      "ConfigurationName": "getworkitem",
      "ConfigurationValue": "Y",
      "PreHookMethodName": "preGetWorkitem",
      "PostHookMethodName": "postGetWorkitem"
    }
  ]
}
```

Defining the hook method and their execution using hook service

Deploy the hook service war file in the environment to facilitate the hook implementation for the workitem service. In this environment, you can define the hook methods as per the requirements. It enables you to incorporate custom logic and operations through the hooks implementation in the workitem service.

The Hook service includes a Java class file named *HookImplementation* where you must define the pre-hook method and post-hook method with the below-defined signatures. By adhering to these specific method signatures, you can ensure that the hook methods are correctly implemented and compatible with the hook service.

Pre hook method signature

This section describes the Pre hook method along with its parameters.

```
public WFHookProperties preGetWorkitem (WFHookProperties hookProperties) {
        String requestBody=hookProperties.getRequestBody();
        JSONObject requestJson=new JSONObject(requestBody);
        requestJson.put("QueueId", 140);
        hookProperties.setRequestBody(requestJson.toString());
        return hookProperties;
    }
```

The pre-hook method name is the same as the data returned in the "get-available-configuration" endpoint.

"PreHookMethodName": "preGetWorkitem" // Data returned in configuration service

The return type of the method must be the *WFHookProperties* class, where you can update the payload. This payload serves as the input for the endpoint for which the hook gets executed. By modifying the payload within the *WFHookProperties* object, you can customize the data that is sent as input to the endpoint during the execution of the hook. This method consumes a single argument, which is an object of the *WFHookProperties* class defined in the hook service.

The Pre hook method consumes a single argument, which is an object of the WFHookProperties class defined in the hook service. The WFHookProperties class has two parameters:

- httpHeaders — This parameter is of type Map and contains all the headers passed in the request. The httpHeaders parameter can be used to access and modify the request headers, but it cannot be updated for the request.
- requestBody — This parameter holds the request body or payload. The request body can be obtained using the getRequestBody method and can be updated using the setRequestBody method.
  By utilizing the WFHookProperties object, you can access the request headers and manipulate the request body within the pre-hook or post-hook methods, allowing for customization and modification as needed for the specific hook implementation.
  - You can implement the custom logic to modify or update the request body.
  - Pre hook method gets invoked before the execution of the requested workitem API or endpoint.

Post hook method signature

This section describes the Post hook method signature.

```
public WFHookProperties postGetWorkitem (WFHookProperties hookProperties) {
        String requestBody=hookProperties.getRequestBody();
        JSONObject requestJson=new JSONObject(requestBody);
        requestJson.put("QueueId", 160);
        hookProperties.setRequestBody(requestJson.toString());
        return hookProperties;

    }
```

- The Post hook Method invokes after the execution of the requested endpoint or workitem API.
- The Post hook method receives the response body of the executed workitem API or endpoint.
- The Post-hook method name must be same as the data returned in the "get-available-configurations" endpoint.

"PostHookMethodName": "postGetWorkitem" // Data returned in configuration service

The return type of the method must be the WFHookProperties class, where you can update the response. This response served as the output for the endpoint for which the hook gets executed. By modifying the response within the WFHookProperties object, you can customize the data that are sent as output to the endpoint during the execution of the hook.

- This method consumes a single argument, which is an object of the WFHookProperties class defined in the hook service. The WFHookProperties class has two parameters:
  1. httpHeaders: This parameter is of type Map and contains all the headers passed in the request. The httpHeaders parameter can be used to access and modify the request headers, but it cannot be updated for the request.
  2. requestBody: This parameter holds the request body or payload. The request body can be obtained using the getRequestBody method and can be updated using the setRequestBody method.
     By utilizing the WFHookProperties object, developers can access and manipulate the request headers and body within the pre-hook or post-hook methods, allowing for customization and modification as needed for the specific hook implementation. You can implement the custom logic to modify or update the request body.
  3. After defining the pre-hook method and post-hook method in the HookImplementation class, you must create the class file for this implementation. Then, replace the existing file in the *com.newgen.hook.implementation* package of the hook service with the newly created class file. This ensures that the updated hook implementation is incorporated into the hook service, allowing for the execution of the defined hook method during the processing of requests.
  4. After replacing the file in the appropriate package, restart the container or server where the hook service is running. This restart applies the changes to the hook implementation class and reflects in the running instance of the hook service. Once the container restarts, it utilizes the updated hook implementation, ensuring that the modified pre-hook and post-hook methods get executed.
  5. Once the hook service is up and running, you can execute the hook-enabled workitem API or endpoint. After the workitem API execution, the pre-hook

method and post-hook method gets executed. This allows you to incorporate their custom logic and operations before and after the execution of the workitem API. The hook methods allows you to extend or modify the behavior of the workitem API, enhancing its functionality and allowing for additional processing or to perform actions during the request processing lifecycle.

For example, if you have enabled the hook for the *getWorkItem* API and defined the corresponding hook methods, then you can execute a request for the *getWorkItem* API. Upon executing the request, the hook methods trigger and execute as part of the request processing flow. This allows you to inject custom logic and perform specific actions before and after the *getWorkItem* API execution. The hook methods provide an opportunity to modify the request of the *getWorkItem* API.

The error handling mechanism throws an error if any error occurs during the hook execution for the workitem service endpoint or invocation of the hook method, leading to the termination of the requested endpoint's execution. Also, this mechanism handles any issues or exceptions that occur during the hook execution and prevents further processing of the requested endpoint. This ensures system reliability and facilitates timely error resolution.

Configuring the custom microservices

This chapter describes how to configure a custom microservices framework.

The following are the prerequisites to configure and deploy the custom microservices:

- Spring Boot: Version 3.1.6
- Java: Version 17
- Gradle: Version 8.4
- Apache Tomcat: Version 10 or above
- Apache Kafka: latest version

The custom-microservice framework streamlines the process of creating new custom APIs. With this framework, there's no need to create individual controller methods for each new custom API. Instead, there's a global controller for the service, which handles various types of requests such as GET, PUT, POST, and DELETE.

This global controller accepts the multiple HeaderParams, RequestParams, RequestBody (in JSON format), and up to 5 path variables. Using this, you can handle different types of requests and effectively manage the associated parameters.

Related topic(s)

- Binary files overview
- JAR files configurations

Binary files overview

The Binary files play a crucial role in the overall functionality and architecture of the custom-microservice framework.

The JAR files to configure the custom microservices are as follows:

- **custom-service.war** — This WAR file includes the core functionalities of the custom service. It includes a global controller to accept various types of requests, such as GET, PUT, POST, and DELETE. This global controller acts as a centralized entry point for incoming requests.
  After receiving the request, the controller delegates control to the Java class(CustomControllerEndPoints.java), which contains the implementation of the custom API. Further, you can send the control to the service layer and the Data Access Object (DAO) layer.This structure handles the requests processing and their execution effectively. Then, the controller in the custom-service.jar incorporates the Circuit Breaker functionality, providing an additional layer of reliability and resilience. This feature enables the system to manage failure in case of issues with external dependencies.
- **custom-logging-service.jar** — This JAR file manages the logging requests, responses, and errors that occur within the custom service. It facilitates efficient tracking and troubleshooting of any issues that arise during the service's operation.
- **custom-gateway.jar** — This JAR file acts as the gateway for incoming requests, routing them to the appropriate services and managing communication between various microservices. It enhances the security and scalability of the overall architecture.

- Deploy all the *WAR* files on the Apache Tomcat server.
- Execute the below command to run all the *JAR* files separately:
  `java -jar war_name.war`

JAR files configurations

This section explains the JAR file configurations required in the *application.properties* file.

Specify the following configurations:

- server.port — Update the server port configuration as per your environment requirements. This ensures that the JAR files are accessible on the desired port.

- active profile — Specify the active profile for each JAR file based on the environment. For example, development, staging, production, Database: MSSQL, PostgreSQL, and Oracle. This enables the appropriate configurations.
- Kafka server (IP/PORT) — Provide the IP address and Port details of the Kafka server in the *application.properties* files. This establishes the connection between the microservices and Kafka for efficient message communication.
- Redis cache (IP/PORT) — Configure the IP address and Port of the Redis cache in the *application.properties* file to enable caching functionality within the custom-microservice framework.
- URL of user-service in custom-gateway — Update the URL of the user-service in the custom-gateway's *application.properties* file. This is necessary for calling the *validateSession* API through the OmniDocs microservice.
- Database or cabinet details — In the **msconfig> datasourceinfo.json** file, provide the required details such as the database name, credentials, and connection URL. This enables the custom-microservice framework to establish a connection with the specified database or cabinet.

You can also add any additional properties or configurations in the *application.properties* file based on the specific requirements. This allows for customization and adaptation of the custom-microservice framework to cater to individual project needs and preferences.

If you want to add a new package, add the package name as *com.newgen.custom.{name-of-package}*. Then, add the class files of the same package in the *custom-service.jar*.

Related topic(s)

- custom-service-workspace
- custom-logging-service
- custom-gateway-workspace

---

custom-service-workspace

The custom-service-workspace allows you to change or modify the *custom-service.war* file.

To configure the sending request in custom service, perform the below steps:

1. Include the following header in the request:
   Tenant-Id — Keep this header value as the name of the cabinet with which user wants to interact. This header is used to establish a connection with the specified cabinet name.

2. msconfig file — This configuration is used to establish a connection that requires a *datasourceinfo.json* file located in the msconfig folder. This JSON file contains the necessary details of the cabinet mentioned in the Tenant-Id header. The msconfig folder, along with the datasourceinfo.json file, must be placed parallel to the *custom-service.jar* file. This allows the service to read the JSON file and retrieve the cabinet details to establish a JDBC connection.

   For the mssql cabinets, add *;encrypt=false* at the end of URL.
   For example: *jdbc:sqlserver://loggingdev.database.windows.net:1433;database=newgenonemssql16nov;encrypt=false*

3. com.service.name — This tag exists in the *application.properties* file and specifies the custom service name.
4. Use this URL structure *http://<IP>:<PORT>/<war-name>/<com.service.name>/<TypeOfHttpMethod>/{api-name}* to interact with the custom-microservice framework.

To use the URL structure, perform the below steps:

1. Replace the *<IP>* and *<PORT>* in the URL with the IP address and the Port number of the server hosting the microservice.

   It ensures that the request is directed to the intended microservice.

2. In the *<war-name>* tag, enter the WAR file name of the deployed service in the Tomcat server.
3. In the *<com.service.name>* tag, enter the custom service name as specified in the *com.service.name* tag of the *application.properties* file.
4. In the *<TypeOfHttpMethod>* tag, enter the http method type. The values are as follows:
   - get — Retrieve the data from the microservice.
   - put — Update or replace existing data within the microservice.
   - post — Create the new data or resources within the microservice.
   - delete — Removes the data or resources from the microservice.

> The tag defines the HTTP method for the request and helps determine the operation need to perform on the microservice. Hence, selecting the correct HTTP method in place of *<TypeOfHttpMethod>* is necessary.

5. Specify the {api-name} in the URL.
6. Pass the path variables up to five in the URL and access them variables from the path parameters mapping with the keys. The parameters with keys are:
   - param1
   - param2
   - param3
   - param4
   - param5

> Within the custom service, the controller manages the processing of the API requests. It reads the API name from the URL and redirects the control to the corresponding method in the CustomControllerEndPoints with the same {api-name}. For more information, refer to the CustomControllerEndPoint.java section.

For Example, *http://192.168.0.1:8080/custom-service/post/updateUserList*.

The above example follows the URL format: h*ttp://<IP:PORT/custom-service/<TypeOfHttpMethod>/{api-name}*

Here,

- IP address: 192.168.0.1
- Port number: 8080
- <TypeOfHttpMethod>: post
- <api-name>: updateUserList

In the custom-service-workspace, there are some exposed files that you can access and modify as needed. You must replace these files in the original JAR of the custom service. Also, you can create new packages within the project and include them in the original JAR file.

Configuring the exposed files in workspace

This section consists of the following:

- CustomControllerEndPoint.java
- com.newgen.custom.exception
- application.properties
- CustomServiceMessages.properties

Configuring the CustomControllerEndPoint.java file

This section describes the role of CustomControllerEndPoints in the custom-service-workspace.

- The controller reads the name of the API from the request URL using the {api-name} tag and directs the control to the corresponding method in the **CustomControllerEndPoints class** with the same name.
- Within the **CustomControllerEndPoints** class, you can handle the request by sending the control to a separate Service class. To achieve this, autowire the appropriate service class within the **CustomControllerEndPoints** class.
- This controller includes the following:
  - The Naming convention for the API implementation method and the parameters
  - Populating the request body from the existing JSON string in the method parameter
  - Returning a response from the custom API

The Naming convention for the API implementation method and the parameters

The name of the method must be same as the API name sent in the request URL.

In the CustomControllerEndPoints class, design the API implementation including the following parameters:

- Request Parameters
- Header Parameters
- Query Parameters
- Path Variables

> Ensure that the method defined in this class must have the return type as Object.

For example, in the case of a GET request, the method signature must be as follows:

public Object {api-name}(Map<String, String> pathParam, HttpHeaders headerParam,
HashMap<String, Object> queryParam)

The above method has the following parameters:

- **pathParam** — A map containing the path variables passed by the user.
- **headerParam** — The HttpHeaders object containing the header parameters of the request.
- **queryParam** — A HashMap represents the query parameters passed by the user.

Similarly, In the method implementation, the **POST**, **PUT**, and **DELETE** types of APIs must have the request body and method parameters that are as follows:

```
public Object {api-name}(Map<String, String> pathParam, HttpHeaders headerParam,
HashMap<String, Object> queryParam, String requestBody)
```

This method includes an additional parameter:

requestBody — Represents a JSON String(request body of the API).

For example, the request type is post, and the name of the API is registerEmployee. So, the API implementation in the CustomControllerEndPoint is as follows:

```
public Object registerEmployee(Map<String, String> pathParam, HttpHeaders
headerParam,
HashMap<String, Object> queryParam, String employeeDetails)
```

Populating the request body from the existing JSON string in the method parameter

To populate the request body from the Java class, use the following approach:

- In the code snippet A, the **ObjectMapper** class maps the data from the request body to Snippet B.

```
ObjectMapper mapper = new ObjectMapper();
```

Snippet A

```
Employee newEmployeeDetails = mapper.readValue(requestBody, Employee.class);
```

Snippet B

- In the code snippet B, the 'readValue' method takes the request body and the target class ('Employee.class') as parameters and converts the JSON data in the request body into a Java object.

Using this approach, you can effectively extract and utilize the data provided in the request body within their method implementation in the *CustomControllerEndPoints* class.

Returning a response from the custom API

To return a response from the custom API, you can construct the response body as per the business logic and data processing requirements of the API in the service layer. Once done, it returns as an object through the respective method in the *CustomControllerEndPoints* class.

Configuring the com.newgen.custom.exception package

This section describes the CustomServiceException as well as its uses.

The CustomServiceException is a global exception handler in the custom-service framework responsible for catching and parsing different types of exceptions. This package allows you to throw custom or generic errors from their business layer.

An overview of it's uses is as follows:

- Handling generic Exception — Refers to a generic exception occurrence. Here, the control reaches to CustomServiceExceptionHandler class and it prepares an exception message for the occurred error.
- Throwing custom errors from the service layer:
  - Create an instance of the CustomServiceException object to throw new errors.
  - Specify the error details, including the main code, sub-error code, error type, subject, and description.
  - Retrieve the subject and description from the *CustomServiceMessages.properties* file by auto wiring a MessageSource object.
  - @Autowired private MessageSource messageSource;

```
throw new CustomServiceException(
    CustomExceptionConstant.OPERATION_FAILED,
    CustomExceptionConstant.NO_AUTHORIZATION,
    messageSource.getMessage(CustomExceptionConstantKey.TEMPORARY,
null,
    CustomExceptionConstantKey.TEMPORARY, locale),

messageSource.getMessage(CustomExceptionConstantKey.OPERATION_FAILED,
null, locale),

messageSource.getMessage(CustomExceptionConstantKey.NO_AUTHORIZATION,
null, locale),
    HttpStatus.NOT_FOUND
);
```

The CustomServiceException constructor has the following parameters:

1. mainCode (int):
   - Represents the main code or main error identifier for the exception.
   - Helps categorize the type of exception or error that occurred.
   - Indicates an integer value.
2. subErrorCode (int):
   - Represents the sub-error code or a more specific error identifier for the exception.
   - Provides additional details about the specific error within the main code category.

- Indicates an integer value.

3. typeOfError (String):
   - Specifies the type of error that occurred.
   - Helps in understanding the nature or classification of the error.
   - It can be a custom-defined string value, such as temporary, critical, validation, and more.
4. subject (String):
   - Represents the subject or key of the error that occurred.
   - Identifies the error message from a properties file or resource bundle that contains localized error messages.
   - Displays the actual error message corresponding to the retrieved subject from the properties file based on the application's locale or language.
5. description (String):
   - Provides a detailed description or additional information about the error.
   - Indicates a key used to retrieve the error description from a properties file or resource bundle. It is similar to the subject.
   - The description is also typically localized based on the application's locale.
6. httpStatusCode (HttpStatus):
   - Specifies the HTTP status code associated with the exception.
   - Determines the status code return to the client when this exception is encountered.
   - Provides a range of pre-defined status codes such as OK, NOT_FOUND, INTERNAL_SERVER_ERROR, and more.

By using these parameters in the **CustomServiceException** constructor, you can create an instance of the exception with specific error details, allowing you to handle and communicate the error in a standardized and consistent manner.

Configuring the application.properties file

In this file, you can provide customization for various aspects of the application.

Configuring the CustomServiceMessages.properties file

The *CustomServiceMessages.properties* file consists of the error messages for specific errors that can occur. A key-value pair represents every error message, where the key represents the error name and the value represents the particular error description.

To use these error messages, add the key-value pairs in the *CustomServiceMessages.properties* file. Also, ensure that the key and value name must be the same as mentioned in the *CustomExceptionConstantKey* class file. This allows the messageSource object to retrieve the error message value from the *CustomServiceMessages.properties* file based on the error name.

Follow this approach to maintain a centralized location for error messages as it becomes easier to manage and customize them as per your application's requirements.

For example, if you want to display a custom exception with a specific message, you must define the key in the *CustomMessages.properties* file as follows:
OPERATION_FAILED=Operation Failed for the request

To retrieve and use this message, you can use the following code:

```
messageSource.getMessage(CustomExceptionConstantKey.OPERATION_FAILED, null,
locale)
```

custom-logging-service.jar

Configuration requires for custom-logging-service.jar file is as follows:

1. nglog — The specified folder containing the configuration to define folder structure that stores logs related to custom-service requests. The logs are:
   - request
   - response
   - error logs
   - transaction logs

   Keep this folder parallel to the *custom-logging-service.jar* file to ensure proper functioning and logging of the custom-service.

2. In the *custom-logging-service.jar* file, modify the following properties in the *application.properties* file:
   - server.port — Specify the port on which the application works. For example, server.port=8088.
   - newgen.loggerTopicName — Set the name of the logger topic. This is a property-specific application and you can customize this accordingly. For example, newgen.loggerTopicName=loggerTopic.
   - Kafka configuration — In the *application.properties* file, define the properties related to the Kafka setup, such as the bootstrap servers. For example, spring.kafka.bootstrap-servers=192.168.156.90:9092.

After making the necessary changes in the *application.properties* file, replace this in the JAR file. Once done, the updated JAR file gets deployed and reflects the new configuration.

custom-gateway-workspace

This section consists of the following:

- com.newgen.gateway.routes.impl.WFRouteConfigKubernetes
- application.properties

- Make the changes only in the *WFRouteConfigKubernetes* and *application.properties* files within the *custom-gateway.jar* file.
- Avoid modifying the other files within the *custom-gateway.jar* to prevent unintended issues or conflicts with the existing functionality of the gateway service.
- In case of any changes or to customize other aspects of the custom-gateway, contact the Newgen support team.

Configuring the WFRouteConfigKubernetes

To add the multiple routes in the gateway *com.newgen.gateway.routes.impl.WFRouteConfigKubernetes*, make the following changes in this file:

For Example,

```
@Value("${newgen.route.sampleServiceURI}")
private String sampleServiceURI;
@Bean
public RouteLocator routes(RouteLocatorBuilder builder) {
return builder.routes().route("<service-name>”,r->r.path(“/<war-name>/<service-
controller-path>/**).filters(f->
f.filter(validateSession)).uri(sampleServiceURI)).build();
}
```

To configure the route properly, perform the below steps:

1. Replace the *<war-name>* with the name of the war for the service.
2. Replace the *<service-name>* with the actual name of the service to which the request needs to be routed. For example, if it is the "company service," use "company-service" as the route name.
3. Replace the *<service-controller-path>* with the controller mapping URL specific to the service. For instance, if the company service has a controller mapping URL of "/company/**", replace it accordingly.

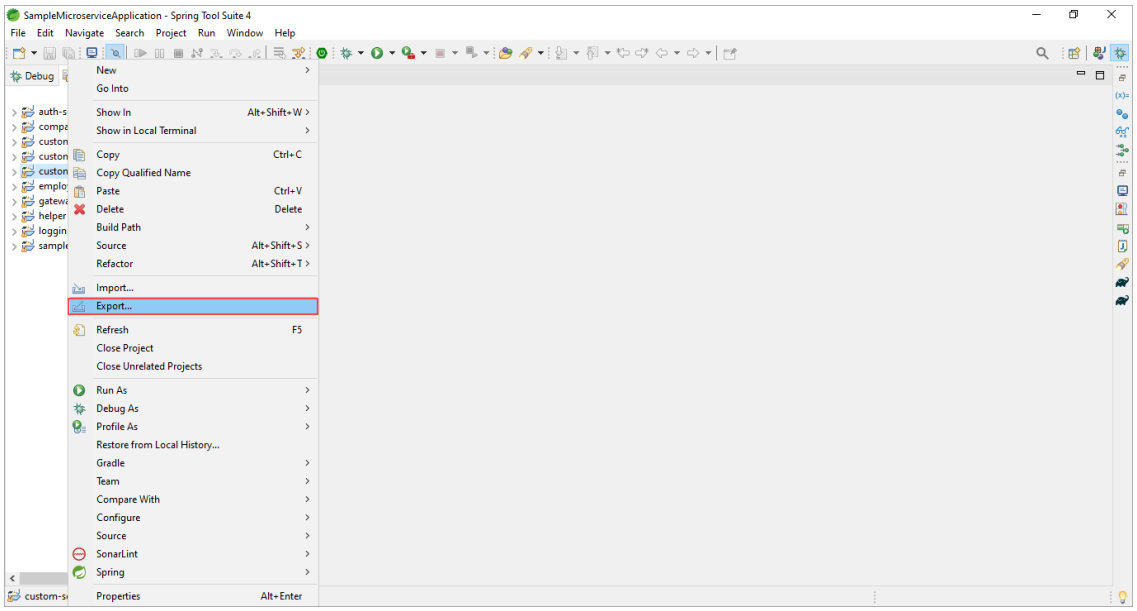
application.properties

Add the new routes with the prefix *newgen.route* followed by the URI name and assign its value.
For Example, *newgen.route.sampleServiceURI=http://192.0.101.1:8282/*

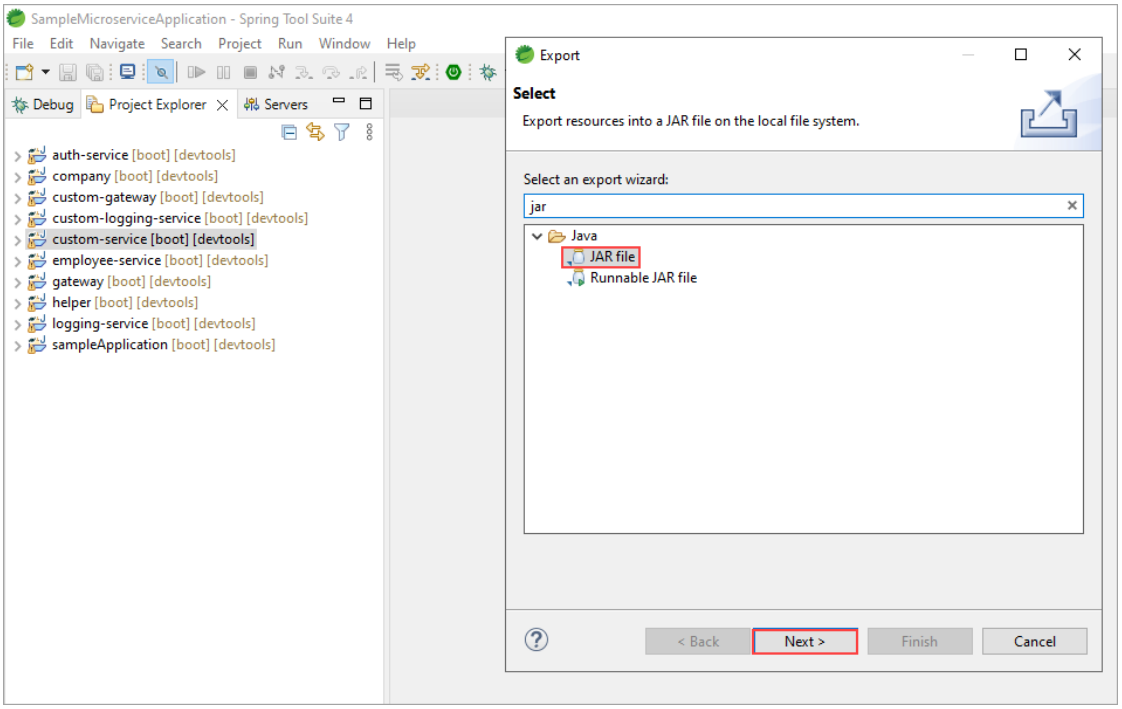
Deploying code changes

To deploy code changes using a JAR file, perform the below steps:

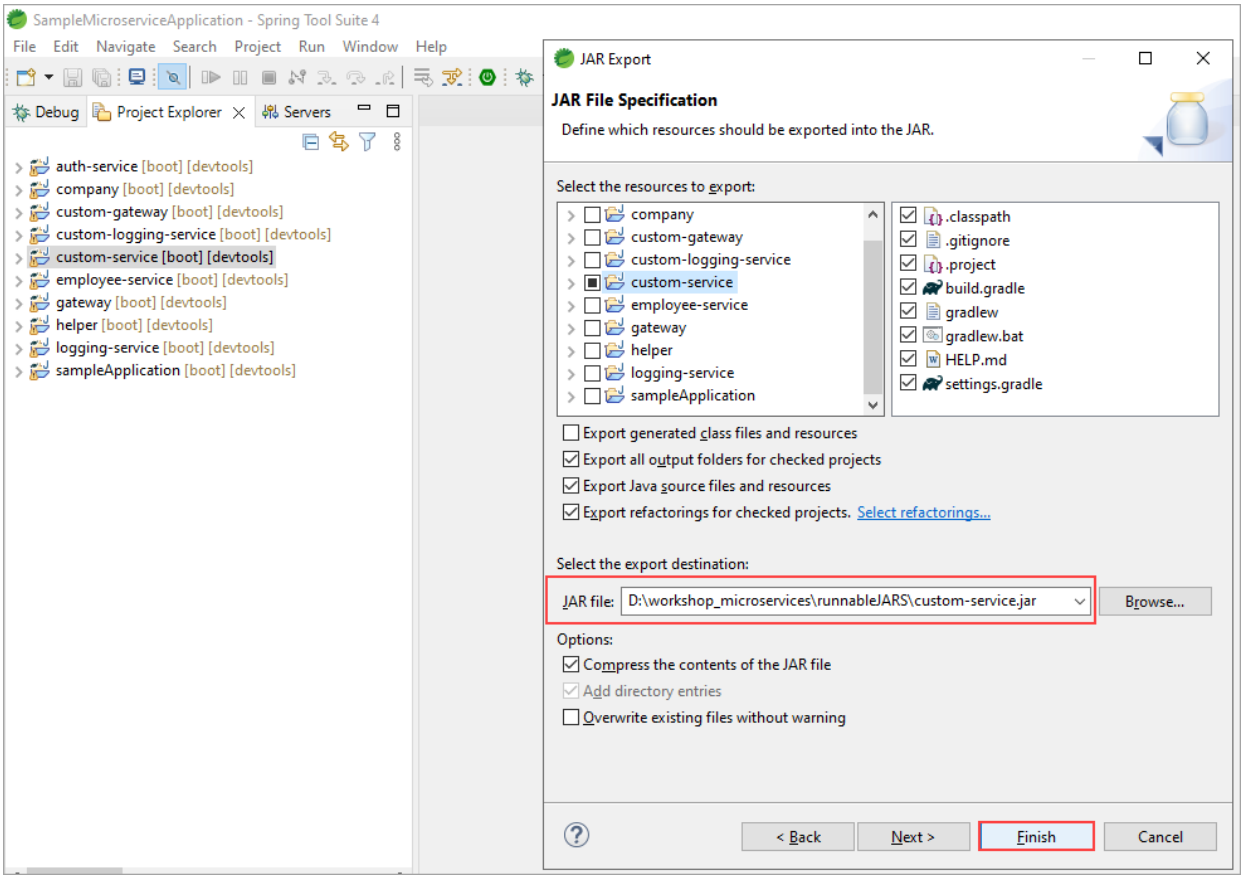
1. After making the changes in the provided workspace, navigate to the project.
   a. Right-click on the project and select **Export**.



   b. Select the JAR file and click **Next**.



c. Enter the location and name of the JAR file. For example, *d:\custom_classes.jar*.
d. Once done, then click **Finish**.



2. Replace the class files in their JAR file of custom-service or custom-gateway with the updated JAR file (custom_classes.jar).
    a. Locate the original JAR file (custom-service.jar or custom-gateway.jar).
    b. Replace the class files inside the JAR file with the updated class files from custom_classes.jar. You can use a file archiving tool like WinRAR or 7-Zip to open and modify the JAR file.
3. Run the JAR file using the command:
    a. Open the command prompt.
    b. Navigate to the directory where the JAR file (custom-service.jar or custom-gateway.jar) exists.
    c. Execute the command: *java -jar custom-service.jar* (replace custom-service.jar with the appropriate JAR file name).

Running the WAR files

To run the WAR files using apache tomcat server, perform the below steps:

1. Install the Apache Tomcat:
    a. Download and install the Apache Tomcat server using below URL: *https://tomcat.apache.org/*
    b. Select the appropriate version for your operating system. The latest version is *https://tomcat.apache.org/download-10.cgi* .

    > For Windows, download the 64-bit Windows core apache tomcat server.

2. Deploy the war file:
    a. Copy the war file that you want to run into the **webapps** directory of your Apache Tomcat installation.
    For example, if your Apache Tomcat installation directory is */opt/tomcat*, then copy the war file to */opt/tomcat/webapps*.
    b. Place the service-specific library files in the Tomcat server's *lib* folder.
    c. Ensure that the *lib* folder within the WAR file is empty so that the WAR files can read the dependencies from the common *lib* folder of the Tomcat server.
    d. Copy and paste the *msconfig* folder containing the *datasource.json* file into the *bin* folder of the Apache tomcat.
    e. Copy and paste the *nglog* folder into the *bin* directory of the Apache Tomcat server. This folder contains the configuration for generating logs in the respective folders under nglog > log > Tenant-Id (cabinetName).
3. Start the Apache Tomcat server:
    a. Open a terminal or command prompt.
    b. Navigate to the *bin* directory of your Apache Tomcat installation.
    c. Open the *catalina.bat* file and add path of *JAVA_HOME.*
    - For example : You can set the *JAVA_HOME=C:\Program Files\Java\jdk-17.0.5* and run the startup script using the below command:
        - Unix or Linux command: `./catalina.sh run`
        - Windows command: `catalina.bat run`
4. Monitor the server startup:
    a. After starting the server, monitor the logs for any errors or warnings.
    b. Open the logs located in the **logs** directory of your Apache Tomcat installation to view the server's output. The primary log file is:
    - For Unix or Linux: `catalina.out`

- For Windows: `catalina.bat`

5. Access the deployed application:
   a. Once the server has started without any errors, access the deployed application.
   b. Open a web browser and enter the URL: *http://localhost:8080/your-war-file-name*
   c. Replace the *your-war-file-name* with the actual name of your war file without the *.war* extension.

Enabling debug mode in Apache Tomcat server

To enable the debug mode in Apache Tomcat server, perform the below steps:

1. Open the *catalina.sh* (Unix/Linux) or *catalina.bat* (Windows) file located in the *bin* directory of your Tomcat installation.
2. Search for the *JAVA_OPTS* or *CATALINA_OPTS* environment variable configuration section.
3. Uncomment or add the below code line to enable remote debugging:

```
set "JAVA_OPTS=%JAVA_OPTS%
-agentlib:jdwp=transport=dt_socket,server=y,address=<debug-port>,suspend=n"
```

   a. Replace the *<debug-port>* with the port number for remote debugging. For example, set the port number to **8000**.
   b. In case you are using CATALINA_OPTS instead of JPDA_OPTS, modify the code line accordingly:

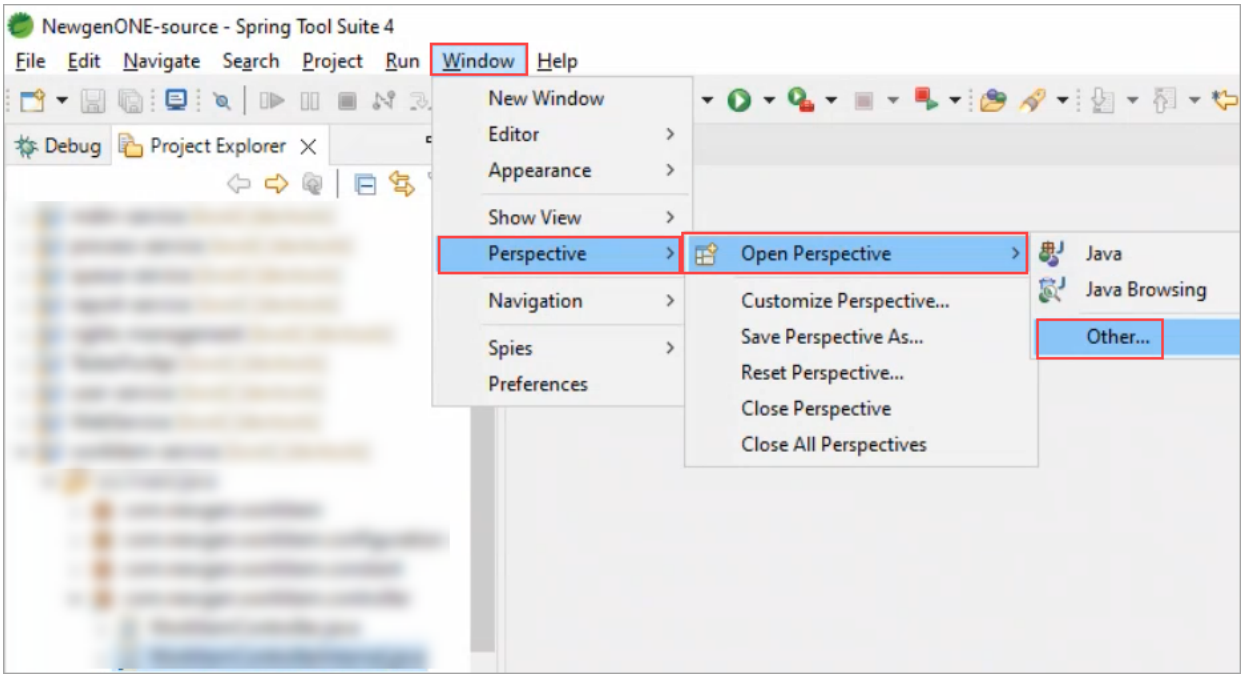
```
CATALINA_OPTS="-
agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=<debug-port>
```

4. Save the changes to the *catalina.sh* or *catalina.bat* file.
5. Start the tomcat server. Your Tomcat server is now running with debugging enabled on the specified port.
6. To connect to the Tomcat server for debugging, use an Integrated Development Environment (IDE) such as IntelliJ IDEA, Eclipse, or Visual Studio Code.
7. Configure a remote debugging session in your IDE with specified host and port of the Tomcat server.

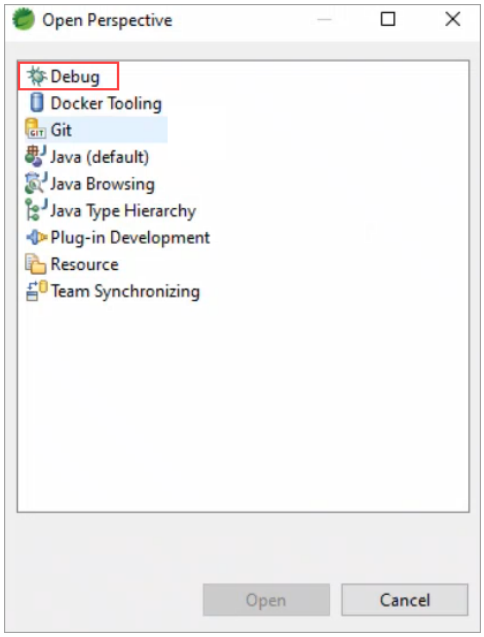
Debugging a JAR file remotely

To debug a JAR file remotely, perform the below steps:

1. To run the jar in debug mode execute the below command :
   java -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8787 -jar custom-service.jar
   This command enables the debug mode for the JAR file and listens on the port 8787 for remote debugging connections.
2. Open the custom-service-workspace in the SpringToolSuite.
3. Click the toolbar menu and go to the **Debug Configurations**. Also, you can use the shortcut.
4. Double-click on Remote Java Application to create a new configuration.
5. Enter the 8787 as the port number and click **Debug** to start the debugging session.
6. If the debugging session doesn't start automatically:
   a. Go to the **Window** tab. The dropdown list appears.



   b. Select **Perspective**>**Open Perspective**.



c. Select **Debug** and click **Open**.

The Debug gets enabled. Now, attach the breakpoints in your code within the custom-service-workspace and start debugging the code as required.

Running Kafka on local system for signing in

To run the Kafka on your system to sign in, perform the below steps:

1. Install the Kafka on your system.
2. Change the path of ZooKeeper and server logs:
    a. Navigate to the folder where you have installed Kafka.
    b. Open the *config* folder and locate the *zookeeper.properties* file.
    c. In the *zookeeper.properties* file, enter the folder path of Kafka and add /zookeeper in the dataDir property. For example, set *dataDir=D:/MyKafka/zookeeper*. This configures ZooKeeper to store its logs in this folder.
    d. Open the *server.properties* file in the same **config** folder.
    e. In the *server.properties* file, locate the *log.dirs* property and set it to the desired path for the server logs. For example, set *log.dirs=D:/MyKafka/kafka-logs*. This creates the server logs in the *kafka-logs* folder.
3. After making these changes, follow the below steps to start Apache ZooKeeper:

a. Open the folder where Apache Kafka is installed and open a command prompt in that location.
b. To start the Apache ZooKeeper, run the below command:

```
D:\MyKafka>kafka_2.13-3.2.0\bin\windows\zookeeper-server-start.bat
kafka_2.13-3.2.0\config\zookeeper.properties
```

4. Start the Kafka server.
5. Open a separate command prompt and execute the below command:

```
D:\MyKafka>kafka_2.13-3.2.0\bin\windows\kafka-server-start.bat
kafka_2.13-3.2.0\config\server.properties
```

Ensure that you have provided the correct paths and adjust them according to your Kafka installation location.

Creating a new service with custom framework

Creating a new services is neccessary to encapsulate a specific set of functionality or business capability that can operate independently from other parts of your system.

To create a new service with custom framework, perform the below steps:

1. Make the below changes in the workspace:
a. Copy the *custom-service-workspace* project and paste it into your workspace location.
b. Rename the workspace folder with the service name.
c. Within the worskpace, access the *settings.gradle* file and modify the *rootProject.name* entry to reflect the selected service name.
d. Open the *application.properties* file in the workspace.
e. Within the *application.properties* file, locate the *com.service.name* tag and rename it with the new service name. Below are some examples of such services:
   - **userService** — Manages user accounts and authentication.
   - **productService** — Handles product information and catalog.
   - **inventoryService** — Tracks product availability.
2. Make the below changes in the WAR file:
a. Create a copy of *custom-service.war* file and rename it to the new service name. For example, *new-service.war* file.

   b. Replace the *settings.gradle* file and *application.properties* file from the workspace in the WAR file.
3. Use the WAR file and created workspace for new service development using custom microservice framework.

Enabling service to service calls

To enable the service-to-service calls using the Feign client in a custom service framework, perform the below steps:

1. Create separate services for communication to develop the individual services.
2. Declare the controller method to make a service-to-service call using the Feign client.

   To access the controller class, use the decompiler.

3. Retrieve the controller method as follows:
   a. Use the provided WAR file to locate the controller method.
   b. Extract the WAR file and find the controller class within it.
4. Fetch the appropriate method:
   a. After identifying the controller class, find the specific method that corresponds to the desired functionality you intend to call from your service.
   b. Match this method with the appropriate method type. For example: GET, POST, PUT, and DELETE for the desired interaction.
5. Use the method in your proxy files:
   a. After identifying the correct controller method, create proxy files or client interfaces in your custom service framework. The proxy files define the Feign client interface and the desired service interaction.
   b. Incorporate the fetched controller method into these proxy files to specify the service-to-service call.

Printing messages in console.logs

Printing messages in console logs allows you to perform debugging, error tracking, flow analysis, performance analysis, gain environmental insights, monitor and ensure

observability, conduct auditing, facilitate communication, perform security analysis, conduct regression testing, and more.

To print messages in console logs, perform the below steps:

1. To print the messages, utilize the *createConsoleLogRequest* method in the *Utility.java* file of the *custom-service-workspace*.
2. Provide the values for the following parameters:
   • Cabinet name
   • String message
3. Enter the desired message into the message string. This string must contain the message that you intend to display in the *console.log* file.
4. Autowire the utility class in the Java file and use the *createConsoleLogRequest* method as required.

Adding dependencies in custom-service framework

This section explains how to add new dependencies to the WAR files using the dependency-service.

To add new dependencies, perform the below steps:

1. Add the necessary dependency to your project in the workspace's *build.gradle* file.
2. Refresh the Gradle project to remove any errors.
3. Create a JAR file of the project and replace the code changes in the WAR file as provided.

   Ensure that *lib* folder within the WAR file is empty before copying the webinf folder of Apache Tomcat.

4. Add the dependency to the *build.gradle* file of the dependency-service.
5. Create an executable JAR file to run the service and then navigate to the location of the *dependency-service*.
6. Open the command prompt and execute the below command to build the WAR file for the service:

   ```
   gradlew clean build --no-build-cache --info
   ```

7. Under the dependency-service folder, navigate to **build> libs** and extract the JAR file named *dependency-service.war* using 7-Zip.

8. Inside the extracted JAR file, navigate to *\WEB-INF\lib* and then copy the new JAR files for the new dependency.
9. Paste the copied JAR files into the *apache-tomcat-9.0.65\lib* folder of Apache Tomcat.

Configuring CORS

CORS (Cross-Origin Resource Sharing) works as a security feature that prevents web pages from making requests that come from unknown sources. This is a fundamental security measure to protect against cross-site request forgery (CSRF) and other security vulnerabilities.

To allow a browser to access the microservice, specify the server's URL, that is the origin of the request in the *application.properties* file of the *custom-gateway.jar* file. This also allows the microservice to recognize the authorized requests.

To add an origin in the custom-gateway, enter the server URLs in the *allowed.origins* property as a comma-separated list with proper HTTP or HTTPS protocols.

For Example,

allowed.origins=http://localhost:8080,https://example.com,https://anotherdomain.com

Sample code

The following sample code covers scenarios to connect a cabinet using OmniDocs (OD) microservice, creating a new workitem with attributes, and then disconnecting the session using OD Service.

```
package com.newgen.sampleclientapplication;

import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;

import org.apache.commons.lang3.exception.ExceptionUtils;
import org.json.JSONException;
import org.json.JSONObject;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.http.HttpEntity;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpMethod;
import org.springframework.http.HttpStatus;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.web.client.RestTemplate;

@SpringBootApplication
public class SampleClientApplication {

    public static final int HTTP_STATUS_CODE = 0;
    public static final int RESPONSE_BODY = 1;
    public static final int MAIN_CODE = 2;
    public static final int SUB_ERROR_CODE = 3;
    public static final String REST_CALL_BROKER_EXCEPTION = "500";
    public static final int REST_CALL_BROKER_EXCEPTION_STACK_TRACE = 2;

    private static RestTemplate restTemplate;
    private static RestCallBrokerUtil restCallBrokerUtil;

    public static void main(String[] args) {
        Long start = System.currentTimeMillis();
        SampleClientApplication restClient = new SampleClientApplication();
        restClient.APIExecution();
        Long end = System.currentTimeMillis();
```

```
        System.out.println("Overall time taken: " + (end - start));
    }

    public void APIExecution() throws JSONException {
        String[] connectResponse = connect();
        JSONObject myResponse = new JSONObject(connectResponse[1]);
        String sessionId = myResponse.optString("userDBId");

        uploadWorkitem(sessionId, 827387, 773773, "Zolo espace");

        disconnectCabinet(sessionId);
    }

    private String[] uploadWorkitem(String sessionId, int phoneNumber,
int contact, String address)
            throws JSONException {
        Map<String, String> headers = new HashMap<>();
        Map<String, String> queryParams = new HashMap<>();
        Map<String, String> pathParams = new HashMap<>();
        String httpMethod = "POST";

        String url = "http://192.168.149.33:8765/workitem-service-context
/workitem/uploadworkitem";
        JSONObject uploadWorkItemBody = new JSONObject(
                "{ \"ProcessDefId\": 20, \"QueueId\": 61,
\"InitiateFromActivityId\": \"\", \"InitiateAlso\": \"N\"}");

        JSONObject attributes = new JSONObject();
        attributes.put("userName", "Harry Potter");
        attributes.put("phoneNumber", phoneNumber);
        attributes.put("Contact", contact);
        attributes.put("Address", address);
        attributes.put("Email", "xyz@newworld.com");
        attributes.put("Qualification", "Master's");

        uploadWorkItemBody.put("Attributes", attributes);

        headers.put("Authorization", sessionId);
        headers.put("Tenant-Id", "ibps5sp2p1cloud");
        String[] response = makeCall(url, httpMethod, headers, queryParams,
pathParams,
                uploadWorkItemBody.toString());
        System.out.println(response[1]);
        System.out.println("Workitem Created !!");
        return response;
    }

    public String[] connect() throws JSONException {
        Map<String, String> headers = new HashMap<>();
        Map<String, String> queryParams = new HashMap<>();
        Map<String, String> pathParams = new HashMap<>();
```

```
        String httpMethod = "POST";
        String url = "http://192.168.149.33:443/omnidocs/v1/connect";

        JSONObject connectInputBody = new JSONObject();
        connectInputBody.put("userPassword", "system123#");
        connectInputBody.put("userName", "rkbuser");
        connectInputBody.put("locale", "en-US");
        connectInputBody.put("userExist", "N");

        queryParams.put("cabinetName", "ibps5sp2p1cloud");

        String[] response = makeCall(url, httpMethod, headers, queryParams,
pathParams,
                connectInputBody.toString());
        System.out.println("Login Successful !!");
        return response;
    }

    public void disconnectCabinet(String sessionId) throws JSONException {
        Map<String, String> headers = new HashMap<>();
        Map<String, String> queryParams = new HashMap<>();
        Map<String, String> pathParams = new HashMap<>();
        String httpMethod = "POST";

        String url = "http://192.168.149.33:443/omnidocs/v1/disconnect";

        JSONObject disconnectCabinetBody = new JSONObject();
        disconnectCabinetBody.put("userDBId", sessionId);

        queryParams.put("cabinetName", "ibps5sp2p1cloud");

        String[] response = makeCall(url, httpMethod, headers, queryParams,
pathParams,
                disconnectCabinetBody.toString());
        System.out.println("Logged Out Successfully !!");
    }

    public static String[] makeCall(String uri, String method,
Map<String, String> headerParams,
            Map<String, String> queryParams, Map<String, String> pathParams,
String requestBody) {
        HttpHeaders headers = new HttpHeaders();

        restTemplate = WFRestTemplateConfig.getRestTemplate();

        restCallBrokerUtil = new RestCallBrokerUtil();
        ResponseEntity<String> responseEntity = new ResponseEntity<>(
HttpStatus.OK);
        String[] response = new String[4];

        JSONObject responseBody;
```

```
        try {
            HttpMethod httpMethod = HttpMethod.valueOf(method);
            restCallBrokerUtil.populateHttpHeaders(headerParams, headers);

            if (queryParams != null && queryParams.size() > 0) {
                uri = restCallBrokerUtil.populateQueryParams(uri, queryParams);
            }

            HttpEntity<String> httpEntity = new HttpEntity<>(requestBody,
headers);
            responseEntity = restTemplate.exchange(uri, httpMethod, httpEntity,
String.class, pathParams);
        } catch (Exception exception) {
            response[HTTP_STATUS_CODE] = REST_CALL_BROKER_EXCEPTION;
            response[RESPONSE_BODY] = exception.toString();
            response[REST_CALL_BROKER_EXCEPTION_STACK_TRACE] =
ExceptionUtils.getStackTrace(exception);
            return response;
        }

        response[HTTP_STATUS_CODE] = responseEntity.getStatusCode().toString();
        response[RESPONSE_BODY] = responseEntity.getBody();

        responseBody = new JSONObject(responseEntity.getBody());
        response[MAIN_CODE] = responseBody.optString("MainCode");
        response[SUB_ERROR_CODE] = responseBody.optString("SubErrorCode");

        return response;
    }
}

class WFRestTemplateConfig {

    private static RestTemplate restTemplate;

    public WFRestTemplateConfig() {
    }

    public static RestTemplate getRestTemplate() {
        if (restTemplate == null) {
            restTemplate = new RestTemplate();
        }
        return restTemplate;
    }

    public void setRestTemplate(RestTemplate restTemplate) {
        WFRestTemplateConfig.restTemplate = restTemplate;
    }
}

class RestCallBrokerUtil {
```

```
    public RestCallBrokerUtil() {
    }

    public void populateHttpHeaders(Map<String, String> headerParameters,
HttpHeaders headers) {
        Iterator<Entry<String, String>> iter = headerParameters.
entrySet().iterator();
        while (iter.hasNext()) {
            Map.Entry<String, String> header = iter.next();
            headers.add(header.getKey(), header.getValue());
        }

        if (headers != null) {
            headers.setContentType(
                    headers.getContentType() == null ? MediaType
.APPLICATION_JSON : headers.getContentType());
        } else {
            // Exception
        }
    }

    public String populateQueryParams(String uri, Map<String, String>
queryParameters) {
        StringBuilder uriWithQueryParams = new StringBuilder(uri);
        uriWithQueryParams.append("?");
        Iterator<Entry<String, String>> iter = queryParameters.entrySet()
.iterator();
        while (iter.hasNext()) {
            Map.Entry<String, String> queryParam = iter.next();
            uriWithQueryParams.append(queryParam.getKey()).append("=").
append(queryParam.getValue()).append("&");
        }
        uriWithQueryParams.deleteCharAt(uriWithQueryParams.length() - 1);
        uri = uriWithQueryParams.toString();
        return uri;
    }

    public String buildURI(String protocol, String ip, String port,
String endPoint) {
        StringBuilder uri = new StringBuilder();
        uri.append(protocol).append("://").append(ip).append(":").append(port)
.append(endPoint);
        return uri.toString();
    }
}
```