
ZAL**NI** DATA PLATFORM

REST API Guide

Release 5.0.2

Zaloni Inc.

Jan 10, 2019

Copyright Note

Copyright © 2018 Zaloni Inc. All rights reserved.

Zaloni believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS”. ZALONI INC. MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any Zaloni software described in this publication requires an applicable software license. For the most up-to-date listing of Zaloni product names, visit zaloni.com. All other trademarks used herein are the property of their respective owners.

Disclaimer

Zaloni Inc. reserves the right to change its products and services at any time to incorporate technological developments. This guide is subject to change without notice. This guide has been prepared with every precaution to ensure accuracy. However, Zaloni assumes no liability for any errors or omissions, nor for any damages resulting from the application or use of this information.

CONTENTS

1	REST API Guide	3
1.1	Verify the ZDP Version	3
1.2	Receiving session cookie	4
1.2.1	Log in to ZDP	4
1.2.2	Get User Information	5
1.3	Lineage API	13
1.3.1	Add/Update Lineage Process	13
1.3.2	Add/Update Lineage Event	15
1.3.3	Search File/Entity Lineage	18
1.3.4	Delete Lineage Event	24
1.3.5	Delete Lineage Process	25
1.4	Ingest API	25
1.4.1	Save or Update a Connection	26
1.4.1.1	Save/update a filesystem connection	26
1.4.1.2	Save/update a database connection	27
1.4.2	Fetch all Connection Details	30
1.4.3	Fetch Connection Details based on Filesystem/Database	35
1.4.4	Fetch Connection Details using Connection Instance Id	39
1.4.5	Delete an Existing Connection	41
1.4.6	Fetch All Landing Zone Servers	42
1.4.7	Add Source Directory to a Server	43
1.4.8	Fetch Source Directories of a Server	45
1.4.9	Get all Flume Ingestion Servers	47
1.4.10	Create Flume Ingestion Agent	48
1.4.11	Get Flume Ingestion Agent Details	54
1.4.12	Search Ingestion History	62
1.4.13	Add an Ingestion History Record	68
1.4.14	Delete an Ingestion History Record	70
1.4.15	Create or Update a File Pattern	71
1.4.15.1	Create a file pattern	74
1.4.15.2	Update a file pattern	74
1.4.16	Search for File Patterns	75
1.4.17	Fetch the Details of a File Pattern	76
1.4.18	Fetch Date Histogram for Ingestion History	77
1.5	Prepare API	80
1.5.1	Search and Fetch Workflows	80
1.5.2	Fetch Workflow Instance Details	84
1.5.3	Delete Workflow	88
1.5.4	Execute Workflow	90
1.5.5	Stop Workflow	91

1.5.6	Stop Workflows in Bulk	92
1.5.7	Restart Failed Workflows in Bulk	94
1.5.8	Fetch Workflow Execution History	95
1.5.9	Fetch Workflow Status	99
1.5.10	Restart Workflow Instance	103
1.5.11	Set Workflow Custom Status	108
1.5.12	Set Workflow Step Custom Status	109
1.5.13	Get Workflow Schedule	110
1.5.14	Save Workflow Schedule	111
1.5.15	Resume/Suspend Workflow Schedule	115
1.5.16	Resume/Suspend Workflow Schedules in Bulk	117
1.5.17	Unschedule Workflow	118
1.5.18	List all Workflow Schedules	119
1.5.19	Update Log Level for Workflow Schedules in Bulk	120
1.5.20	Update Cluster Id for Workflow Schedules in Bulk	122
1.5.21	Update the Execution Priority for Workflow Schedules in Bulk	124
1.5.22	Update the Run As User for Workflow Schedules in Bulk	125
1.5.23	Delete Workflow Schedules in Bulk	127
1.5.24	Fetch the Status of the Workflow Scheduler	128
1.5.25	Pause/Unpause Workflow Schedules Globally	129
1.5.26	Export Workflow	130
1.5.27	Import Workflow	132
1.5.28	Fetch List of Instance Logs	135
1.5.29	Download Workflow Log File	136
1.5.30	List Transformation	137
1.5.31	Create User-Defined Action	139
1.5.32	Fetch all the User-Defined Action Details	147
1.5.33	Fetch the Specific User-Defined Action Details	149
1.5.34	Update the User-Defined Action Details	151
1.5.35	Delete the User-Defined Action	155
1.5.36	Rename the Workflow Name	155
1.5.37	Update the Execution Priority of Workflows in Queue	156
1.5.38	Delete Workflow Instances in Queue	157
1.6	Metadata API	158
1.6.1	Search Entities	158
1.6.2	Create/Update an Entity	162
1.6.2.1	Create an entity	168
1.6.2.2	Update an entity	203
1.6.2.3	Create entity from a Hive table	231
1.6.3	Sync Entity with the Hive table	237
1.6.4	Fetch Schema Parser Attributes	243
1.6.5	Upload Data File for Schema Discovery	245
1.6.6	Fetch Details of an Entity	254
1.6.7	Delete an Entity	264
1.6.8	Validate Structure of an Entity	265
1.6.9	Filter and Fetch Entity Data	268
1.6.9.1	Fetch data of an entity	268
1.6.9.2	Fetch entity data using conditions	271
1.6.10	Profile Data	274
1.6.10.1	Fetch entity level profile metrics	274
1.6.10.2	Fetch field summary of profiled entity	275
1.6.10.3	Fetch profile metrics entity fields	277
1.6.10.4	Search and fetch profile metrics of entities	279
1.6.11	Manage Data Type Mappings	280

1.6.11.1	Fetch data quality datatypes	280
1.6.11.2	Fetch HCatalog datatypes	281
1.6.11.3	Fetch datatype mappings	282
1.6.11.4	Add custom datatype	286
1.6.12	Manage ZDP Entities	287
1.6.12.1	Create a ZDP managed entity	287
1.6.12.2	Create an entity instance	289
1.6.12.3	Edit entity instance	292
1.6.12.4	Fetch details of entity instance	295
1.6.12.5	Search entity instances	296
1.6.12.6	Export entity instances	298
1.6.12.7	Import entity instances	299
1.6.13	Update Sensitivity of an Entity Field	301
1.6.14	Bulk Import of Entities, Ingestion Processes, and Post Ingestion Workflows	302
1.7	Data Quality API	306
1.7.1	Add/Update Simple or Compound Rule (JSON Format)	306
1.7.2	Add Rule Set (JSON Format)	309
1.7.3	Update rule set (JSON Format)	310
1.7.4	View Rules (JSON Format)	312
1.7.5	View Rule by Entity (JSON Format)	313
1.7.6	Delete Rule (JSON Format)	314
1.7.7	Delete Rule Set (JSON Format)	315
1.7.8	Add Rules (CSV Format)	316
1.7.9	Add or Update rules (CSV Format)	316
1.7.10	Fetch Field/File Level DQ Aggregations	317
1.8	Bedrock Namespace API	320
1.8.1	Fetch Bedrock Namespace	320
1.8.2	Save Bedrock Namespace	321
1.8.3	Modify Bedrock Namespace	322
1.8.4	Delete Bedrock Namespace	323
1.9	Category Namespace API	324
1.9.1	Fetch Category Namespace	324
1.9.2	Save Category Namespace	325
1.9.3	Modify Category Namespace	326
1.9.4	Delete Category Namespace	327
1.10	Project Specific API	328
1.10.1	Fetch the List of Allocated Projects	328
1.10.2	Search Projects	331
1.10.3	Fetch Roles and Users Details	333
1.10.4	Create a ZDP Project	335
1.10.5	Fetch the Details of a Project	338
1.10.6	Update Role-User Mapping within a Project	339
1.10.7	Update a Project	341
1.10.8	Share Artifact with Project(s)	343
1.10.9	Unshare Artifact(s) from Project	346
1.10.10	Transfer Project Artifacts	347
1.10.10.1	Delete a project	350
1.10.11	Sync Project Policies in Ranger	350
1.11	Administration API	351
1.11.1	Get System Configuration	351
1.11.2	Global Search API	370
1.11.3	Services Monitor	375
1.11.3.1	Fetch services	375
1.11.3.2	Fetch instance details per service	377

1.11.3.3	Fetch details about ActiveMQ, Elasticsearch, and Logstash services	379
1.11.4	Capacity Scheduler	381
1.11.4.1	Configure capacity-scheduler	381
1.11.4.2	Fetch capacity-scheduler queues	382
1.11.5	Export-Import	384
1.11.5.1	Export ZDP resources	384
1.11.5.2	Download export file	389
1.11.5.3	Import ZDP resources	389
1.11.5.4	Download import log file	390
1.11.6	Data Lifecycle Management	391
1.11.6.1	Add DLM policy	391
1.11.6.2	Fetch DLM policies	392
1.11.6.3	Modify DLM policy	393
1.11.6.4	Delete DLM policy	395
1.11.6.5	Fetch DLM policy states	396
1.11.6.6	Fetch DLM policy age evaluators	397
1.11.6.7	Fetch list of policy definitions	398
1.11.6.8	Execute DLM process	400
1.11.6.9	Fetch list of DLM execution instances	401
1.11.6.10	Fetch list of DLM policy association	402
1.11.6.11	Add a DLM policy association	403
1.11.6.12	Fetch DLM policy association	404
1.11.6.13	Update DLM policy association	405
1.12	Notification API	407
1.13	Deprecated API	409
1.14	API Response Codes	411
1.15	References	412
1.15.1	Add or Update DQ Rules by using CSV file	412
1.15.2	Usage of Correct Parameters (Keys) in JSON	413
1.15.3	Bulk Importing of Metadata by using REST API	413
1.15.3.1	Driver file fields	415
1.15.3.2	Meta file fields	417
1.15.3.3	PII/Sensitivity file fields	418
1.15.4	Reindexing Artifacts	418

REST API Guide

The Zaloni Data Platform (ZDP) is a comprehensive, integrated solution that operationalizes data processes along the entire pipeline from data source to data consumer.

The REST API guide details the various REST APIs that you can use to make REST calls to the ZDP application. This guide includes the various APIs, methods, request payloads, success/error responses for various scenarios.

Important: The reference links used in this guide may not work. Refer the in-product help files to access the end-to-end documentation for ZDP.

REST API GUIDE

The REST API guide is designed for developers who want to leverage the ingestion, workflow, metadata, data quality, and administration modules and the notification feature of the ZDP platform.

This guide provides a description of the Application Programming Interfaces (APIs) and examples on how to use them, including detailed information on the APIs, such as:

- Executing workflows with specific parameters and setting custom status for step or workflow instance
- Obtaining the status of workflow instances
- Sending notification emails
- Exporting or importing workflows
- Managing files or stream-based ingestion
- Adding and updating ingestion history records
- Managing entities and importing metadata in bulk (entities, fields, and file patterns)
- Viewing data quality functions and operators and managing the data quality rules/rule sets
- Getting ACL status for Capacity Scheduler and viewing queues assigned to the user
- Managing Bedrock and Category namespaces

The ZDP platform provides easy-to-use REST API services. The format explained here is suitable for developers who use JSON over REST.

Important:

- If ZDP is deployed to run over HTTPS, ensure to use the same in the API URLs.
 - APIs with mandatory input parameters are marked in asterisk (*) to indicate wherever applicable.
 - ZDP currently supports only single Project Id in the REST API URLs (wherever applicable). The support for multiple project Ids is a **PREVIEW** feature.
-

1.1 Verify the ZDP Version

This API allows you to check the ZDP release version deployed in a specific server instance.

URL

```
http://<bedrock-host>:port/bedrock-app/services/rest/system/info
```

Method

GET

Example Response

```
{
  "responseMessage": "SUCCESS",
  "restUri": "/system/info",
  "result": {
    "bedrockVersion": "4.4.0"
  },
  "page": null
}
```

1.2 Receiving session cookie

A session cookie validates a user's privilege to call one of the ZDP services. When a service is used, the session cookie must be sent with the service request as an HTTP header.

- To receive a session cookie, a service client first must call the *Log in to ZDP* service.
- The *Log in to ZDP* service allows the client to authenticate itself to ZDP with the username and password. When the authentication is successful, this service returns a session cookie.
- The session cookie is returned as a cookie header identified by the name `JSESSIONID`.
- The session cookie must be passed in the subsequent service calls. If authentication fails, the *Log in to ZDP* service returns a HTTP response code of 401.
- A session cookie is temporary and the time period can be configured by the ZDP administrator. The default interval is 24 hours. The cookie expires if the client does not make a call within the specified interval.

The following screenshot is a browser-based REST client and displays how the *Log in to ZDP* service, with a valid username and password returns a session cookie.

The following screenshot displays how the `JSESSIONID` must be sent in the subsequent service calls. Here, a request is being made to the *Fetch Workflow Status* service. The `JSESSIONID` key is case sensitive; when passed (manually) during subsequent ReST calls.

If the service calls are made by using the API wrapper that ships with ZDP, the handling of the session cookie is done by the wrapper. The developer only needs to set the username and password.

Note: Against each service, there is a *Response in case of failure* list, which provides the responses that the service returns against a specific error. The services do not handle any errors apart from the ones mentioned in the list.

If a service is called without passing a valid session cookie in the header, an HTTP response code of *401* is returned.

1.2.1 Log in to ZDP

This API allows you to log in to a ZDP instance and must be called before any other service to receive a session cookie. This service accepts a username and password.

URL

```
http://<bedrock-host>:port/bedrock-app/services/rest/login
```

Method

```
POST
```

Request Payload

```
{
  "username": "<This is a valid ZDP username.>",
  "password": "<This is the user's password.>",
  "permissionNeeded": "<This indicates if permission is needed or not.>"
}
```

Example Request

```
{
  "username": "john",
  "password": "password"
}
```

Example Response

```
{
  "responseMessage": "User logged in successfully",
  "restUri": "/login",
  "result": null,
  "page": null
}
```

Response in case of a failure

If an invalid user Id or password is used and authentication fails:

```
{
  "result": "Authentication Failed due to wrong user id or password.",
  "status": {
    "responseCode": 401,
    "responseMessage": "Authentication Failed due to wrong user id or password.",
    "responseType": "info",
    "result": null
  }
}
```

1.2.2 Get User Information

This API allows you to fetch details of the user currently logged-in to a ZDP instance, such as roles assigned, granted permissions, etc. The session cookie received for the logged-in user must be passed as a URL parameter while using the ReST (or subsequent) call.

URL

```
http://<bedrock-host>:port/bedrock-app/services/rest/users/me
```

Method

GET

Example Response

```

{
  "responseMessage": "SUCCESS",
  "restUri": "/users/me",
  "result": {
    "id": 0,
    "chunkSize": 0,
    "status": 11,
    "userId": "admin21",
    "userStatus": "Enabled",
    "username": "admin21 admin123",
    "justification": null,
    "userCategory": "BEDROCK_INTERNAL",
    "numAssociatedProjects": null,
    "defaultProjectId": 1,
    "firstName": null,
    "lastName": null,
    "projectId": null,
    "roleList": [
      "Global Administrator"
    ],
    "secUserRolesVOList": [
      {
        "roleId": 1,
        "roleName": "Global Administrator",
        "roleDescription": "This is a system role",
        "roleType": "SYSTEM_DEFINED",
        "isGlobal": true,
        "newUserRoleWhenDeleted": 0,
        "secModules": [
          {
            "moduleId": 1,
            "moduleTechnicalName": "metadata",
            "modulePrettyName": "Metadata",
            "secModulePermissions": [
              {
                "modulePermissionId": 25,
                "permissionPrettyName": "Manage Entity Instances",
                "permissionTechnicalName": "md_manage_entity_instance",
                "description": "This permission grants the ability to_
↪manage entity instances",
                "permissionType": "USER_MANAGED",
                "isGlobal": true,
                "secDependentPermissions": [
                  {
                    "permissionMapId": 6,
                    "permissionTechnicalName": "md_manage_entity_
↪instance",
                    "requiredTechnicalName": "md_view_entity_instance
↪"
                  }
                ]
              },
              {
                "modulePermissionId": 24,

```

```

        "permissionPrettyName": "View Entity Instances",
        "permissionTechnicalName": "md_view_entity_instance",
        "description": "This permission grants the ability to_
↪view entity instances",
        "permissionType": "USER_MANAGED",
        "isGlobal": true,
        "secDependentPermissions": [
            {
                "permissionMapId": 7,
                "permissionTechnicalName": "md_view_entity_
↪instance",
                "requiredTechnicalName": "md_view_entity_type"
            }
        ]
    },
    {
        "modulePermissionId": 26,
        "permissionPrettyName": "Manage Managed List",
        "permissionTechnicalName": "md_manage_managed_list",
        "description": "This permission grants the ability to_
↪manage managed lists",
        "permissionType": "USER_MANAGED",
        "isGlobal": true,
        "secDependentPermissions": []
    },
    {
        "modulePermissionId": 27,
        "permissionPrettyName": "Manage Metadata Configuration",
        "permissionTechnicalName": "md_manage_metadata_config",
        "description": "This permission grants the ability to_
↪manage metadata configuration",
        "permissionType": "USER_MANAGED",
        "isGlobal": true,
        "secDependentPermissions": []
    }
],
{
    "moduleId": 2,
    "moduleTechnicalName": "ingestion",
    "modulePrettyName": "Ingestion",
    "secModulePermissions": [
        {
            "modulePermissionId": 8,
            "permissionPrettyName": "Manage Landing Zone Directories
↪",
            "permissionTechnicalName": "ingestion_manage_landing_
↪zone_directory",
            "description": "This permission grants the ability to_
↪manage landing zone directory and flume agents",
            "permissionType": "USER_MANAGED",
            "isGlobal": true,
            "secDependentPermissions": [
                {
                    "permissionMapId": 9,
                    "permissionTechnicalName": "ingestion_manage_
↪landing_zone_directory",
                    "requiredTechnicalName": "ingestion_view_landing_
↪zone_directory"
                }
            ]
        }
    ]
}

```

```

        },
        {
            "permissionMapId": 10,
            "permissionTechnicalName": "ingestion_manage_
↪landing_zone_directory",
            "requiredTechnicalName": "ingestion_view_file_
↪patterns"
        }
    ]
},
{
    "modulePermissionId": 19,
    "permissionPrettyName": "Manage Automated Data Inventory
↪",
    "permissionTechnicalName": "ingestion_manage_automated_
↪data_inventory",
    "description": "This permission allows user to create_
↪new Automated Data Inventory Profiles.",
    "permissionType": "USER_MANAGED",
    "isGlobal": true,
    "secDependentPermissions": [
        {
            "permissionMapId": 20,
            "permissionTechnicalName": "ingestion_manage_
↪automated_data_inventory",
            "requiredTechnicalName": "ingestion_view_
↪automated_data_inventory"
        }
    ]
},
{
    "modulePermissionId": 17,
    "permissionPrettyName": "Ingest Data",
    "permissionTechnicalName": "ingestion_ingest_data",
    "description": "This is ZDP system permission used to_
↪ingest data to HDFS",
    "permissionType": "USER_MANAGED",
    "isGlobal": true,
    "secDependentPermissions": []
},
{
    "modulePermissionId": 10,
    "permissionPrettyName": "Manage Landing Zone Servers",
    "permissionTechnicalName": "ingestion_manage_landing_
↪zone_server",
    "description": "This permission grants the ability to_
↪manage file based and stream ingest servers",
    "permissionType": "USER_MANAGED",
    "isGlobal": true,
    "secDependentPermissions": [
        {
            "permissionMapId": 11,
            "permissionTechnicalName": "ingestion_manage_
↪landing_zone_server",
            "requiredTechnicalName": "ingestion_view_landing_
↪zone_servers"
        }
    ]
}
]

```



```

    },
    {
      "modulePermissionId": 14,
      "permissionPrettyName": "View Landing Zone Servers",
      "permissionTechnicalName": "ingestion_view_landing_zone_
↪servers",
      "description": "This permission grants the ability to_
↪view file based and stream ingest servers",
      "permissionType": "USER_MANAGED",
      "isGlobal": true,
      "secDependentPermissions": []
    },
    {
      "modulePermissionId": 13,
      "permissionPrettyName": "Manage Ingestion History",
      "permissionTechnicalName": "ingestion_manage_ingestion_
↪history",
      "description": "This permission grants the ability to_
↪insert, update and delete ingestion records into solr",
      "permissionType": "USER_MANAGED",
      "isGlobal": true,
      "secDependentPermissions": []
    },
    {
      "modulePermissionId": 9,
      "permissionPrettyName": "Get Files To Ingest",
      "permissionTechnicalName": "ingestion_get_file_to_ingest
↪",
      "description": "This is ZDP system permission used to_
↪fetch the list of files from BDCA slave",
      "permissionType": "SYSTEM_MANAGED",
      "isGlobal": true,
      "secDependentPermissions": []
    },
    {
      "modulePermissionId": 21,
      "permissionPrettyName": "Manage Connection",
      "permissionTechnicalName": "ingestion_manage_connection",
      "description": "This permission grants the ability to_
↪manage connection",
      "permissionType": "USER_MANAGED",
      "isGlobal": true,
      "secDependentPermissions": []
    },
    {
      "modulePermissionId": 20,
      "permissionPrettyName": "View Automated Data Inventory",
      "permissionTechnicalName": "ingestion_view_automated_
↪data_inventory",
      "description": "This permission allows user to view_
↪Automated Data Inventory Profiles.",
      "permissionType": "USER_MANAGED",
      "isGlobal": true,
      "secDependentPermissions": []
    },
    {
      "modulePermissionId": 18,
      "permissionPrettyName": "Evict Cache",

```

```

        "permissionTechnicalName": "ingestion_evict_cache",
        "description": "This is ZDP system permission used to_
↪evict cache",
        "permissionType": "SYSTEM_MANAGED",
        "isGlobal": true,
        "secDependentPermissions": []
    },
    {
        "modulePermissionId": 11,
        "permissionPrettyName": "View Landing Zone Directories",
        "permissionTechnicalName": "ingestion_view_landing_zone_
↪directory",
        "description": "This permission grants the ability to_
↪view landing zone directory and flume agents",
        "permissionType": "USER_MANAGED",
        "isGlobal": true,
        "secDependentPermissions": [
            {
                "permissionMapId": 13,
                "permissionTechnicalName": "ingestion_view_
↪landing_zone_directory",
                "requiredTechnicalName": "ingestion_view_landing_
↪zone_servers"
            }
        ]
    },
    {
        "modulePermissionId": 16,
        "permissionPrettyName": "Update BDCA",
        "permissionTechnicalName": "ingestion_update_bdca",
        "description": "This is ZDP system permission used to_
↪update BDCA",
        "permissionType": "SYSTEM_MANAGED",
        "isGlobal": true,
        "secDependentPermissions": []
    }
],
{
    "moduleId": 3,
    "moduleTechnicalName": "workflow",
    "modulePrettyName": "Workflow",
    "secModulePermissions": [
        {
            "modulePermissionId": 7,
            "permissionPrettyName": "Notify Workflow Finished",
            "permissionTechnicalName": "wf_notify_workflow_finished",
            "description": "This is ZDP system permission used to_
↪notify workflow finished",
            "permissionType": "SYSTEM_MANAGED",
            "isGlobal": true,
            "secDependentPermissions": []
        },
        {
            "modulePermissionId": 4,
            "permissionPrettyName": "View Preview Features",
            "permissionTechnicalName": "wf_view_preview_features",
            "description": "This permission grants the ability to_
↪view preview features in workflows",

```

```

        "permissionType": "USER_MANAGED",
        "isGlobal": true,
        "secDependentPermissions": []
    }
  ],
  },
  {
    "moduleId": 5,
    "moduleTechnicalName": "admin",
    "modulePrettyName": "Administration",
    "secModulePermissions": [
      {
        "modulePermissionId": 44,
        "permissionPrettyName": "View Projects",
        "permissionTechnicalName": "admin_view_project",
        "description": "This permission grants the ability to
↪view projects and resource groups",
        "permissionType": "USER_MANAGED",
        "isGlobal": true,
        "secDependentPermissions": []
      },
      {
        "modulePermissionId": 38,
        "permissionPrettyName": "Manage Notifications",
        "permissionTechnicalName": "admin_manage_notification",
        "description": "This permission grants the ability to
↪manage notification templates",
        "permissionType": "USER_MANAGED",
        "isGlobal": true,
        "secDependentPermissions": [
          {
            "permissionMapId": 18,
            "permissionTechnicalName": "admin_manage_
↪notification",
            "requiredTechnicalName": "admin_view_notification
↪"
          }
        ]
      },
      {
        "modulePermissionId": 45,
        "permissionPrettyName": "View System Configuration",
        "permissionTechnicalName": "admin_view_system_
↪configuration",
        "description": "This permission grants the ability to
↪view system configurations, view workflow executors and search users",
        "permissionType": "USER_MANAGED",
        "isGlobal": true,
        "secDependentPermissions": []
      },
      {
        "modulePermissionId": 54,
        "permissionPrettyName": "Manage DLM",
        "permissionTechnicalName": "admin_manage_dlm",
        "description": "This is dlm permission
↪regarding\n\t\t\ttdlm policies operation",
        "permissionType": "USER_MANAGED",
        "isGlobal": true,

```

```

        "secDependentPermissions": []
    }
}
},
{
    "moduleId": 6,
    "moduleTechnicalName": "data_quality",
    "modulePrettyName": "Data Quality",
    "secModulePermissions": [
        {
            "modulePermissionId": 32,
            "permissionPrettyName": "View Rules/Rule Sets and_
↪Functions",
            "permissionTechnicalName": "dq_view_rule",
            "description": "This permission grants the ability to_
↪view rules, rulesets and functions",
            "permissionType": "USER_MANAGED",
            "isGlobal": true,
            "secDependentPermissions": []
        },
        {
            "modulePermissionId": 33,
            "permissionPrettyName": "Manage Rules/Rule Sets and_
↪Functions",
            "permissionTechnicalName": "dq_manage_rule",
            "description": "This permission grants the ability to_
↪manage rules, rulesets and functions(custom only)",
            "permissionType": "USER_MANAGED",
            "isGlobal": true,
            "secDependentPermissions": [
                {
                    "permissionMapId": 14,
                    "permissionTechnicalName": "dq_manage_rule",
                    "requiredTechnicalName": "dq_view_rule"
                }
            ]
        }
    ]
},
{
    "moduleId": 8,
    "moduleTechnicalName": "common_authorization",
    "modulePrettyName": "Common Authorization",
    "secModulePermissions": [
        {
            "modulePermissionId": 51,
            "permissionPrettyName": "Common Permission",
            "permissionTechnicalName": "common_permission",
            "description": "This permission is used to allocate_
↪permissions for all common resources.",
            "permissionType": "USER_MANAGED",
            "isGlobal": true,
            "secDependentPermissions": []
        }
    ]
}
},
"createdDate": null,

```

```

        "modifiedDate": null,
        "createdBy": null,
        "modifiedBy": null,
        "userExists": true
    },
    "msg": null
},
"page": null
}

```

1.3 Lineage API

You can view the lineage for files and entities in the UI. The following section lists the APIs that can be invoked to add, modify, or delete lineage process and lineage events, external to ZDP.

These APIs can be used to set linkage with file/entity lineage processes, existing in a ZDP server instance, to meet the business requirements.

To know about the lineage APIs in detail, contact - support.zaloni.com

1.3.1 Add/Update Lineage Process

This API allows you to add or update a lineage process. A process is a workflow or task that you might want to perform on the data/files.

URL

```

http://<bedrock-host>:port/bedrock-app/services/rest/lineage/sources/{source_name}/
↳processes/{process_Id}

```

Note:

- *source_name* is the actual source of the lineage inflow which can be external or internal. Internal source is identified as *ZDP*.
- *process_id* is any unique identification number for a process. For internal source (*ZDP*), the value for *process_id* is *workflowid.instanceid*.
- You must pass the values for the *source_name* and *process_Id* parameters in the API URL.

Method

```
PUT
```

Request Payload

```

{
  *"name": "<This is the name of the process that can help you to understand the_
  ↳process easily.>",
  *"startTime": "<This is the start time of the process. The value for this parameter_
  ↳cannot be greater than that of endTime. Must be in the format- <yyyy-MM-dd>T
  ↳<HH:mm:ss>Z.>",
  *"endTime": "<This is the end time of the process. Must be in the format- <yyyy-MM-dd>
  ↳T<HH:mm:ss>Z.>",
}

```

```
*"type": "<This is the type of the process. This value can be workflow, query, dlm,
↳etc.>",
  "modifiedBy": "<This is the user who is modifying the process. If this parameter is
↳not applicable, set it to blank.>",
*"finalStatus": "<This is the status of the process. This value can be SUCCESS.>",
  "modifiedTime": "<This is the time when the process was modified. This parameter is
↳optional if the source is BEDROCK. Must be in the format- <yyyy-MM-dd>T<HH:mm:ss>Z.
↳For example, 2015-12-25T18:30:00Z>",
*"createdBy": "<This is the user who is creating the process.>",
  "creationTime": "<This is the time when the process was created. This parameter is
↳optional while creating a process, if the source is BEDROCK. The value for this
↳parameter cannot be greater than that of modifiedTime, endTime, startTime. Must be
↳in the format- <yyyy-MM-dd>T<HH:mm:ss>Z>",
*"initiatedBy": "<This is the user who initiated the lineage process.>",
  "version": "<This is the version of the lineage process.>",
  "processProperties": {"<This indicates the properties for the process as key-value
↳pairs.>"}
```

Example Request

To create a lineage process:

```
{
  "name": "Process_SB",
  "startTime": "2016-03-05T18:30:00Z",
  "endTime": "2016-03-07T18:38:00Z",
  "type": "Workflow",
  "modifiedBy": "admin",
  "finalStatus": "FAIL",
  "modifiedTime": "2016-03-06T18:30:00Z",
  "createdBy": "admin",
  "creationTime": "2016-03-05T18:30:00Z",
  "initiatedBy": "admin_bedrock",
  "version": "1",
  "processProperties": {
    "SB": "Test"
  }
}
```

Example Response

- Scenario 1:

If a lineage process is added successfully:

```
{
  "responseMessage": "Process added successfully [id=1234,
↳name=WallMartProcess1, type=Workflow].",
  "restUri": null,
  "result": null
}
```

- Scenario 2:

If an existing lineage process is updated:

```
{
  "responseMessage": "Process updated successfully [id=226, name=Process_SB,
↳type=FILE].",
```

```

    "restUri": null,
    "result": null
  }

```

Response in Case of Failure

If the request payload has error:

```

{
  "responseMessage": "Wrong date format provided for Process end time.Please provide_
↳the date in yyyy-MM-dd'T'HH:mm:ss'Z' format only.",
  "restUri": null,
  "result": "/bedrock-app/services/rest/lineage/sources/my/processes/2"
}

```

1.3.2 Add/Update Lineage Event

This API allows you to add or update a lineage event under an existing lineage process. Every process contains a set of events/steps. These events/steps are the actual operations that are performed on the data/file. Some of the events/steps in ZDP are **AVRO Action**, **File Move Action**, **VSAM Parser Action**, **Watermark Action**, **Token Masking Action**, etc.

URL

```

http://<bedrock-host>:port/bedrock-app/services/rest/lineage/sources/{source_name}/
↳processes/{process_Id}/events/{event_Id}

```

Note:

- *event_Id* is the unique identifier for a lineage event for an entity. This property is open for extension. The recommended convention is:
 - *processid.stepid.processInstanceId* for workflow items
 - *UUID* for ingestion events
 - Any other unique identifier by internal and external sources.
- *source_name* and *process_Id* must be associated with an existing process.
- You must pass the values for the *source_name*, *process_Id*, and *event_Id* parameters in the API URL.

Method

PUT

Request Payload

```

{
  "parentProcessId": "<This is the Id of an existing process.>",
  "type": "<This is the type of event for which lineage needs to be captured. For_
↳example, Watermark Action/Shell/Database/pig-script. This is analogous to stepType_
↳in Bedrock.>",
  "displayName": "<This is the user-friendly display name of the event. In terms of_
↳DAG (Direct Acyclic Graph), this is the edge between two input and output nodes._
↳This is analogous to stepName in Bedrock.>",
  "status": "<This is status of the lineage event. It can be Success or Fail.>",
}

```

```

"userId": "<This is the user identifier of the user executing the event. This can be
↳ of String data type.>",
"eventStartDate": "<This is the start date of the event. The value for this
↳ parameter cannot be greater than that of eventEndDate and parentProcessCreationTime.
↳ Must follow the format- yyyy-MM-ddThh:mm:ssZ.>",
"eventEndDate": "<This is the end date of the event. Must follow the format- yyyy-MM-
↳ ddThh:mm:ssZ.>",
"input": [
  {
    "entity": {
      "technicalName": "<This is the technical name of the input entity.>",
      "version": "<This is the entity version number.>",
      "namespace": "<This is the namespace to logically group entities
↳ through a naming system. Nomenclature is '.' separated. Bedrock entities can have
↳ source schema and source platform in this property.>"
      "entityId": "<This is the entity identifier. It can be Bedrock
↳ specific entity Id or any external entity identifier.>"
    },
    "datasetProperties": { "<This is the list of key-value properties which
↳ might be specific to a particular lineage record. For example, workflow actions
↳ might need some parameters to be persisted for display in the lineage graph.>" },
    "datasetType": " <This is the type of the dataset. A few values are: FILE,
↳ RECORD, RDD.>",
    "displayName": " <This is the user-friendly name to display a context about
↳ the input node of the lineage DAG (Direct Acyclic Graph).>",
    "dataSet": "<This has the details of the dataset being taken as input. For
↳ example, /tmp/DQ-report-546-435.csv.>"
    "datasetStage": "<This is the dataset stage to refer to the stage where the
↳ dataset currently reside. This can be HDFS (input dataset in datalake) , SOURCE
↳ (input dataset in place from where it is being imported to datalake).>"
  }
],
"output": [
  {
    "displayName": "<This is the user-friendly name to display a context about
↳ the output node of the lineage graph.>",
    "dataSet": "<This has the details of the dataset being taken as output. For
↳ example, /tmp/DQ-report-546-435.csv.>",
    "datasetStage": "<This is the dataset stage to refer to the stage where
↳ the dataset currently resides. This can be HDFS (output dataset in datalake) ,
↳ PROVISION (output dataset in place from where it is being imported to datalake).>"
    "datasetProperties": { "<This is the list of key-value properties which
↳ might be specific to a particular lineage record. For example, workflow actions
↳ might need some parameters to be persisted for display in the lineage graph.>" },
    "datasetType": "<This is the type of the dataset. A few values are FILE,
↳ RECORD, RDD.>",
    "entity": {
      "technicalName": "<This is the technical name of the output entity.>",
      "version": "<This is the entity version number.>",
      "namespace": "<This is the namespace to logically group entities
↳ through a naming system. Nomenclature is '.' separated. Bedrock entities can have
↳ source schema and source platform in this property.>"
      "entityId": "<This is the entity identifier. It can be Bedrock
↳ specific entity Id or any external entity identifier.>"
    }
  }
],
"otherParams": { "<This is the placeholder for list of miscellaneous properties.>"
↳ }

```



```
}
```

Example Request

```
{
  "input": [
    {
      "entity": {
        "technicalName": "ONE_UPDATED",
        "version": "1",
        "namespace": "b_srcschema.b_srcpltfm"
      },
      "datasetProperties": {},
      "datasetType": "File",
      "displayName": "Watermark Action",
      "dataSet": "/user/bedrock/wm_input"
    }
  ],
  "otherParams": {
    "Execution Engine": "Map Reduce"
  },
  "status": "Step_Status",
  "userId": "sachin",
  "eventStartDate": "2016-12-11T18:30:00Z",
  "eventEndDate": "2016-12-11T18:30:00Z",
  "output": [
    {
      "datasetProperties": {},
      "datasetType": "RECORD",
      "entity": {
        "technicalName": "TWO_UPDATED",
        "version": "1",
        "namespace": "ao_ss.ao_sp"
      },
      "displayName": "Watermark Action",
      "dataSet": "/user/bedrock/wm_output"
    }
  ],
  "type": "Watermark Action",
  "displayName": "Two.1"
}
```

Example Response

- Scenario 1:

If a lineage event is added:

```
{
  "responseMessage": "Event added successfully [source=source,process id=1234,↵
  ↵event id=1, event name=Two.1, type=Watermark Action].",
  "restUri": null,
  "result": null
}
```

- Scenario 2:

If a lineage event is updated:

```
{
  "responseMessage": "Event updated successfully [source=source,process id=1234,
  ↪event id=1, event name=Two.1, type=Watermark Action].",
  "restUri": null,
  "result": null
}
```

Response in Case of Failure

If the request payload has error:

```
{
  "responseMessage": "Event start date  can not be blank.",
  "restUri": null,
  "result": "/bedrock-app/services/rest/lineage/sources/my/processes/2/events/21"
}
```

1.3.3 Search File/Entity Lineage

This API allows you to search and fetch file and entity lineages.

URL

```
http://<bedrock-host>:port/bedrock-app/services/rest/lineage/search
```

Method

```
POST
```

Request Payload

```
{
  *"type": "<This is the lineage search type. Set it to FILE to perform a search_
  ↪for file lineage. Set it to ENTITY to perform an entity search.>",
  "depth": "<This is the depth (level) of lineage records (from the most recent),
  ↪ to be displayed. Set it to -1 to view all the lineage records.>",
  "queryData": "<This is the search query defined as key: value pairs. The query_
  ↪parameters are listed below. Refer Example Request 1 and Example Request 2 for more_
  ↪details.>"
  {
    *"dataSet": "<This is the location of files for which lineage will be_
    ↪queried. Applicable for FILE type search only.>",
    *"technicalName": "<This is the technical name of the entity. Applicable_
    ↪for ENTITY type search only.>",
    *"version": "<This is the entity version for which lineage is queried._
    ↪Applicable for ENTITY type search only.>",
    *"namespace": "<This follows the format: {Entity_Source_Schema}.{Entity_
    ↪Source_Platform} of the entity type being queried. Applicable for ENTITY type_
    ↪search only.>"
  }
}
```

Example Request

- Scenario 1:
For *FILE* lineage:

```
{
  "type": "FILE",
  "depth": 2,
  "queryData": {
    "dataSet": "/user/bedrock/files"
  }
}
```

- Scenario 2:

For *ENTITY* lineage:

```
{
  "type": "ENTITY",
  "depth": 2,
  "queryData": {
    "technicalName": "Two",
    "version": "1",
    "namespace": "src_schema.src_platform"
  }
}
```

Example Response

- Scenario 1:

If the lineage of a *FILE* is fetched successfully:

```
{
  "responseMessage": null,
  "restUri": null,
  "result": {
    "processRecords": [
      {
        "id": "2",
        "name": "DLM Lineage Records",
        "startTime": "2016-07-15T16:02:57Z",
        "endTime": "2016-07-15T16:02:57Z",
        "procespassTime": 1468598577058,
        "type": "External",
        "modifiedBy": null,
        "finalStatus": "SUCCESS",
        "modifiedTime": "2016-07-15T16:02:57Z",
        "createdBy": null,
        "creationTime": "2016-07-15T16:02:57Z",
        "initiatedBy": "admin",
        "version": "1",
        "source": "BEDROCK",
        "processProperties": null
      }
    ],
    "events": []
  },
  "page": null
}
```

- Scenario 2:

If the lineage of an *ENTITY* is fetched successfully:

```
{
  "responseMessage": null,
  "result": {
    "processRecords": [
      {
        "id": "102",
        "name": "DIBYA_CDC_BASIC",
        "startTime": "2016-12-11T18:30:00Z",
        "endTime": "2016-12-11T18:30:00Z",
        "type": "Workflow",
        "modifiedBy": "THIRD_Creation",
        "finalStatus": "THIRD_UPDATE",
        "modifiedTime": "2016-12-11T18:30:00Z",
        "createdBy": "THIRD",
        "creationTime": "2016-12-11T18:30:00Z",
        "initiatedBy": "sachin_admin",
        "version": "THIRD_UPDATE",
        "source": "TeraData",
        "processProperties": {
          "ABC": "HELLO_TERADATA_ONE"
        }
      }
    ],
    "events": [
      {
        "id": "1011",
        "parentProcessId": "102",
        "type": "Watermark Action",
        "displayName": "Two.1",
        "status": "Step_Status",
        "eventStartDate": "1456310147000",
        "eventEndDate": "1456310172000",
        "userId": "sachin",
        "input": [
          {
            "entity": {
              "namespace": "a.a",
              "technicalName": "Two",
              "version": "1"
            },
            "displayName": "Watermark Action",
            "dataSet": "2.in2",
            "datasetType": "File",
            "datasetProperties": {
              "NA": "NA"
            }
          }
        ],
        "output": [
          {
            "entity": {
              "namespace": "a.a",
              "technicalName": "Two",
              "version": "1"
            },
            "displayName": "Watermark Action",
            "dataSet": "1.in2",
            "datasetType": "RECORD",
```

```

        "datasetProperties": {
            "NA": "NA"
        }
    },
    ],
    "otherParams": {
        "this": null
    }
},
{
    "id": "1012",
    "parentProcessId": "102",
    "type": "Watermark Action",
    "displayName": "Two.1",
    "status": "Step_Status",
    "eventStartDate": "1456310147000",
    "eventEndDate": "1456310172000",
    "userId": "sachin",
    "input": [
        {
            "entity": {
                "namespace": "a.a",
                "technicalName": "Two",
                "version": "1"
            },
            "displayName": "Watermark Action",
            "dataSet": "2.in1",
            "datasetType": "File",
            "datasetProperties": {
                "NA": "NA"
            }
        }
    ],
    "output": [
        {
            "entity": {
                "namespace": "a.a",
                "technicalName": "Two",
                "version": "1"
            },
            "displayName": "Watermark Action",
            "dataSet": "1.in",
            "datasetType": "RECORD",
            "datasetProperties": {
                "NA": "NA"
            }
        }
    ],
    "otherParams": {
        "this": null
    }
},
{
    "id": "1011",
    "parentProcessId": "102",
    "type": "Watermark Action",
    "displayName": "Two.1",
    "status": "Step_Status",

```

```

    "eventStartDate": "1456310147000",
    "eventEndDate": "1456310172000",
    "userId": "sachin",
    "input": [
      {
        "entity": {
          "namespace": "a.a",
          "technicalName": "Two",
          "version": "1"
        },
        "displayName": "Watermark Action",
        "dataSet": "2.in2",
        "datasetType": "File",
        "datasetProperties": {
          "NA": "NA"
        }
      }
    ],
    "output": [
      {
        "entity": {
          "namespace": "a.a",
          "technicalName": "Two",
          "version": "1"
        },
        "displayName": "Watermark Action",
        "dataSet": "1.in2",
        "datasetType": "RECORD",
        "datasetProperties": {
          "NA": "NA"
        }
      }
    ],
    "otherParams": {
      "this": null
    }
  },
  {
    "id": "1011",
    "parentProcessId": "102",
    "type": "Watermark Action",
    "displayName": "Two.1",
    "status": "Step_Status",
    "eventStartDate": "1456310147000",
    "eventEndDate": "1456310172000",
    "userId": "sachin",
    "input": [
      {
        "entity": {
          "namespace": "a.a",
          "technicalName": "Two",
          "version": "1"
        },
        "displayName": "Watermark Action",
        "dataSet": "2.in2",
        "datasetType": "File",
        "datasetProperties": {
          "NA": "NA"
        }
      }
    ]
  }
]

```

```

    }
  },
  "output": [
    {
      "entity": {
        "namespace": "a.a",
        "technicalName": "Two",
        "version": "1"
      },
      "displayName": "Watermark Action",
      "dataSet": "1.in2",
      "datasetType": "RECORD",
      "datasetProperties": {
        "NA": "NA"
      }
    }
  ],
  "otherParams": {
    "this": null
  }
},
{
  "id": "1012",
  "parentProcessId": "102",
  "type": "Watermark Action",
  "displayName": "Two.1",
  "status": "Step_Status",
  "eventStartDate": "1456310147000",
  "eventEndDate": "1456310172000",
  "userId": "sachin",
  "input": [
    {
      "entity": {
        "namespace": "a.a",
        "technicalName": "Two",
        "version": "1"
      },
      "displayName": "Watermark Action",
      "dataSet": "2.in1",
      "datasetType": "File",
      "datasetProperties": {
        "NA": "NA"
      }
    }
  ],
  "output": [
    {
      "entity": {
        "namespace": "a.a",
        "technicalName": "Two",
        "version": "1"
      },
      "displayName": "Watermark Action",
      "dataSet": "1.in",
      "datasetType": "RECORD",
      "datasetProperties": {
        "NA": "NA"
      }
    }
  ]
}

```

```
    }
  },
  "otherParams": {
    "this": null
  }
}
]
```

Response in Case of Failure

If no lineage meets the search criteria:

```
{
  "responseMessage": null,
  "restUri": null,
  "result": {
    "processRecords": null,
    "events": null
  }
}
```

1.3.4 Delete Lineage Event

This API allows you to delete an existing lineage event.

URL

```
http://<bedrock-host>:port/bedrock-app/services/rest/lineage/sources/{source_name}/
↳processes/{process_Id}/events/{event_Id}
```

Note: You must pass the values for the *source_name*, *process_Id*, and *event_Id* parameters in the API URL.

Method

```
DELETE
```

Example Request

If a lineage event is deleted:

```
{
  "responseMessage": "Event deleted successfully [source=source,process id=1234,
↳event id=1]",
  "restUri": null,
  "result": null
}
```

Response in Case of Failure

If no lineage event is found:


```
{
  "responseMessage": "No event records found for the process record [event id=1,
  ↳ process id=1234,source=source]",
  "restUri": null,
  "result": null
}
```

1.3.5 Delete Lineage Process

This API allows you to delete a lineage process. Deleting a process will delete the events associated with that process.

URL

```
http://<bedrock-host>:port/bedrock-app/services/rest/lineage/sources/{source_name}/
  ↳ processes/{process_Id}
```

Note: You must pass the values for the *source_name* and *process_Id* parameters in the API URL.

Method

DELETE

Example Response

If the process record has been deleted:

```
{
  "responseMessage": "Process deleted successfully [id=226,source= sb_sc]",
  "restUri": null,
  "result": null
}
```

Response in Case of Failure

If the *source_name* and *process_Id* combination does not exist:

```
{
  "responseMessage": "No lineage records found for process record [id=227,source=sb_sc]
  ↳ ",
  "restUri": null,
  "result": null
}
```

1.4 Ingest API

The ingest API services allow you to perform various tasks in ZDP, such as:

- Save, update, or fetch details of the connections in ZDP.
- Register or fetch details of a landing zone (or Flume) server.
- Create or fetch details of source directories (or Flume agents) in ZDP.
- Query or update the records in the ingestion history.
- Create, update, delete, or fetch details of a file pattern.

1.4.1 Save or Update a Connection

The following APIs can be invoked to add or update a filesystem (or database) connection in ZDP.

1.4.1.1 Save/update a filesystem connection

This API allows you to add or update a filesystem connection in a ZDP server instance.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/filesystems/{file_
↪system_Id}/connections
```

Note: You must pass the *file_system_Id* parameter in the API URL. This can be 1 (for *LOCAL*), 2 (for *HDFS*), 3 (for *Amazon S3*), or 4 (for *Azure BLOB*).

Method:

```
POST
```

Request Payload:

```
{
  *"connectionInstanceId": "<This is the connection instance Id. Set it to 0_
↪when you define a new connection or set the corresponding connectionInstanceId for_
↪the respective connection that needs an update.>"
  *"fileSystemUri": "<This is the file system URI. The following are the_
↪default scheme names:
    For Local                file:///
    For S3                   s3n://
    For HDFS                 hdfs://
    For Azure                wasbs://>",
  *"connectionInstanceName": "<This is the name of the connection.>",
  "fileSystemProperties": [
    {
      "connectionPropertyId": "<To save a new connection, set it to 0. To_
↪update an existing connection, set it to the respective connectionPropertyId.>",
      "fileSystemPropertyKey": "<This is the file system property KEY (if_
↪any).>",
      "fileSystemPropertyValue": " <This is the file system property value_
↪for the corresponding key(s).>"
    }
  ],
  "scope": "<This is the connection type. For example, PUBLIC or PRIVATE. A_
↪private connection can be updated only by the user who created the connection. If_
↪you do not provide any value for this parameter, the system defaults the value to_
↪PUBLIC.>",
  "description": "<This is the description of the connection.>"
}
```

Example Request:

To define a new connection:

```
{
  "connectionInstanceId":0 ,
```

```
{
  "fileSystemUri": "hdfs://",
  "connectionInstanceName": "test_connection",
  "fileSystemProperties": [
    {
      "connectionPropertyId": 0,
      "fileSystemPropertyKey": "a",
      "fileSystemPropertyValue": "dd"
    }
  ],
  "scope": "PUBLIC",
  "description": "This is a test connection."
}
```

Example Response:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Connection instance 16 saved successfully ",
    "result": 16
  },
  "result": "Connection instance 16 saved successfully "
}
```

Response in Case of Failure:

If a connection with the same name or URI exists:

```
{
  "status": {
    "responseType": "ERROR",
    "responseCode": 500,
    "responseMessage": "File system connection instance with same name or
↪URI already exists",
    "result": null
  },
  "result": null
}
```

1.4.1.2 Save/update a database connection

This API allows you to add or update a database connection in a ZDP server instance.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/database/{database_Id}/
↪connections
```

Note: You must pass the *database_Id* parameter in the API URL. The value must be set as 5.

Method:

POST

Request Payload:

```
{
  *"connectionInstanceId": "<To add a database connection, set this value as 0.
  ↳To update an existing database connection, set this value as the corresponding
  ↳connection Id.>" ,
  *"connectionInstanceName": "<This is the name of the connection instance.>",
    "description": "<This is a brief description (if any) for the connection.>",
  *"connectionType": "<This is the type of connection. Set the value as RDBMS.>
  ↳",
  *"databaseUri": "<This is the URL of the relational database that needs to be
  ↳connected.>",
    "databaseProperties": ["<This is an array of database properties (if any) in
  ↳the form of key:value pairs. To add a database property, the following format must
  ↳be followed.>"
      {
        "connectionPropertyId": "<To add a database property, set
        ↳this value as 0. To update an existing database property, set this value as the
        ↳corresponding Key Id.>" ,
        "databasePropertyKey": "<This is the database property.>",
        "databasePropertyValue": "<This is the value of the database
        ↳property.>"
      }
    ],
  *"dbType": "<This is the source type of the relational database. The probable
  ↳values are the ORACLE, MYSQL, SQLSERVER, POSTGRESQL, TERADATA.>",
  *"driverName": "<This is the fully qualified driver class name based on the
  ↳defined relational database (or dbType).>",
  "dbVersion": "<This is the version of the relational database that needs to
  ↳be connected. This parameter value acts only as a marker.>",
  "schemaName": "<This is the name of the database schema to which the
  ↳connection points. Generally, the schema name may apply to certain relational
  ↳databases and can be a mandatory parameter for those database sources (or dbType)
  ↳only.>",
  "userName": "<This is the username of the relational database. This
  ↳parameter is mandatory while saving a new database connection.>",
  "userPassword": "<This is the password of the database (if any) that needs
  ↳to be connected. This parameter is mandatory while saving a new database connection.
  ↳>",
  "storeUsernamePassword": "<If you want ZDP application to store the database
  ↳credentials, set this value to true. If not, set the value as false.>",
  "scope": "<This is the connection type. For example, PUBLIC or PRIVATE. A
  ↳private connection can be updated only by the user who created the connection. If
  ↳you do not provide any value for this parameter, the system defaults the value to
  ↳PUBLIC.>"
}
```

Example Request:

To add a *MYSQL* database connection:

```
{
  "connectionInstanceId": 0,
  "connectionInstanceName": "MYSQL_88 Connect",
  "description": "",
  "connectionType": "RDBMS",
  "databaseUri": "jdbc:mysql://192.168.2.88:3306/bedrockInternal",
  "databaseProperties": [
    {

```

```

        "connectionPropertyId": 0,
        "databasePropertyKey": "hra_comp",
        "databasePropertyValue": "true"
    },
    {
        "connectionPropertyId": 0,
        "databasePropertyKey": "appliance",
        "databasePropertyValue": "false"
    }
],
"dbType": "MYSQL",
"dbVersion": "",
"schemaName": "",
"userName": "*****",
"userPassword": "*****",
"driverName": "com.mysql.jdbc.Driver",
"storeUsernamePassword": false,
"scope": "PUBLIC"
}

```

Example Response:

If a database connection is saved successfully:

```

{
    "responseMessage": "Connection Instance 9 saved successfully",
    "restUri": "/bedrock-app/services/rest/ingestion/database5/connections",
    "result": {
        "connectionInstanceId": 9,
        "connectionInstanceName": "MYSQL_88 Connect",
        "description": "",
        "createdBy": "admin",
        "createdTime": "12/20/2016 18:21:50",
        "modifiedBy": "admin",
        "modifiedTime": "12/20/2016 18:21:50",
        "databaseId": 5,
        "connectionType": "RDBMS",
        "databaseUri": "jdbc:mysql://192.168.2.88:3306/bedrockInternal",
        "databaseProperties": [
            {
                "connectionPropertyId": 1,
                "databasePropertyKey": "hra_comp",
                "databasePropertyValue": "true"
            },
            {
                "connectionPropertyId": 2,
                "databasePropertyKey": "appliance",
                "databasePropertyValue": "false"
            }
        ],
        "dbType": "MYSQL",
        "dbVersion": "",
        "schemaName": "",
        "userName": null,
        "userPassword": null,
        "driverName": "com.mysql.jdbc.Driver",
        "storeUsernamePassword": false,
        "scope": "PUBLIC"
    }
}

```

```

    },
    "page": null
}

```

Response in Case of Failure:

If a connection with the same name or URI exists:

```

{
    "responseMessage": "java.lang.IllegalStateException: Connection instance with_
↪same name or uri already exists",
    "restUri": "/bedrock-app/services/rest/ingestion/database/5/connections",
    "result": null,
    "page": null
}

```

1.4.2 Fetch all Connection Details

This API allows you to fetch the details of all the connections existing in a ZDP instance. This service has provisions for advanced search options as well.

URL:

```

http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/{connection_type}/
↪connections/search

```

Note:

- You must pass a value for the *connection_type* parameter in the URL.
- To fetch details of all the filesystem connections, such as *LOCAL*, *HDFS*, *AMAZON S3*, or *AZURE BLOB STORAGE*; set the *connection_type* parameter as *filesystems* in the URL.
- To fetch details of all the *RDBMS* type connections, set the *connection_type* parameter as *database* in the URL.

Method:

```
POST
```

Request Payload:

```

{
    "currentPage": "<Set the page number from which the results need to be_
↪fetched. This is dependent on the set chunkSize.>",
    "chunkSize": "<This is the number of records (or page-size) that must be_
↪fetched within a page. Set a numerical value to fetch records corresponding to the_
↪specified value. When set as '0' (or unspecified), all results are fetched.>",
    "sortBy": "<Sort the results based on the following properties. For example,_
↪modifiedTime.>",
    "orderBy": "<This is the sequence in which the results are ordered by. For_
↪example, set DESC for descending or ASC for ascending.>",
    "searchCriteria": "<This is the search criteria based on which the matching_
↪connection details are fetched. The search mode is LIKE and any combination of the_
↪search parameters can be used.>"
}

```

```

        "searchMode": "<Specify the type of search that must be performed by
        ↳ using the value set for the searchValue parameter. Set LIKE to perform a like
        ↳ search, EXACT to match the exact value, NOT to perform an invert search, or RANGE
        ↳ to perform a search using a defined range.>",
        "searchAttribute": "<This is the name of the attribute based on which
        ↳ the search is performed. Leave it blank if not applicable. Probable values can be
        ↳ connectionInstanceId, connectionInstanceName, databaseUri, connectionType.>",
        "searchValue": "<Provide the value for which search is performed.>",
        "dataType": "<This is the type of the range and is a mandatory
        ↳ parameter when the searchMode is set as RANGE. Set DATE to search connection
        ↳ instances using a date range, or INT to search between a range of connection Ids.>",
        "rangeStart": "<Set the start date/connection Id from which the
        ↳ connections details need to be fetched. This parameter is mandatory if the
        ↳ searchMode is set as RANGE; else set the value as null.>",
        "rangeEnd": "<Set the end date/connection Id for which the connection
        ↳ instances need to be fetched. This parameter is mandatory if the searchMode is set
        ↳ as RANGE; else set the value as null.>"
    }
}

```

Note: You can use single/multiple parameters to fetch the details of the connection list. You can perform an advanced search by using an array of search options under *searchCriteria*. (Refer to Example Request 2).

Example Request:

- Scenario 1:

```

{
    "currentPage": 1,
    "chunkSize": 10,
    "sortBy": "modifiedTime",
    "orderBy": "DESC",
    "searchCriteria": []
}

```

- Scenario 2:

```

{
    "currentPage": 1,
    "chunkSize": "5",
    "sortBy": "modifiedTime",
    "orderBy": "DESC",
    "searchCriteria": [
        {
            "searchMode": "LIKE",
            "searchAttribute": "connectionInstanceName",
            "searchValue": "s3_",
            "dataType": "",
            "rangeStart": null,
            "rangeEnd": null
        }
    ]
}

```

Example Response:

- Scenario 1:

To fetch the details of all the filesystem connections existing in a ZDP instance:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Success",
    "result": null
  },
  "result": {
    "resultList": [
      {
        "connectionInstanceId": 1,
        "connectionInstanceName": "Local File System",
        "description": "This connection is used to",
        "createdBy": "admin",
        "createdTime": "12/26/2016 20:06:21",
        "modifiedBy": "admin",
        "modifiedTime": "12/26/2016 20:06:21",
        "filesystemId": 1,
        "filesystem": "LOCAL",
        "filesystemUri": "file:///",
        "filesystemProperties": [],
        "scope": "PUBLIC"
      },
      {
        "connectionInstanceId": 2,
        "connectionInstanceName": "HDFSConnection",
        "description": "Only API",
        "createdBy": "admin",
        "createdTime": "12/27/2016 12:47:12",
        "modifiedBy": "admin",
        "modifiedTime": "12/27/2016 12:47:12",
        "filesystemId": 2,
        "filesystem": "HDFS",
        "filesystemUri": "hdfs://hdp-node1.zalonilabs.",
        "filesystemProperties": [],
        "scope": "PUBLIC"
      },
      {
        "connectionInstanceId": 4,
        "filesystemId": 3,
        "filesystem": "AMAZON S3",
        "filesystemUri": "s3n://bedrock-ing2",
        "connectionInstanceName": "s3_con1",
        "filesystemProperties": [
          {
            "connectionPropertyId": 4,
            "filesystemPropertyKey": "fs.s3n.",
            "filesystemPropertyValue": "AKIAJVRJ2Y4HCAILVKA"
          }
        ]
      }
    ]
  }
}
```



```

        "connectionPropertyId": 5,
        "fileSystemPropertyKey": "fs.s3n.
↪awsSecretAccessKey",
        "fileSystemPropertyValue":
↪"u7Coz2n1VYMLnKa24KgIR6wFweLRQb04IRw/sjjE"
    },
    ],
    "description": "",
    "createdBy": null,
    "createdTime": "03/01/2016 13:55:38",
    "modifiedBy": null,
    "modifiedTime": "03/02/2016 12:44:35",
    "scope": "PUBLIC"
  }
},
"totalRecords": 3,
"currentPage": 1,
"chunkSize": 5
}
}

```

- Scenario 2:

To fetch the details of all the database (or *RDBMS*) connections existing in a ZDP instance:

```

{
  "responseMessage": "Success",
  "restUri": "/bedrock-app/services/rest/ingestion/database/connections/
↪search",
  "result": {
    "resultList": [
      {
        "connectionInstanceId": 2,
        "connectionInstanceName": "MySQL",
        "description": "",
        "createdBy": "admin",
        "createdTime": "12/23/2016 18:28:29",
        "modifiedBy": "admin",
        "modifiedTime": "12/23/2016 18:28:29",
        "databaseId": 5,
        "connectionType": "RDBMS",
        "databaseUri": "jdbc:mysql://192.168.1.203:3306/
↪bedrock430",
        "databaseProperties": [],
        "dbType": "MYSQL",
        "dbVersion": "",
        "schemaName": "",
        "userName": "bedrockdba",
        "userPassword": "ENC(HCObiluPT8/139niQe/
↪PK26dho5V3BL1)",
        "driverName": "com.mysql.jdbc.Driver",
        "storeUsernamePassword": true,
        "scope": "PUBLIC"
      },
      {
        "connectionInstanceId": 9,
        "connectionInstanceName": "MYSQL_88 Connect",
        "description": "",

```

```

        "createdBy": "admin",
        "createdTime": "12/20/2016 18:21:50",
        "modifiedBy": "admin",
        "modifiedTime": "12/20/2016 18:21:50",
        "databaseId": 5,
        "connectionType": "RDBMS",
        "databaseUri": "jdbc:mysql://192.168.2.88:3306/
↪bedrock421dec",
        "databaseProperties": [
            {
                "connectionPropertyId": 1,
                "databasePropertyKey": "hra_comp",
                "databasePropertyValue": "true"
            },
            {
                "connectionPropertyId": 2,
                "databasePropertyKey": "appliance
↪",
                "databasePropertyValue": "false"
            }
        ],
        "dbType": "MYSQL",
        "dbVersion": "",
        "schemaName": "",
        "userName": null,
        "userPassword": null,
        "driverName": "com.mysql.jdbc.Driver",
        "storeUsernamePassword": false,
        "scope": "PUBLIC"
    },
    "totalRecords": 2,
    "currentPage": 1,
    "chunkSize": 20
},
"page": null
}

```

Response in Case of Failure:

- Scenario 1:

If the search criteria does not match any of the existing filesystem connections:

```

{
    "status": {
        "responseType": "ERROR",
        "responseCode": 500,
        "responseMessage": "No file system connections found",
        "result": null
    },
    "result": null
}

```

- Scenario 2:

If the search criteria does not match any of the existing database (or *RDBMS* type) connections:

```
{
  "responseMessage": "No database connections found",
  "restUri": "/bedrock-app/services/rest/ingestion/database/connections/search",
  "result": null,
  "page": null
}
```

1.4.3 Fetch Connection Details based on Filesystem/Database

This API allows you to fetch the existing connections based on the file system (such as *LOCAL*, *HDFS*, *AMAZON S3*, or *AZURE BLOB STORAGE*) or relational database (such as *RDBMS*).

URL:

- Fetch connection details based on the type of the filesystem.

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/filesystems/{file_
↪system_Id}/connections
```

You must pass the *file_system_Id* parameter in the API URL.

- 1 for *LOCAL*
- 2 for *HDFS*
- 3 for *AMAZON S3*
- 4 for *AZURE BLOB STORAGE*

Note: If the *file_system_Id* parameter is set to 0 in the URL, all connection instances irrespective of the file systems, are fetched.

- Fetch connection details based on the database.

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/database/{database_
↪Id}/connections
```

You must pass the *database_Id* parameter in the URL as 5.

Method:

```
GET
```

Example Response:

- Scenario 1:

If the *file_system_Id* parameter in the URL is set to 3 (indicating *Amazon S3* connections):

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Success",
    "result": null
  },
  "result": [
    {
```

```

        "connectionInstanceId": 4,
        "fileSystemId": 3,
        "fileSystem": "AMAZON S3",
        "fileSystemUri": "s3n://bedrock-testing2",
        "connectionInstanceName": "s3_con1",
        "fileSystemProperties": [
            {
                "connectionPropertyId": 4,
                "fileSystemPropertyKey": "fs.s3n.
↪awsAccessKeyId",
                "fileSystemPropertyValue":
↪"AKIAJVRJYJ2Y4HCAILVKA"
            },
            {
                "connectionPropertyId": 5,
                "fileSystemPropertyKey": "fs.s3n.
↪awsSecretAccessKey",
                "fileSystemPropertyValue":
↪"u7Coz2n1VYMLnKa24KgIR6wFweLRQb04IRw/sjjE"
            }
        ],
        "description": "",
        "createdBy": null,
        "createdTime": "03/01/2016 13:55:38",
        "modifiedBy": null,
        "modifiedTime": "03/02/2016 12:44:35",
        "scope": "PUBLIC"
    },
    {
        "connectionInstanceId": 13,
        "fileSystemId": 3,
        "fileSystem": "AMAZON S3",
        "fileSystemUri": "s3n://bedrock-testing",
        "connectionInstanceName": "s3_con2",
        "fileSystemProperties": [
            {
                "connectionPropertyId": 9,
                "fileSystemPropertyKey": "fs.s3n.
↪awsAccessKeyId",
                "fileSystemPropertyValue":
↪"AKIAJVRJYJ2Y4HCAILVKA"
            },
            {
                "connectionPropertyId": 10,
                "fileSystemPropertyKey": "fs.s3n.
↪awsSecretAccessKey",
                "fileSystemPropertyValue":
↪"u7Coz2n1VYMLnKa24KgIR6wFweLRQb04IRw/sjjE"
            }
        ],
        "description": "",
        "createdBy": null,
        "createdTime": "03/02/2016 12:46:51",
        "modifiedBy": null,
        "modifiedTime": "03/02/2016 12:46:51",
        "scope": "PUBLIC"
    }
]

```

```
}

```

- Scenario 2:

If the `database_Id` parameter in the URL is set as 5 (indicating *RDBMS* connection type):

```
{
  "responseMessage": "Success",
  "restUri": "/bedrock-app/services/rest/ingestion/database/5/connections",
  "result":
  [
    {
      "connectionInstanceId": 3,
      "connectionInstanceName": "MYSQL Connect",
      "description": "",
      "createdBy": "admin",
      "createdTime": "12/19/2016 16:32:21",
      "modifiedBy": "admin",
      "modifiedTime": "12/19/2016 16:32:21",
      "databaseId": 5,
      "connectionType": "RDBMS",
      "databaseUri": "jdbc:mysql://192.168.1.203:3306/bedrock_
↪unit_test",
      "databaseProperties": [],
      "dbType": "MYSQL",
      "dbVersion": "",
      "schemaName": "",
      "userName": "bedrockdba",
      "userPassword": "ENC(/dFSQc1E1ZX8sBTJUJx+U7GqQKnaNwF)",
      "driverName": "com.mysql.jdbc.Driver",
      "storeUsernamePassword": true,
      "scope": "PUBLIC"
    },
    {
      "connectionInstanceId": 4,
      "connectionInstanceName": "MS",
      "description": "",
      "createdBy": "admin",
      "createdTime": "12/19/2016 17:42:49",
      "modifiedBy": "admin",
      "modifiedTime": "12/19/2016 17:44:58",
      "databaseId": 5,
      "connectionType": "RDBMS",
      "databaseUri": "jdbc:sqlserver://192.168.1.197:1433;
↪databaseName=testjoydip",
      "databaseProperties": [],
      "dbType": "SQLSERVER",
      "dbVersion": "",
      "schemaName": "",
      "userName": "sa",
      "userPassword": "ENC(C6qM1v4N2mYz+7hU7YYHWyN/In3Xgmb+)",
      "driverName": "com.microsoft.sqlserver.jdbc.
↪SQLServerDriver",
      "storeUsernamePassword": true,
      "scope": "PUBLIC"
    },
    {
      "connectionInstanceId": 9,

```

```

        "connectionInstanceName": "MYSQL_88 Connect",
        "description": "",
        "createdBy": "admin",
        "createdTime": "12/20/2016 18:21:50",
        "modifiedBy": "admin",
        "modifiedTime": "12/20/2016 18:21:50",
        "databaseId": 5,
        "connectionType": "RDBMS",
        "databaseUri": "jdbc:mysql://192.168.2.88:3306/
↪bedrock20dec",
        "databaseProperties": [
            {
                "connectionPropertyId": 1,
                "databasePropertyKey": "hra_comp",
                "databasePropertyValue": "true"
            },
            {
                "connectionPropertyId": 2,
                "databasePropertyKey": "appliance",
                "databasePropertyValue": "false"
            }
        ],
        "dbType": "MYSQL",
        "dbVersion": "",
        "schemaName": "",
        "userName": null,
        "userPassword": null,
        "driverName": "com.mysql.jdbc.Driver",
        "storeUsernamePassword": false,
        "scope": "PUBLIC"
    },
    "page": null
}

```

- Scenario 3:

If no connections exist for the specified type of filesystem:

```

{
    "status": {
        "responseType": "info",
        "responseCode": 200,
        "responseMessage": "Success",
        "result": null
    },
    "result": null
}

```

- Scenario 4:

If no *RDBMS* connections exist:

```

{
    "responseMessage": "No Connections Found with Connection Type Id : 5",
    "restUri": null,
    "result": null,
    "page": null
}

```

1.4.4 Fetch Connection Details using Connection Instance Id

This API allows you to fetch the connection details based on the connection instance Id.

Note: The *connection_instance_Id* parameter must be passed in the API URL.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/{connection_type}/  
connections/{connection_instance_Id}
```

Note:

- To fetch the details of connection instance that belongs to any of the filesystems, set the *connection_type* parameter as *filesystems* in the above URL.
- To fetch the details of a connection instance of *RDBMS* type, set the *connection_type* parameter as *database* in the above URL.

Method:

```
GET
```

Example Response:

- Scenario 1:

If the connection instance Id belongs to a specific filesystem:

```
{  
  "status": {  
    "responseType": "info",  
    "responseCode": 200,  
    "responseMessage": "Success",  
    "result": null  
  },  
  "result": {  
    "connectionInstanceId": 5,  
    "connectionInstanceName": "HDFS CONNECT",  
    "description": "",  
    "createdBy": "admin",  
    "createdTime": "12/20/2016 16:21:28",  
    "modifiedBy": "admin",  
    "modifiedTime": "12/20/2016 16:21:28",  
    "fileSystemId": 2,  
    "fileSystem": "HDFS",  
    "fileSystemUri": "hdfs://192.168.1.203:8020",  
    "fileSystemProperties": [],  
    "scope": "PUBLIC"  
  }  
}
```

- Scenario 2:

If the connection instance Id belongs to a relational database:

```
{
  "responseMessage": "Success",
  "restUri": "/bedrock-app/services/rest/ingestion/database/connections/9",
  "result": {
    "connectionInstanceId": 9,
    "connectionInstanceName": "MYSQL_88 Connect",
    "description": "",
    "createdBy": "admin",
    "createdTime": "12/20/2016 18:21:50",
    "modifiedBy": "admin",
    "modifiedTime": "12/20/2016 18:21:50",
    "databaseId": 5,
    "connectionType": "RDBMS",
    "databaseUri": "jdbc:mysql://192.168.2.88:3306/bedrock20dec",
    "databaseProperties": [
      {
        "connectionPropertyId": 1,
        "databasePropertyKey": "hra_comp",
        "databasePropertyValue": "true"
      },
      {
        "connectionPropertyId": 2,
        "databasePropertyKey": "appliance",
        "databasePropertyValue": "false"
      }
    ],
    "dbType": "MYSQL",
    "dbVersion": "",
    "schemaName": "",
    "userName": null,
    "userPassword": null,
    "driverName": "com.mysql.jdbc.Driver",
    "storeUsernamePassword": false,
    "scope": "PUBLIC"
  },
  "page": null
}
```

Response in Case of Failure:

If there is no record that exists for the speified connection instance Id:

```
{
  "status": {
    "responseType": "ERROR",
    "responseCode": 500,
    "responseMessage": "No file system connection instance found",
    "result": null
  },
  "result": null
}
```


1.4.5 Delete an Existing Connection

This API allows you to delete an existing connection irrespective of the file system (*LOCAL*, *HDFS*, *S3*, or *AZURE*) by using *connectionInstanceId*. To get the *connectionInstanceId*, refer to the [Fetch all Connection Details](#) API.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/{connection_type}/  
↪connections/{connection_instance_Id}
```

Note:

- You must pass the *connection_type* parameter in the URL to indicate the type of connection under which a connection instance needs to be deleted.
 - Set the value for the *connection_type* parameter as *filesystems* in the URL to delete a connection instance that belongs to any of the filesystem.
 - Set the value for the *connection_type* parameter as *database* in the URL to delete a database connection instance.
- You must also pass the value for *connection_instance_Id* parameter in the API URL.

Method:

```
DELETE
```

Example Response:

- Scenario 1:

If a filesystem connection instance is deleted:

```
{  
  "status": {  
    "responseType": "info",  
    "responseCode": 200,  
    "responseMessage": "Connection Instance deleted successfully",  
    "result": null  
  },  
  "result": "Connection Instance deleted successfully"  
}
```

- Scenario 2:

If a database connection instance is deleted:

```
{  
  "responseMessage": "Success",  
  "restUri": "/bedrock-app/services/rest/ingestion/database/connections/3",  
  "result": "Connection Instance deleted successfully",  
  "page": null  
}
```

Response in Case of Failure:

- Scenario 1:

If the filesystem connection instance Id is not found (or does not exist):

```
{
  "status": {
    "responseType": "warn",
    "responseCode": 412,
    "responseMessage": "Connection Instance not found",
    "result": null
  },
  "result": "Connection Instance not found"
}
```

- Scenario 2:

If the database connection instance Id is not found (or does not exist):

```
{
  "responseMessage": "No Connections Found with Connection Type Id : 5",
  "restUri": null,
  "result": null,
  "page": null
}
```

1.4.6 Fetch All Landing Zone Servers

This API allows you to fetch all the landing zone Servers.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/publish/
↪ retrieveLandingZoneServers
```

Method:

```
GET
```

Example Response:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Success",
    "result": null
  },
  "result": {
    "data": [
      {
        "serverId": "1",
        "serverName": "Local",
        "ipAddress": "192.168.1.85",
        "username": "bedrock",
        "password": "password",
        "pemFilePath": null,
        "encryption": null,
        "updateTs": 1453384880000,
        "lzDirectories": [
          {
            "lzDirId": 42,

```

```

    "dirPath": "/home/bedrock/mbaruahl23",
    "description": "",
    "totalFPCount": 0,
    "totalSlaves": 8,
    "serverId": 1,
    "serverName": "Local",
    "fileSystemId": 1,
    "fileSystemName": "LOCAL",
    "notificationDetails": null,
    "fileSystemUri": "file:///",
    "fileSystemConnectionInstanceId": 1,
    "fileSystemConnectionInstanceName":

↪ "Local-Test",

    "allowIngestion": false
  }
]
},
"totalRecords": 1,
"currentPage": 0,
"chunkSize": 0
}
}

```

1.4.7 Add Source Directory to a Server

This API allows you to add a source directory under a landing zone server.

URL:

```

http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/publish/
↪ saveLZDirectories

```

Method:

POST

Request Payload:

```

{
  *"fileSystemId": "<This is the file system where the source directory is located. Use
  ↪ 1 for LOCAL, 2 for HDFS, 3 for S3, 4 for AZURE. This value is non-editable. If you
  ↪ do not provide this value, this defaults to 1.>",
  "totalSlaves": "<This is the number of slaves associated with the master agent.>",
  *"serverId": "<This is the Id of the landing zone server.>",
  *"dirPath": "<This is the path for the source directory. This API creates the source
  ↪ directory even if the specified location does not exist under the server,
  ↪ physically.>",
  "description": "<This is the description of the landing zone directory.>",
  *"fileSystemUri": "<This is the file system URI for the storage system on which the
  ↪ source directory is created. This comprises of the format: <authority://scheme>.
  ↪ Since the file system uri is mentioned with connection instance, this will be
  ↪ deprecated and removed in the future versions of ZDP.>",
  "notificationDetails": [
    {
      "eventId": "<This is the Id of the ingestion event for which notification
      ↪ mail is intended. Applicable for local source directory only.>",

```

```

        "distributionIdList": ["<This is the comma-separated distribution list Ids of
↳the distribution list defined in the ZDP application. Applicable for local source
↳directory only.>"]
    }
},
* "fileSystemConnectionInstanceId": "<This is the connection instance associated with
↳the landing zone directory. The connection instance must be present before creating
↳landing zone directory.>",
* "allowIngestion": "<This is the flag that decides if the source directory can be
↳used for ingestion or monitoring. You can set this parameter to true for ingestion
↳and false for monitoring.>",
  "projects": [
    {
      "id": "<This is the Id of the project to which you want to restrict this source
↳directory.>",
      "name": "<This is the name of the project.>",
      "description": "<This is the description for the project.>",
      "createdBy": "<This is the user who created the project.>",
      "modifiedBy": "<This is the user who modified the project.>"
    }
  ],
  "runAsUser": "<This is the user who can start/stop the BDCA agent.>"
* "lzDirId": "<This is the Id of the source directory. The value is zero for a new
↳landing zone directory.>",
* "allowsFailOver": "<This is the option that allows the BDCA (corresponding to the
↳source directory) to fail over to an active ingestion warden that belongs to the
↳same cluster as the parent warden for the failed BDCA. To enable fail-over, set
↳this parameter to true.>",
}

```

Example Request:

```

{
  "fileSystemId": 1,
  "totalSlaves": "1",
  "serverId": "1",
  "dirPath": "/home/bedrock/jeet/testLz",
  "description": "",
  "fileSystemUri": "",
  "notificationDetails": [],
  "fileSystemConnectionInstanceId": 1,
  "allowIngestion": true,
  "projects": [
    {
      "id": 2,
      "name": "project1",
      "description": "",
      "createdBy": "admin",
      "modifiedBy": "admin",
      "roleAssociations": null
    }
  ],
  "runAsUser": "do2user1",
  "lzDirId": 0
}

```

Example Response:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Success",
    "result": 79
  },
  "result": "LZ Directory '/home/bedrock/jeet/testLz' saved successfully"
}
```

Note: The Id of the newly added source directory is returned as the value against the *result* property.

Response in Case of Failure:

If the provided landing zone server Id is incorrect:

```
{
  "result": "Landing Zone Server was not found",
  "status": {
    "responseCode": 412,
    "result": null,
    "responseMessage": "Success",
    "responseType": "warn"
  }
}
```

1.4.8 Fetch Source Directories of a Server

This API allows you to fetch all the directories of a landing zone server.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/publish/
↳getLzDirectoriesByServer/{server_Id}
```

Note: You must pass the *server_Id* parameter in the API URL.

Method:

GET

Example Response:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Success",
    "result": null
  },
  "result": {
    "data": [
      {

```

```

        "lzDirId": 27,
        "dirPath": "/user/bedrock/jeet/testLz",
        "description": "",
        "totalFPCount": 0,
        "totalSlaves": 1,
        "serverId": 5,
        "serverName": "192.168.1.186",
        "fileSystemId": 1,
        "fileSystemName": "LOCAL",
        "notificationDetails": null,
        "fileSystemUri": "file:///",
        "fileSystemConnectionInstanceId": 1,
        "fileSystemConnectionInstanceName": "Local Conn",
        "allowIngestion": true
        "modifiedTime": 1535977476000,
        "allowsFailOver": false,
        "createdTime": null,
        "createdBy": null,
        "modifiedBy": null,
        "isRestrictedToProjects": null,
        "projects": [],
        "runAsUser": "zaloni"
    },
    {
        "lzDirId": 26,
        "dirPath": "/home/bedrock/jeet/lz",
        "description": "",
        "totalFPCount": 0,
        "totalSlaves": 8,
        "serverId": 5,
        "serverName": "192.168.1.186",
        "fileSystemId": 1,
        "fileSystemName": "LOCAL",
        "notificationDetails": null,
        "fileSystemUri": "file:///",
        "fileSystemConnectionInstanceId": 1,
        "fileSystemConnectionInstanceName": "Local Conn",
        "allowIngestion": true
        "modifiedTime": 1535977476000,
        "allowsFailOver": false,
        "createdTime": null,
        "createdBy": null,
        "modifiedBy": null,
        "isRestrictedToProjects": null,
        "projects": [],
        "runAsUser": "zaloni"
    }
],
    "totalRecords": 2,
    "currentPage": 1,
    "chunkSize": 1000
}

```

Response in Case of Failure:

If the provided server Id does not exist:

```
{
  "result": "No record(s) found",
  "status": {
    "responseCode": 204,
    "result": null,
    "responseMessage": "Success",
    "responseType": "info"
  }
}
```

1.4.9 Get all Flume Ingestion Servers

This API allows you to fetch all Flume ingestion servers.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/flume/publish/
↪getLandingZoneFlumeServers
```

Method:

```
GET
```

Example Response:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Success",
    "result": null
  },
  "result": {
    "data": [
      {
        "flServerId": 1,
        "flServerName": "localhost",
        "flServerIpAddress": "129.168.1.246",
        "flServerUsername": "cloudera",
        "flServerPassword": "password",
        "flServerIngestionFailedDir": "/home/cloudera/lz",
        "flServerUpdateTs": 1425593681000,
        "flServerInstallationDir": "/home/cloudera/localhost",
        "flServerInstallationMode": 1,
        "flServerInstallationStatus": 2
      }
    ],
    "totalRecords": 1,
    "currentPage": 0,
    "chunkSize": 0
  }
}
```

Response in Case of Failure:

If the user is not authenticated:

```
{
  "result": "User must login first.",
  "status": {
    "responseCode": 401,
    "responseMessage": "User must login first.",
    "responseType": "info",
    "result": null
  }
}
```

1.4.10 Create Flume Ingestion Agent

This API allows you to create a new Flume agent within a project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/{project_Id}/flumeAgents
```

Note: You must pass the *project_Id* parameter in the API URL. To create a Flume ingestion agent under the public project, set the *project_Id* parameter to *1*.

Method:

```
POST
```

Example Request:

```
{
  *"flServerId": 2,
    "flAgentId": 0,
    "flServerName": "",
    "flServerInstallationDir": "",
    "flAgentStateId": 1,
  *"flAgentName": "agent1",
  *"flAgentPort": "1234",
  *"flSourceTypeId": "1",
  *"flSourceProperties": [
    {
      "flAgentId": 0,
      "flSourcePropertyId": "1",
      "flSourcePropertyKeyId": 1,
      "flSourcePropertyKey": "threads",
      "flSourcePropertyValue": "3"
    },
    {
      "flAgentId": 0,
      "flSourcePropertyId": "1",
      "flSourcePropertyKeyId": 2,
      "flSourcePropertyKey": "compression-type",
      "flSourcePropertyValue": ""
    },
    {
      "flAgentId": 0,
      "flSourcePropertyId": "1",
      "flSourcePropertyKeyId": 3,
```



```

        "flSourcePropertyKey": "ipFilter",
        "flSourcePropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSourcePropertyId": "1",
        "flSourcePropertyKeyId": 4,
        "flSourcePropertyKey": "ipFilter.rules",
        "flSourcePropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSourcePropertyId": "1",
        "flSourcePropertyKeyId": 5,
        "flSourcePropertyKey": "interceptors",
        "flSourcePropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSourcePropertyId": "1",
        "flSourcePropertyKeyId": 6,
        "flSourcePropertyKey": "interceptors.*",
        "flSourcePropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSourcePropertyId": "1",
        "flSourcePropertyKeyId": 7,
        "flSourcePropertyKey": "ssl",
        "flSourcePropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSourcePropertyId": "1",
        "flSourcePropertyKeyId": 8,
        "flSourcePropertyKey": "keystore",
        "flSourcePropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSourcePropertyId": "1",
        "flSourcePropertyKeyId": 9,
        "flSourcePropertyKey": "keystore-password",
        "flSourcePropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSourcePropertyId": "1",
        "flSourcePropertyKeyId": 10,
        "flSourcePropertyKey": "keystore-type",
        "flSourcePropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSourcePropertyId": "1",
        "flSourcePropertyKeyId": 11,
        "flSourcePropertyKey": "exclude-protocols",
        "flSourcePropertyValue": ""
    }

```

```

    },
    *"flChanTypeId": "1",
    "flChanProperties": [
      {
        "flAgentId": 0,
        "flChanPropertyId": "1",
        "flChanPropertyKeyId": 1,
        "flChanPropertyKey": "capacity",
        "flChanPropertyValue": "10000"
      },
      {
        "flAgentId": 0,
        "flChanPropertyId": "1",
        "flChanPropertyKeyId": 2,
        "flChanPropertyKey": "transactionCapacity",
        "flChanPropertyValue": ""
      },
      {
        "flAgentId": 0,
        "flChanPropertyId": "1",
        "flChanPropertyKeyId": 3,
        "flChanPropertyKey": "keep-alive",
        "flChanPropertyValue": ""
      },
      {
        "flAgentId": 0,
        "flChanPropertyId": "1",
        "flChanPropertyKeyId": 4,
        "flChanPropertyKey": "byteCapacityBufferPercentage",
        "flChanPropertyValue": ""
      },
      {
        "flAgentId": 0,
        "flChanPropertyId": "1",
        "flChanPropertyKeyId": 5,
        "flChanPropertyKey": "byteCapacity",
        "flChanPropertyValue": ""
      }
    ],
    *"flSinkTypeId": 1,
    "flSinkProperties": [
      {
        *"flAgentId": 0,
        *"flSinkPropertyId": 1,
        *"flSinkPropertyKeyId": 1,
        *"flSinkPropertyKey": "hdfs.path",
        *"flSinkPropertyValue": "/user/bedrock/fl"
      },
      {
        *"flAgentId": 0,
        *"flSinkPropertyId": 1,
        *"flSinkPropertyKeyId": 2,
        *"flSinkPropertyKey": "hdfs.rollInterval",
        *"flSinkPropertyValue": "30"
      },
      {
        *"flAgentId": 0,

```

```

        *"flSinkPropertyId": 1,
        *"flSinkPropertyKeyId": 3,
        *"flSinkPropertyKey": "hdfs.rollSize",
        *"flSinkPropertyValue": "0"
    },
    {
        *"flAgentId": 0,
        *"flSinkPropertyId": 1,
        *"flSinkPropertyKeyId": 4,
        *"flSinkPropertyKey": "hdfs.rollCount",
        *"flSinkPropertyValue": "0"
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 5,
        "flSinkPropertyKey": "hdfs.idleTimeout",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 6,
        "flSinkPropertyKey": "hdfs.batchSize",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 7,
        "flSinkPropertyKey": "hdfs.codeC",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 8,
        "flSinkPropertyKey": "hdfs.fileType",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 9,
        "flSinkPropertyKey": "hdfs.maxOpenFiles",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 10,
        "flSinkPropertyKey": "hdfs.minBlockReplicas",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 11,

```

```
        "flSinkPropertyKey": "hdfs.writeFormat",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 12,
        "flSinkPropertyKey": "hdfs.callTimeout",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 13,
        "flSinkPropertyKey": "hdfs.threadsPoolSize",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 14,
        "flSinkPropertyKey": "hdfs.rollTimerPoolSize",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 15,
        "flSinkPropertyKey": "hdfs.kerberosPrincipal",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 16,
        "flSinkPropertyKey": "hdfs.kerberosKeytab",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 17,
        "flSinkPropertyKey": "hdfs.proxyUser",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 18,
        "flSinkPropertyKey": "hdfs.round",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 19,
        "flSinkPropertyKey": "hdfs.roundValue",
        "flSinkPropertyValue": ""
    }
```

```

    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 20,
        "flSinkPropertyKey": "hdfs.timeZone",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 21,
        "flSinkPropertyKey": "hdfs.useLocalTimeStamp",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 22,
        "flSinkPropertyKey": "hdfs.closeTries",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 23,
        "flSinkPropertyKey": "hdfs.retryInterval",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 24,
        "flSinkPropertyKey": "serializer",
        "flSinkPropertyValue": ""
    },
    {
        "flAgentId": 0,
        "flSinkPropertyId": 1,
        "flSinkPropertyKeyId": 25,
        "flSinkPropertyKey": "serializer.*",
        "flSinkPropertyValue": ""
    }
],
*"filePatternList": [
    {
        "filePatternId": "1"
    }
]
}

```

Example Response:

```

{
    "responseMessage": "Flume Agent 'agent_test' saved successfully",
    "restUri": null,
    "result": "7",
    "page": null
}

```

```
}
```

Response in Case of Failure:

If the user is not authenticated:

```
{
  "responseMessage": "User does not have access to the file pattern id: 1",
  "restUri": "/bedrock-app/services/rest/projects/1/flumeAgents",
  "result": null,
  "page": null
}
```

1.4.11 Get Flume Ingestion Agent Details

This API allows you to get a Flume agent's details.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/flume/publish/
  ↪ getFlumeAgentDetails/{Flume_agent_Id}
```

Note: You must pass the *Flume_agent_Id* parameter in the API URL.

Method:

```
GET
```

Example Response:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Success",
    "result": null
  },
  "result": {
    "data": {
      "flAgentId": 1,
      "flAgentName": "agent1",
      "flAgentPort": 1234,
      "flAgentStateId": 3,
      "flServerId": 2,
      "flServerName": "myFlumeServer",
      "flServerIpAddress": "192.168.1.246",
      "flServerInstallationDir": "/home/cloudera/myFlumeServerDir",
      "flSourceType": 1,
      "flSourceType": "AVRO",
      "flChanTypeId": 1,
      "flChanType": "Memory",
      "flSinkTypeId": 1,
      "flSinkType": "BEDROCK HDFS",
      "flAgentUpdateTs": 1425595726000,
      "flAgentAliveSince": 1425595457000,
    }
  }
}
```

```
"flAgentIsAlive": true,
"flAgentIsStopped": false,
"filePatternList": [
  {
    "filePatternId": 1,
    "patternPrefix": "data",
    "patternSuffix": ".cvs",
    "destination": "",
    "fileSystemName": "HDFS",
    "entityId": null,
    "entityVersion": null,
    "entityName": null
  }
],
"flSourceProperties": [
  {
    "flSourcePropertyId": 1,
    "flAgentId": 1,
    "flSourcePropertyKey": "threads",
    "flSourcePropertyKeyId": 1,
    "flSourcePropertyLabel": "Threads",
    "flSourcePropertyIsRequired": false,
    "flSourcePropertyDefaultValue": "",
    "flSourcePropertyValue": "3"
  },
  {
    "flSourcePropertyId": 2,
    "flAgentId": 1,
    "flSourcePropertyKey": "compression-type",
    "flSourcePropertyKeyId": 2,
    "flSourcePropertyLabel": "Compression Type",
    "flSourcePropertyIsRequired": false,
    "flSourcePropertyDefaultValue": "inone",
    "flSourcePropertyValue": ""
  },
  {
    "flSourcePropertyId": 3,
    "flAgentId": 1,
    "flSourcePropertyKey": "ipFilter",
    "flSourcePropertyKeyId": 3,
    "flSourcePropertyLabel": "IP Filter",
    "flSourcePropertyIsRequired": false,
    "flSourcePropertyDefaultValue": "false",
    "flSourcePropertyValue": ""
  },
  {
    "flSourcePropertyId": 4,
    "flAgentId": 1,
    "flSourcePropertyKey": "ipFilter.rules",
    "flSourcePropertyKeyId": 4,
    "flSourcePropertyLabel": "IP Filter Rules",
    "flSourcePropertyIsRequired": false,
    "flSourcePropertyDefaultValue": "",
    "flSourcePropertyValue": ""
  },
  {
    "flSourcePropertyId": 5,
    "flAgentId": 1,
```

```

        "flSourcePropertyKey": "interceptors",
        "flSourcePropertyKeyId": 5,
        "flSourcePropertyLabel": "Interceptors",
        "flSourcePropertyIsRequired": false,
        "flSourcePropertyDefaultValue": "",
        "flSourcePropertyValue": ""
    },
    {
        "flSourcePropertyId": 6,
        "flAgentId": 1,
        "flSourcePropertyKey": "interceptors.*",
        "flSourcePropertyKeyId": 6,
        "flSourcePropertyLabel": "Interceptors.*",
        "flSourcePropertyIsRequired": false,
        "flSourcePropertyDefaultValue": "",
        "flSourcePropertyValue": ""
    },
    {
        "flSourcePropertyId": 7,
        "flAgentId": 1,
        "flSourcePropertyKey": "ssl",
        "flSourcePropertyKeyId": 7,
        "flSourcePropertyLabel": "SSL",
        "flSourcePropertyIsRequired": false,
        "flSourcePropertyDefaultValue": "false",
        "flSourcePropertyValue": ""
    },
    {
        "flSourcePropertyId": 8,
        "flAgentId": 1,
        "flSourcePropertyKey": "keystore",
        "flSourcePropertyKeyId": 8,
        "flSourcePropertyLabel": "Keystore",
        "flSourcePropertyIsRequired": false,
        "flSourcePropertyDefaultValue": "",
        "flSourcePropertyValue": ""
    },
    {
        "flSourcePropertyId": 9,
        "flAgentId": 1,
        "flSourcePropertyKey": "keystore-password",
        "flSourcePropertyKeyId": 9,
        "flSourcePropertyLabel": "Keystore Password",
        "flSourcePropertyIsRequired": false,
        "flSourcePropertyDefaultValue": "",
        "flSourcePropertyValue": ""
    },
    {
        "flSourcePropertyId": 10,
        "flAgentId": 1,
        "flSourcePropertyKey": "keystore-type",
        "flSourcePropertyKeyId": 10,
        "flSourcePropertyLabel": "Keystore Type",
        "flSourcePropertyIsRequired": false,
        "flSourcePropertyDefaultValue": "JKS",
        "flSourcePropertyValue": ""
    }
]

```



```

        "flSourcePropertyId": 11,
        "flAgentId": 1,
        "flSourcePropertyKey": "exclude-protocols",
        "flSourcePropertyKeyId": 11,
        "flSourcePropertyLabel": "Exclude Protocols",
        "flSourcePropertyIsRequired": false,
        "flSourcePropertyDefaultValue": "SSLv3",
        "flSourcePropertyValue": ""
    },
    "flChanProperties": [
        {
            "flChanPropertyId": 1,
            "flAgentId": 1,
            "flChanPropertyKey": "capacity",
            "flChanPropertyKeyId": 1,
            "flChanPropertyLabel": "Capacity",
            "flChanPropertyIsRequired": false,
            "flChanPropertyDefaultValue": "100",
            "flChanPropertyValue": "10000"
        },
        {
            "flChanPropertyId": 2,
            "flAgentId": 1,
            "flChanPropertyKey": "transactionCapacity",
            "flChanPropertyKeyId": 2,
            "flChanPropertyLabel": "Transaction Capacity",
            "flChanPropertyIsRequired": false,
            "flChanPropertyDefaultValue": "100",
            "flChanPropertyValue": ""
        },
        {
            "flChanPropertyId": 3,
            "flAgentId": 1,
            "flChanPropertyKey": "keep-alive",
            "flChanPropertyKeyId": 3,
            "flChanPropertyLabel": "Keep Alive",
            "flChanPropertyIsRequired": false,
            "flChanPropertyDefaultValue": "3",
            "flChanPropertyValue": ""
        },
        {
            "flChanPropertyId": 4,
            "flAgentId": 1,
            "flChanPropertyKey":
↪ "byteCapacityBufferPercentage",
            "flChanPropertyKeyId": 4,
            "flChanPropertyLabel": "Byte Capacity Buffer %
↪ ",
            "flChanPropertyIsRequired": false,
            "flChanPropertyDefaultValue": "20",
            "flChanPropertyValue": ""
        },
        {
            "flChanPropertyId": 5,
            "flAgentId": 1,
            "flChanPropertyKey": "byteCapacity",
            "flChanPropertyKeyId": 5,

```

```

        "flChanPropertyLabel": "Byte Capacity",
        "flChanPropertyIsRequired": false,
        "flChanPropertyDefaultValue": "",
        "flChanPropertyValue": ""
    },
],
"flSinkProperties": [
    {
        "flSinkPropertyId": 1,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.path",
        "flSinkPropertyKeyId": 1,
        "flSinkPropertyLabel": "HDFS Path",
        "flSinkPropertyIsRequired": true,
        "flSinkPropertyDefaultValue": "",
        "flSinkPropertyValue": "/user/cloudera/qa"
    },
    {
        "flSinkPropertyId": 2,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.rollInterval",
        "flSinkPropertyKeyId": 2,
        "flSinkPropertyLabel": "Rolling Interval",
        "flSinkPropertyIsRequired": true,
        "flSinkPropertyDefaultValue": "30",
        "flSinkPropertyValue": "30"
    },
    {
        "flSinkPropertyId": 3,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.rollSize",
        "flSinkPropertyKeyId": 3,
        "flSinkPropertyLabel": "Rolling Size",
        "flSinkPropertyIsRequired": true,
        "flSinkPropertyDefaultValue": "1024",
        "flSinkPropertyValue": "0"
    },
    {
        "flSinkPropertyId": 4,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.rollCount",
        "flSinkPropertyKeyId": 4,
        "flSinkPropertyLabel": "Rolling Count",
        "flSinkPropertyIsRequired": true,
        "flSinkPropertyDefaultValue": "10",
        "flSinkPropertyValue": "0"
    },
    {
        "flSinkPropertyId": 5,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.idleTimeout",
        "flSinkPropertyKeyId": 5,
        "flSinkPropertyLabel": "Idle Time Out",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "0",
        "flSinkPropertyValue": ""
    },
]

```

```

        "flSinkPropertyId": 6,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.batchSize",
        "flSinkPropertyKeyId": 6,
        "flSinkPropertyLabel": "Batch Size",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "100",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 7,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.codec",
        "flSinkPropertyKeyId": 7,
        "flSinkPropertyLabel": "Codec",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 8,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.fileType",
        "flSinkPropertyKeyId": 8,
        "flSinkPropertyLabel": "File Type",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "SequenceFile",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 9,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.maxOpenFiles",
        "flSinkPropertyKeyId": 9,
        "flSinkPropertyLabel": "Max Open Files",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "5000",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 10,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.minBlockReplicas",
        "flSinkPropertyKeyId": 10,
        "flSinkPropertyLabel": "Min Block Replicas",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 11,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.writeFormat",
        "flSinkPropertyKeyId": 11,
        "flSinkPropertyLabel": "Write Format",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "",
        "flSinkPropertyValue": ""
    }

```

```
    },
    {
        "flSinkPropertyId": 12,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.callTimeout",
        "flSinkPropertyKeyId": 12,
        "flSinkPropertyLabel": "Call Time Out",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "10000",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 13,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.threadsPoolSize",
        "flSinkPropertyKeyId": 13,
        "flSinkPropertyLabel": "Threads Pool Size",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "10",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 14,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.rollTimerPoolSize",
        "flSinkPropertyKeyId": 14,
        "flSinkPropertyLabel": "Roll Timer PoolSize",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "1",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 15,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.kerberosPrincipal",
        "flSinkPropertyKeyId": 15,
        "flSinkPropertyLabel": "Kerberos Principal",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 16,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.kerberosKeytab",
        "flSinkPropertyKeyId": 16,
        "flSinkPropertyLabel": "Kerberos Keytab",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 17,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.proxyUser",
        "flSinkPropertyKeyId": 17,
        "flSinkPropertyLabel": "Proxy User",
        "flSinkPropertyIsRequired": false,
```

```

        "flSinkPropertyDefaultValue": "",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 18,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.round",
        "flSinkPropertyKeyId": 18,
        "flSinkPropertyLabel": "Timestamp Rounded Down

→",

        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "false",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 19,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.roundValue",
        "flSinkPropertyKeyId": 19,
        "flSinkPropertyLabel": "Timestamp Rounded_

→Value",

        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "1",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 20,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.timeZone",
        "flSinkPropertyKeyId": 20,
        "flSinkPropertyLabel": "Time Zone",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "Local Time",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 21,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.useLocalTimeStamp",
        "flSinkPropertyKeyId": 21,
        "flSinkPropertyLabel": "Use Local TimeStamp",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 22,
        "flAgentId": 1,
        "flSinkPropertyKey": "hdfs.closeTries",
        "flSinkPropertyKeyId": 22,
        "flSinkPropertyLabel": "Close Tries",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "0",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 23,
        "flAgentId": 1,

```

```

        "flSinkPropertyKey": "hdfs.retryInterval",
        "flSinkPropertyKeyId": 23,
        "flSinkPropertyLabel": "Retry Interval",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "180",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 24,
        "flAgentId": 1,
        "flSinkPropertyKey": "serializer",
        "flSinkPropertyKeyId": 24,
        "flSinkPropertyLabel": "Serializer",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "TEXT",
        "flSinkPropertyValue": ""
    },
    {
        "flSinkPropertyId": 25,
        "flAgentId": 1,
        "flSinkPropertyKey": "serializer.*",
        "flSinkPropertyKeyId": 25,
        "flSinkPropertyLabel": "Serializer.*",
        "flSinkPropertyIsRequired": false,
        "flSinkPropertyDefaultValue": "",
        "flSinkPropertyValue": ""
    }
]
},
"totalRecords": 0,
"currentPage": 0,
"chunkSize": 0
}
}

```

Response in Case of Failure:

If the user is not authenticated:

```

{
    "result": "User must login first.",
    "status": {
        "responseCode": 401,
        "responseMessage": "User must login first.",
        "responseType": "info",
        "result": null
    }
}

```

1.4.12 Search Ingestion History

This API allows you to search the ingestion history, based on the specified criteria.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/ingestions/search
```

Method:

POST

Request Payload:

```

{
  *"ingestionProperties":
  {
    "inventoryId": [ "<This is the Id of ingestion history record.>" ],
    "status": [ "<This is the ingestion status and can be one of the following:
    ↳ SUCCESS, INGESTION FAILED, WORKFLOW FAILED, INCOMPLETE.>" ],
    "fileName": [ "<This is the name of the file. You can specify a list of
    ↳ string values for this parameter.>" ],
    "effectiveDate": [ "<This is the effective date.>" ],
    "ingestedAs": [ "<This is the user who performed the ingestion.>" ],
    "timestampFrom": "<This is the start time of the ingestion. The format is:
    ↳ <yyyy-MM-dd>T<HH:mm:ss>Z.>",
    "timestampTo": "<This is the end time of the ingestion. The format is: <yyyy-
    ↳ MM-dd>T<HH:mm:ss>Z.>",
    "sourceSizeFrom": "<This is the minimum size of the source.>",
    "sourceSizeTo": "<This is the maximum size of the source.>",
    "entityProperty": [
      {
        "propertyKey": "<This is the key for the entity property. The
        ↳ acceptable keys are entityIdVersion, entityName, dataFormat, subjectArea,
        ↳ sourcePlatform, sourceSchema.>",
        "propertyValue": [ "<This is the value for the entity property.>" ]
      }
    ],
    "filePatternProperty": [
      {
        "propertyKey": "<This is the key for the file pattern property. The
        ↳ acceptable keys are filePatternId, destinationPath, md5sum, and fileExtension.>",
        "propertyValue": [ "<This is the value for the file pattern property.>
        ↳ " ]
      }
    ],
    "landingZoneProperty": [
      {
        "propertyKey": "<This is the key for the landing zone property. The
        ↳ acceptable key is sourcePath.>"
        "propertyValue": [ "<This is the value for the landing zone property.>
        ↳ " ]
      }
    ]
  },
  *"pageProperties":
  {
    "chunkSize": "<This parameter determines the number of search results that
    ↳ must be fetched per page.>",
    "sortBy": "<This is the criterion according to which the fetched results are
    ↳ sorted. A few probable values are: ingestionTime, sourceSize, targetSize, startTime.
    ↳ >",
    "page": 0,
    "sortOrder": "<This is the sort order of the page. This can be ASC or DESC.>"
  }
}

```

Example Request:

```
{
  "ingestionProperties": {
  },
  "pageProperties": {
  }
}
```

Example Response:

- Scenario 1:

If you do not specify any parameters for *ingestionproperties* and *pageproperties*:

```
{
  "responseMessage": "Success",
  "restUri": "/bedrock-app/services/rest/ingestion/ingestions/search",
  "result": [
    {
      "id": "99dee1ba-686f-48fd-b6b1-8628a012a479",
      "ingestionTime": "1490604151012",
      "ingestionTimeFormatted": "2017-03-27 14:12:31.012",
      "sourceFile": "file:///home/bedrock/zps/victor8.txt",
      "filePatternId": 1,
      "triggerFile": "file:///home/bedrock/zps/victor8.done",
      "entity": null,
      "entityIdVersion": "0.0",
      "destFile": null,
      "sourceRowCount": 0,
      "targetRowCount": 0,
      "sourceSize": "700.0MB",
      "sourceSizeValue": "700.0MB",
      "targetSize": "0B",
      "targetSizeValue": null,
      "expirationDate": null,
      "targetTable": null,
      "publishedFlag": null,
      "wfInstanceId": 0,
      "additionalFiles": null,
      "notes": null,
      "md5sum": null,
      "status": "INGESTION FAILED",
      "effectiveDate": null,
      "error": "com.zaloni.bedrock.collection.agent.exceptions.
↪BedrockException: java.io.IOException: java.io.IOException: java.io.
↪IOException: Directory /home/pankaj/ could not be created",
      "subjectArea": null,
      "sourcePlatform": null,
      "sourceSchema": null,
      "format": null,
      "fileName": "victor8.txt",
      "fileExtension": ".txt",
      "controlFileContents": null,
      "lzServerName": "LZ Sewaiyaan",
      "lzDirPath": "/home/bedrock/zps",
      "ingestedAs": "",
      "fileChecksum": "NA",
    }
  ]
}
```



```

    "startTime": "1490604150708",
    "startTimeFormatted": "2017-03-27 14:12:30.708",
    "targetGB": "0B",
    "sourceGB": "700.0MB",
    "sourceGBStr": "700.0MB",
    "targetGBStr": null
  },
  {
    "id": "b77d9132-d85c-464b-93c9-6c74b5de2511",
    "ingestionTime": "1490604144786",
    "ingestionTimeFormatted": "2017-03-27 14:12:24.786",
    "sourceFile": "file:///home/bedrock/zps/victor7.txt",
    "filePatternId": 1,
    "triggerFile": "file:///home/bedrock/zps/victor7.done",
    "entity": null,
    "entityIdVersion": "0.0",
    "destFile": null,
    "sourceRowCount": 0,
    "targetRowCount": 0,
    "sourceSize": "543.0MB",
    "sourceSizeValue": "543.0MB",
    "targetSize": "0B",
    "targetSizeValue": null,
    "expirationDate": null,
    "targetTable": null,
    "publishedFlag": null,
    "wfInstanceId": 0,
    "additionalFiles": null,
    "notes": null,
    "md5sum": null,
    "status": "INGESTION FAILED",
    "effectiveDate": null,
    "error": "com.zaloni.bedrock.collection.agent.exceptions.
↪BedrockException: java.io.IOException: java.io.IOException: java.io.
↪IOException: Directory /home/pankaj/ could not be created",
    "subjectArea": null,
    "sourcePlatform": null,
    "sourceSchema": null,
    "format": null,
    "fileName": "victor7.txt",
    "fileExtension": ".txt",
    "controlFileContents": null,
    "lzServerName": "LZ Sewaiyaan",
    "lzDirPath": "/home/bedrock/zps",
    "ingestedAs": "",
    "fileChecksum": "NA",
    "startTime": "1490604144693",
    "startTimeFormatted": "2017-03-27 14:12:24.693",
    "targetGB": "0B",
    "sourceGB": "543.0MB",
    "sourceGBStr": "543.0MB",
    "targetGBStr": null
  }
],
"page": {
  "chunkSize": 20,
  "currentPage": 1,
  "sortBy": "ingestionTime",

```

```

    "sortOrder": "DESC",
    "totalRecords": 8
  }
}

```

- Scenario 2:

If the status is SUCCESS:

```

{
  "responseMessage": "Success",
  "restUri": "/bedrock-app/services/rest/ingestion/ingestions/search",
  "result": [
    {
      "id": "3eba4b05-4a49-4eal-b467-1182e7858190",
      "ingestionTime": "1491207983000",
      "ingestionTimeFormatted": "2017-04-03 13:56:23",
      "sourceFile": "file:///home/bedrock/testJoydip/newfolder/mynewfolder/
↪test12.txt",
      "filePatternId": 3,
      "triggerFile": "file:///home/bedrock/testJoydip/newfolder/mynewfolder/
↪test12.done",
      "entity": null,
      "entityIdVersion": "0.0",
      "destFile": "file:///home/bedrock/testJoydipNew/test12.txt",
      "sourceRowCount": 0,
      "targetRowCount": 0,
      "sourceSize": "0B",
      "sourceSizeValue": "0B",
      "targetSize": "4.0KB",
      "targetSizeValue": null,
      "expirationDate": null,
      "targetTable": null,
      "publishedFlag": null,
      "wfInstanceId": 0,
      "additionalFiles": "file:///home/bedrock/testJoydipNew/test12.add,file:/
↪//home/bedrock/testJoydipNew/test12.meta,file:///home/bedrock/testJoydipNew/
↪test12.ctrl,file:///home/bedrock/testJoydipNew/test12.md5",
      "notes": null,
      "md5sum": null,
      "status": "SUCCESS",
      "effectiveDate": null,
      "error": null,
      "subjectArea": null,
      "sourcePlatform": null,
      "sourceSchema": null,
      "format": null,
      "fileName": "test12.txt",
      "fileExtension": ".txt",
      "controlFileContents": "",
      "lzServerName": "LZ Sewaiyaan",
      "lzDirPath": "/home/bedrock/testJoydip",
      "ingestedAs": "",
      "fileChecksum": "d41d8cd98f00b204e9800998ecf8427e",
      "startTime": "1491207983000",
      "startTimeFormatted": "2017-04-03 13:56:23",
      "targetGB": "4.0KB",
      "sourceGB": "0B",
    }
  ]
}

```

```

        "sourceGBStr": "0B",
        "targetGBStr": null
    },
    {
        "id": "ad848193-7f46-4eac-9a3e-253ed8d4851f",
        "ingestionTime": "1491190890000",
        "ingestionTimeFormatted": "2017-04-03 09:11:30",
        "sourceFile": "file:///home/bedrock/testJoydip/newfolder/newfolder/
↪test2.txt",
        "filePatternId": 3,
        "triggerFile": "file:///home/bedrock/testJoydip/newfolder/newfolder/
↪test2.done",
        "entity": null,
        "entityIdVersion": "0.0",
        "destFile": "file:///home/bedrock/testJoydipNew/test2.txt",
        "sourceRowCount": 0,
        "targetRowCount": 0,
        "sourceSize": "0B",
        "sourceSizeValue": "0B",
        "targetSize": "0B",
        "targetSizeValue": null,
        "expirationDate": null,
        "targetTable": null,
        "publishedFlag": null,
        "wfInstanceId": 0,
        "additionalFiles": "file:///home/bedrock/testJoydipNew/test2.add,file:///
↪/home/bedrock/testJoydipNew/test2.meta,file:///home/bedrock/testJoydipNew/test2.
↪ctrl,file:///home/bedrock/testJoydipNew/test2.md5",
        "notes": null,
        "md5sum": null,
        "status": "SUCCESS",
        "effectiveDate": null,
        "error": null,
        "subjectArea": null,
        "sourcePlatform": null,
        "sourceSchema": null,
        "format": null,
        "fileName": "test2.txt",
        "fileExtension": ".txt",
        "controlFileContents": "",
        "lzServerName": "LZ Sewaiyaan",
        "lzDirPath": "/home/bedrock/testJoydip",
        "ingestedAs": "",
        "fileChecksum": "d41d8cd98f00b204e9800998ecf8427e",
        "startTime": "1491190890000",
        "startTimeFormatted": "2017-04-03 09:11:30",
        "targetGB": "0B",
        "sourceGB": "0B",
        "sourceGBStr": "0B",
        "targetGBStr": null
    }
],
"page": {
    "chunkSize": 20,
    "currentPage": 1,
    "sortBy": "ingestionTime",
    "sortOrder": "DESC",
    "totalRecords": 20
}

```

```
}
}
```

- Scenario 3:

If there are no records that match the searched criteria:

```
no content
```

Response in Case of Failure:

- Scenario 1:

If the *pageProperties* parameter is not included in the request payload:

```
{
  "responseMessage": "Fields ingestionProperties and pageProperties needs to be_
  ↪mentioned in the Payload.",
  "restUri": "/bedrock-app/services/rest/ingestion/ingestions/search",
  "result": null,
  "page": null
}
```

- Scenario 2:

If the date format in the request payload is incorrect:

```
{
  "responseMessage": "Error parsing date from given string ::as",
  "restUri": "/bedrock-app/services/rest/ingestion/ingestions/search",
  "result": null,
  "page": null
}
```

1.4.13 Add an Ingestion History Record

This API allows you to write (save) a record to the ingestion history. It is recommended to add valid entries to the ZDP ingestion history.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/publish/addFileIndex
```

Method:

```
POST
```

Request Payload:

The *id* parameter is a mandatory field and all other parameters can be set as per the requirement.

```
{
  "additionalFiles": "<This is the comma-separated list of additional files.>",
  "triggerFile": "<This is the trigger file location.>",
  "entityIdVersion": "<This is the entity Id and version in the format: 10.1>",
  "targetRowCount": "<This is currently a placeholder for future use. This_
  ↪should be 0.>",
  "targetGB": "<This is the size of the target file.>",
}
```

```

    "expirationDate": "<This is currently a placeholder for future use. This_
↪should be null.>",
    "sourceGB": "<This is the size of the source file.>",
    "entity": "<This is the business name of the corresponding entity type.>",
    "ingestionTime": "<This is the timestamp of the ingestion in milliseconds.>",
    "wfInstanceId": "<This is the post ingestion workflow instance Id.>",
    "publishedFlag": "<This is currently a placeholder for future use. This_
↪should be null.>",
    "targetTable": "<This is currently a placeholder for future use. This should_
↪be null.>",
    *id": "<This is the watermark Id.>",
    "filePatternId": "<This is the file pattern Id.>",
    "sourceFile": "<This is the location of the source file.>",
    "destFile": "<This is the location of the destination file.>",
    "sourceRowCount": "<This is currently a placeholder for future use. This_
↪should be 0.>",
    "status": "<This is the status of the ingestion: SUCCESS, WORKFLOW FAILED,
↪INCOMPLETE, or INGESTION FAILED.>",
    "md5sum": "<This is the MD5 checksum of the source file.>",
    "effectiveDate": "<This is the effective date.>",
    "notes": "<This is the additional note that can be added to describe the_
↪record.>",
    "error": "<This is the error that occurred in case of ingestion failure.>",
    "format": "<This is the data file format.>",
    "fileName": "<This is the file name of the data file.>",
    "fileExtension": "<This is the file suffix of the data file.>",
    "sourcePlatform": "<This is the source platform of the entity type.>",
    "sourceSchema": "<This is the source schema of the entity type.>",
    "sourceGBStr": "<This is the GB representation of source file size.>",
    "controlFileContents": "<This is the control file contents in JSON format.>"
    "ingestedAs": "<This is the user who performs ingestion.>",
    "fileChecksum": "<This is the suffix of the file whose MD5 value is compared_
↪with the user-defined MD5 value.>"
}

```

Note: The *id* field is unique. If the *id* that is passed already exists in a record, that record is updated; otherwise, a new record is created. If you want to update the values for certain fields, you must pass the values in the input payload. If you do not pass any values, the old values are retained.

Example Request:

```

{
  "id": "c12549aa-ab76-49b0-bf1e-75f96714aaa2",
  "status": "SUCCESS",
  "error": "None",
  "format": "DELIMITED",
  "entity": "DH_TM_FIXED",
  "fileName": " QA_1_TOKEN_FX.DAT",
  "fileExtension": ".dat",
  "sourcePlatform": "Oracle",
  "sourceSchema": "Sales",
  "filePatternId": 13355,
  "entityIdVersion": "1621.1",
  "md5sum": "QATest_c12549aa-ab76-49b0-bf1e-75f96714aaa2",
  "effectiveDate": "2014-04-29 10:16:31",
  "subjectArea": "",

```

```
"wfInstanceId": 67067,
"ingestionTime": "1398780973547",
"sourceFile": "/home/bedrock/SB/LZ/QA_1_TOKEN_FX.DAT",
"triggerFile": "/home/bedrock/SB/LZ/QA_1_TOKEN_FX.meta",
"destFile": "hdfs://gphd-node1.zaloni.net:8020/user/brdev/SB/QA_1_TOKEN_FX.DAT",
"sourceRowCount": 0,
"targetRowCount": 0,
"sourceGB": "19500",
"targetGB": "19500",
"expirationDate": null,
"targetTable": null,
"publishedFlag": null,
"additionalFiles": null,
"notes": null,
"sourceGBStr": null
  "ingestedAs": "Dibya",
"fileChecksum": "d41d8cd98f00b204e9800998ecf8427e"
}
```

Example Response:

```
{
  "result": "Record saved successfully",
  "status": {
    "responseCode": 200,
    "result": null,
    "responseMessage": "Success",
    "responseType": "info"
  }
}
```

1.4.14 Delete an Ingestion History Record

This API allows you to delete a record from the ingestion history.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/publish/
↪deleteFileIndex/{inventory_Id}
```

Note: You must pass the *inventory_Id* parameter in the API URL.

Method:

```
GET
```

Example Response:

```
{
  "result": "Record deleted successfully",
  "status": {
    "responseCode": 200,
    "result": null,
    "responseMessage": "Success",
    "responseType": "info"
  }
}
```

```
}
}
```

Response in Case of Failure:

If inventory Id is invalid/non-existing:

```
{
  "result": "No record found with id 123457888",
  "status": {
    "responseCode": 200,
    "result": null,
    "responseMessage": "Success",
    "responseType": "info"
  }
}
```

1.4.15 Create or Update a File Pattern

This API allows you to add or update a file pattern within a project.

Methods:

- To create file pattern:

```
POST
```

- To update file pattern:

```
PUT
```

Request Payload:

Important: The request payload for the update scenario is same as the create scenario. However, if you do not send any specific parameter in the payload, the system removes the existing information for that specific parameter.

```
{
  *"filePatternId": "<This is the Id of the file pattern. While creating a new pattern,
  ↳ it should be set to 0.>",
  *"patternPrefix": "<This is the file pattern prefix i.e. the regular expression to
  ↳ match the file name.>",
  *"patternSuffix": "<This is the file pattern suffix i.e. the file extension
  ↳ beginning with a dot (.)>",
  "description": "<This is the description of the file pattern.>",
  "byPassIngestion": "<This indicates whether to skip the ingestion process. Possible
  ↳ values are true and false.>",
  *"destination": "<This is the location where the files are ingested. Supports the
  ↳ BEDROCK-level namespace.>",
  "retryAttempts": "<This is the retry attempts. Deprecated and will be removed in
  ↳ the future versions of ZDP. Use 0.>",
  "triggerFileSuffix": "<This is the extension of the trigger file beginning with a
  ↳ dot (.)>",
  "controlFileSuffix": "<This is the extension of the control file beginning with a (
  ↳ .)>",
  "metaFileSuffix": "<This is the extension of the meta file beginning with a (.)>",
```

```

    "deleteAfterIngestion": "<This indicates whether to retain the files in archive_
    ↪directory after ingestion is completed. Possible values are: true and false.>",
    "checksumFileSuffix": "<This is the extension of the checksum file beginning with a_
    ↪(.)>",
    "workflowId": "<This is the Id of the post ingestion workflow.>",
    "workflowName": "<This is the name of the post ingestion workflow.>",
    "globalParameter": "<This is the comma-separated global parameters of the post_
    ↪ingestion workflow.>",
    "wfLevelParamList": "<This is the workflow-level parameters of the post ingestion_
    ↪workflow. See the examples below for more details.>",
    "modifiedTime": "<This is the time when the file pattern was modified.>",
    "createdTime": "<This is the time when the file pattern was created.>",
    "entityId": "<This is the Id of the entity that is associated with the file pattern.
    ↪>",
    "entityVersion": "<This is the version of the entity that is associated with the_
    ↪file pattern.>",
    "entityName": "<This is the name of the entity that is associated with the file_
    ↪pattern.>",
    "frequency": "<This is the frequency value. This must be a number.>",
    "frequencyUnit": "<This is the unit for frequency.>",
    "dataFileFormatId": "<The file format Ids are: 1 for DELIMITED, 2 for FIXED_LENGTH,
    ↪3 for HL7, 4 for X12, 6 for JSON, 7 for CUSTOM, 8 for AVRO, 9 for ORC, 10 for_
    ↪PARQUET, 11 for RC, 12 for SEQUENCE. This parameter is not required if the_
    ↪dataFileFormat parameter is defined.>",
    "dataFormat": "<The file formats are: DELIMITED, FIXED_LENGTH, HL7, X12, XML, JSON,
    ↪CUSTOM, AVRO, ORC, PARQUET, RC, SEQUENCE. This parameter is not required if the_
    ↪dataFileFormatId parameter is defined.>",
    "delimiter": "<This is the delimiter for the file pattern. This parameter is_
    ↪required if the set dataFileFormatId parameter is DELIMITED. If the delimiter is_
    ↪non-printable, provide the corresponding unicode value. For example, the unicode_
    ↪value for Ctrl+A delimiter corresponds to \\u0001 (\\ to escape).>",
    "otherFiles": "<This is the other types of files associated with the file pattern.>
    ↪",
    "lzDirectories": "<This is an array of source directories with which the pattern is_
    ↪associated. Refer to the below examples for more information.>",
    "filePatternAttributes": "<This is an array of custom attributes associated with_
    ↪the pattern. Refer to the below examples for more information.>",
    "flAgents": "<This indicates the Flume agents that must be linked with the current_
    ↪file pattern.>",
    "wfNamespaceList": "<This is an array of workflow namespace variables objects to be_
    ↪used for the post ingestion workflow.>",
    "scriptLocation": "<This is the pre-ingestion script location.>",
    "scriptTimeout": "<This is the script execution timeout value in seconds.>",
    "adminCapacityQueues": { "<If no queue is configured, set this value as null. If_
    ↪configured, add the following key and set the queue name.>"
        "queueName": "<This is the queue name (if any).>"
    },
    "fileSystemId": "<This is the file system where the landing zone directory is_
    ↪located. Use 1 for LOCAL, 2 for HDFS, 3 for S3, 4 for AZURE. This value is non-
    ↪editable. If not provided, it defaults to 1.>",
    "fileName": "<This is the name of the file system.>",
    "destinationFileSystemUri": "<This is the filesystem URI for the destination_
    ↪storage system. The file system URI comprises of the form scheme://authority.>",
    "destinationFileSystemProperties": { "<Set the properties for destination file_
    ↪system (if any) as part of Key:Value pair.>"
        "key_name": "<value>"
    },
    "fileSystemConnectionInstanceId": "<This is the Id of the destination connection_
    ↪URL. This value can be retrieved by using ZDP connection APIs.>",

```



```

    "fileSystemConnectionInstanceName": "<This is the file system connection instance_
    ↳name. This value can be retrieved by using ZDP connection APIs.>",
    "batchedIngestionFlag": "<Set it to True to enable batched mode of ingestion._
    ↳Default is False.>",
    "ingestionBatchSize": "<This is a configuration property that sets the batch size_
    ↳when the property.>",
    "ingestionBatchWindow": "<This is the value that sets the window size when batched_
    ↳ingestion is enabled.>",
    "bedrockNamespaceMap": {},
    "preIngestionScriptArgs": "<These are the values that can be passed as arguments_
    ↳for the pre-ingestion scripts (if defined). The default or inbuilt arguments that_
    ↳ZDP provides are: ${INPUT_FILE}, ${ADDITIONAL_FILES}, ${FILE_PATTERN_ID}, ${ENTITY_
    ↳ID}, ${ENTITY_VERSION}, ${LZ_DIRECTORY}. Namespace is also supported.>",
    "userId": "<This is the name of the user who has created the file pattern.>",
    "ingestAs": "<This is the user owning the file pattern. Applicable when the_
    ↳impersonation feature is enabled.>"
  }

```

Example Request:

```

{
  "patternPrefix": "Project_FilePattern74[0-9]+",
  "patternSuffix": ".dat",
  "otherFiles": "",
  "description": "",
  "destination": "/user/bedrock/GlobalDest42",
  "destinationFileSystemUri": "hdfs://hdp-node1.zalonilabs.com:8020",
  "retryAttempts": "0",
  "triggerFileSuffix": ".done",
  "deleteAfterIngestion": "false",
  "globalParameter": "",
  "lzDirectories": [ {
    "lzDirId": "3"
  }
],
  "frequency": "1",
  "frequencyUnit": "",
  "wfLevelParamList": "",
  "delimiter": "",
  "entityId": "",
  "entityVersion": "",
  "destinationFileSystemProperties": {
    "": ""
  },
  "workflowId": "",
  "dataFileFormatId": "",
  "dataFormat": "",
  "filePatternId": 0,
  "adminCapacityQueues": {},
  "controlFileSuffix": ".ctl",
  "metaFileSuffix": ".meta",
  "fileSystemConnectionInstanceId": "9",
  "fileSystemId": "1",
  "ingestAs": "bedrock",
  "checksumFileSuffix": ".md5"
}

```

Example Response:

```
{
  "responseMessage": "File Pattern saved successfully",
  "restUri": null,
  "result": "47",
  "page": null
}
```

Note: The Id of the newly added pattern is returned as the value of the *result* property.

Response in Case of Failure:

```
{
  "responseMessage": "Error while checking project permissions",
  "restUri": "/bedrock-app/services/rest/projects/2/filePatterns",
  "result": null,
  "page": null
}
```

1.4.15.1 Create a file pattern

This API allows you to add a file pattern to a specific project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/{project_Id}/
↪filePatterns
```

Note: You must pass the *project_Id* parameter in the API URL. To create a file pattern under the public project, set the *project_Id* parameter to *1*.

1.4.15.2 Update a file pattern

This API allows you to update a file pattern in a project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/filePatterns/{file_pattern_Id}?
↪projectId={project_Id}
```

Note:

- You must pass the *project_Id* and *file_pattern_Id* parameters in the API URL. To update a file pattern under the public project, set the *project_Id* parameter to *1*.
- The request payload for the update scenario is same as the create scenario. However, if you do not send any specific parameter in the payload, the system removes the existing information for that specific parameter.

1.4.16 Search for File Patterns

This API allows you to search for file patterns by using the pattern prefix. This service returns only the basic information of the patterns (identical to the file pattern search feature in the ZDP UI).

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/filePatterns/search?projectIds=
↪{project_Id}&patternPrefix={pattern_prefix}
```

Note: You must pass the *project_Id* and *pattern_prefix* parameters in the API URL. To search for file pattern under the public project, set the *project_Id* parameter to *1*.

Method:

```
GET
```

Example Response:

```
{
  "responseMessage": "SUCCESS",
  "restUri": null,
  "result": {
    "data": [
      {
        "patternPrefix": ".dat",
        "destinationFileSystemUri": null,
        "createdTime": null,
        "fileSystemId": 1,
        "triggerFileSuffix": ".done",
        "fileSystemName": "LOCAL",
        "fileSystemConnectionInstanceName": null,
        "flAgents": null,
        "filePatternId": "46",
        "ingestionBatchSize": 0,
        "ingestionBatchWindow": 0,
        "userId": null,
        "preIngestionScriptArgs": null,
        "batchedIngestionFlag": false,
        "patternSuffix": ".txt",
        "destinationFileSystemProperties": null,
        "ingestAs": "SidTest",
        "bedrockNamespaceMap": {},
        "fileSystemConnectionInstanceId": 0
      }
    ],
    "totalRecords": 1,
    "currentPage": 0,
    "chunkSize": 0
  },
  "page": null
}
```

Response in Case of Failure:

If no matching records are found:

```
no content
```

1.4.17 Fetch the Details of a File Pattern

This API allows you to fetch the details of a file pattern.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/filePatterns/{file_pattern_Id}?
↳projectId={project_Id}
```

Note: You must pass the *project_Id* and *file_pattern_Id* parameters in the API URL. To fetch the details of a file pattern under the public project, set the *project_Id* parameter to *1*.

Method:

```
GET
```

Example Response:

```
{
  "responseMessage": "SUCCESS",
  "restUri": null,
  "result": {
    "filePatternId": "47",
    "patternPrefix": "Project_FilePattern75[0-9]+",
    "patternSuffix": ".dat",
    "description": "",
    "byPassIngestion": false,
    "destination": "/user/bedrock/GlobalDest42",
    "retryAttempts": "0",
    "triggerFileSuffix": ".done",
    "controlFileSuffix": ".ctl",
    "metaFileSuffix": ".meta",
    "deleteAfterIngestion": false,
    "checksumFileSuffix": ".md5",
    "workflowId": null,
    "workflowName": null,
    "globalParameter": null,
    "wfLevelParamList": null,
    "modifiedTime": "06/20/2016 19:15:20",
    "createdTime": "06/20/2016 19:15:20",
    "entityId": null,
    "entityVersion": null,
    "entityName": null,
    "frequency": "1",
    "frequencyUnit": "",
    "dataFileFormatId": null,
    "dataFormat": null,
    "delimiter": "",
    "otherFiles": "",
    "lzDirectories": [
      {
        "lzDirId": 3,
        "dirPath": "/home/bedrock/rbacIngestLZ",

```

```

        "description": "through automation",
        "totalFPCount": 0,
        "totalSlaves": 2,
        "serverId": 2,
        "serverName": "RBAC Server",
        "fileSystemId": 1,
        "fileSystemName": "LOCAL",
        "notificationDetails": null,
        "fileSystemUri": "file:///",
        "fileSystemConnectionInstanceId": 8,
        "fileSystemConnectionInstanceName": "LocalConnection",
        "allowIngestion": true,
        "modifiedTime": 1466418449000
    }
],
"filePatternAttributes": [],
"flAgents": [],
"wfNamespaceList": [],
"scriptLocation": null,
"scriptTimeout": 0,
"adminCapacityQueues": null,
"fileSystemId": 1,
"fileSystemName": "LOCAL",
"destinationFileSystemUri": "hdfs://hdp-nodel.zalonilabs.com:8020",
"destinationFileSystemProperties": {
    "": ""
},
"fileSystemConnectionInstanceId": 9,
"fileSystemConnectionInstanceName": "HDFSConnection",
"batchedIngestionFlag": false,
"ingestionBatchSize": 0,
"ingestionBatchWindow": 0,
"bedrockNamespaceMap": {},
"preIngestionScriptArgs": null,
"userId": null,
"ingestAs": "bedrock"
},
"page": null
}

```

Response in Case of Failure:

If you do not have access to the file pattern:

```

{
  "responseMessage": "Access Denied due to missing associations of following artifact_
  ↪identifiers - 48",
  "restUri": "/bedrock-app/services/rest/filePatterns/48",
  "result": null,
  "page": null
}

```

1.4.18 Fetch Date Histogram for Ingestion History

This API allows you to view the date histogram for ingestion history records for ZDP entities. By using this API, you can:

- Fetch the total number files ingested per day/month in a certain date range for a specific entity.
- Fetch the total file size on a per day/month basis.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/aggregates
```

Method:

```
POST
```

Request Payload:

```
{
  "aggregateType": "<This is the type of aggregate. This supports only_
  ↳dateHistogram. >",
  "entityList": [
    "<This is the list of entities for which the_
    ↳aggregates are retrieved. An empty list indicates all the entities. To set multiple_
    ↳entities, use the format:
    ↳
    ↳      <EntityId_1.versionId>,
    ↳      <EntityId_2.versionId>>"
    ],
  "intervalType": "<This is an optional property. The histogram buckets are_
  ↳ fetched based on the set interval type. The probable values can be DAY or YEAR. The_
  ↳ default value is DAY.>",
  "status": [
    "<This is the status of the process. This value can be_
    ↳ SUCCESS.>"
    ],
  "dateRangeStart": "<This is the start date and time in the format: <yyyy-MM-dd>
  ↳ T<HH:mm:ss>Z. If you pass this, dateRangeEnd also must be passed.>",
  "dateRangeEnd": "<This is the end date and time in the format: <yyyy-MM-dd>T
  ↳ <HH:mm:ss>Z.>"
}
```

Example Request:

```
{
  "aggregateType": "dateHistogram",
  "entityList": [],
  "intervalType": "DAY",
  "status": ["SUCCESS"],
  "dateRangeStart": "2017-03-26T00:00:00.235-07:00",
  "dateRangeEnd": "2017-04-4T00:00:00.235-07:00"
}
```

Example Response:

```
{
  "responseMessage": "Success",
  "restUri": "/bedrock-app/services/rest/ingestion/aggregates",
  "result": {
    "records": [
      {
        "label": "2017-03-27T00:00:00.000+05:30",
        "fileCount": 3,
        "fileSize": 25
      }
    ]
  }
}
```

```

    },
    {
      "label": "2017-03-28T00:00:00.000+05:30",
      "fileCount": 0,
      "fileSize": 0
    },
    {
      "label": "2017-03-29T00:00:00.000+05:30",
      "fileCount": 0,
      "fileSize": 0
    },
    {
      "label": "2017-03-30T00:00:00.000+05:30",
      "fileCount": 0,
      "fileSize": 0
    },
    {
      "label": "2017-03-31T00:00:00.000+05:30",
      "fileCount": 1,
      "fileSize": 0
    },
    {
      "label": "2017-04-01T00:00:00.000+05:30",
      "fileCount": 0,
      "fileSize": 0
    },
    {
      "label": "2017-04-02T00:00:00.000+05:30",
      "fileCount": 0,
      "fileSize": 0
    },
    {
      "label": "2017-04-03T00:00:00.000+05:30",
      "fileCount": 16,
      "fileSize": 0
    }
  ],
  "totalFileSize": "25 B",
  "totalFileCount": 20
},
"page": null
}

```

Response in Case of Failure:

• Scenario 1:

If you specify an invalid entity (connectionTypeId or versions less than 1):

```

{
  "responseMessage": "Invalid entity id.version foramt <0.0>",
  "restUri": "/bedrock-app/services/rest/ingestion/aggregates",
  "result": null,
  "page": null
}

```

• Scenario 2:

If you specify invalid interval type:

```
{
  "responseMessage": "intervalType supports DAY and MONTH only",
  "restUri": "/bedrock-app/services/rest/ingestion/aggregates",
  "result": null,
  "page": null
}
```

- Scenario 3:

If you specify invalid dates:

```
{
  "responseMessage": "Error in parsing dates",
  "restUri": "/bedrock-app/services/rest/ingestion/aggregates",
  "result": null,
  "page": null
}
```

- Scenario 4:

If the aggregation type is not implemented:

```
{
  "status": null,
  "result": "Unimplemented aggregation type"
}
```

1.5 Prepare API

The prepare APIs allow you to perform various operations on workflows and schedules, such as execute specific workflow, set custom status for a workflow or a step in a workflow, export/import workflow(s), resume/suspend workflow schedules, etc.

1.5.1 Search and Fetch Workflows

This API allows you to fetch the list of workflows that belong to a specific project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/search?projectIds=
→ {project_Id}
```

Note: You must pass the *project_Id* parameter in the API URL. To fetch the workflows under the public project, set the *project_Id* parameter to 1.

Method:

```
POST
```

Request Payload:

You can pass anyone of the following parameters to fetch the desired output:


```
{
    "chunkSize": "<This is the number of records required to be fetched. Set a_
↪value to get the records corresponding to the specified value. When set as '0' (or_
↪unspecified), all results are fetched.>",
    "currentPage": "<This is the current page required for paginating the request_
↪depending on the total records and chunkSize.>",
    "wfName": "<This is the name of the workflow.>",
    "sortBy": "<This is the field name over which the response result is sorted._
↪The default sorting is on the modified date so that the latest workflow execution_
↪instance comes on the top.>",
    "desc": "<This is the parameter to indicate if you want to fetch the_
↪description. Set it to true or false accordingly.>"
}
```

Example Request:

To fetch the top 20 workflows based on the *modifiedDate*:

```
{
    "chunkSize": "20",
    "currentPage": 1,
    "wfName": "",
    "sortBy": "modifiedDate",
    "desc": true
}
```

Example Response:

If single (or comma-separated) *project_Id* is provided in the request URL:

```
{
    "responseMessage": "success",
    "restUri": null,
    "result": {
        "currentPage": 1,
        "totalRecords": 8,
        "workFlowDetails": [
            {
                "CSVValues": "7,FDQ002,unassigned,admin,01/24/2017_
↪13:41:54,admin,01/24/2017 13:41:54,null,null",
                "category": "unassigned",
                "createdBy": "admin",
                "createdDate": "01/24/2017 13:41:54",
                "lastExecutionStatus": null,
                "lastExecutionTime": null,
                "modifiedBy": "admin",
                "modifiedDate": "01/24/2017 13:41:54",
                "ownerProjectId": 1,
                "ownerProjectName": "Public Project",
                "shared": false,
                "sharedWith": 0,
                "wfId": 7,
                "wfName": "FDQ002"
            },
            {
                "CSVValues": "8,Lin_Ent_Del_WF,unassigned,admin,01/24/
↪2017 13:42:56,admin,01/24/2017 13:43:02,SUCCESS,01/24/2017 13:43:29",
                "category": "unassigned",
                "createdBy": "admin",
```

```

        "createdDate": "01/24/2017 13:42:56",
        "lastExecutionStatus": "SUCCESS",
        "lastExecutionTime": "01/24/2017 13:43:29",
        "modifiedBy": "admin",
        "modifiedDate": "01/24/2017 13:43:02",
        "ownerProjectId": 1,
        "ownerProjectName": "Public Project",
        "shared": false,
        "sharedWith": 0,
        "wfId": 8,
        "wfName": "Lin_Ent_Del_WF"
    },
    {
        "CSVValues": "10,AVRO_After_PArquet,unassigned,admin,
↪01/24/2017 13:46:01,admin,01/24/2017 13:47:33,SUCCESS,01/24/2017 13:48:09",
        "category": "unassigned",
        "createdBy": "admin",
        "createdDate": "01/24/2017 13:46:01",
        "lastExecutionStatus": "SUCCESS",
        "lastExecutionTime": "01/24/2017 13:48:09",
        "modifiedBy": "admin",
        "modifiedDate": "01/24/2017 13:47:33",
        "ownerProjectId": 1,
        "ownerProjectName": "Public Project",
        "shared": false,
        "sharedWith": 0,
        "wfId": 10,
        "wfName": "AVRO_After_PArquet"
    },
    {
        "CSVValues": "9,SQ0op_imp,unassigned,admin,01/24/2017_
↪13:43:13,admin,01/24/2017 13:54:05,SUCCESS,01/24/2017 13:54:09",
        "category": "unassigned",
        "createdBy": "admin",
        "createdDate": "01/24/2017 13:43:13",
        "lastExecutionStatus": "SUCCESS",
        "lastExecutionTime": "01/24/2017 13:54:09",
        "modifiedBy": "admin",
        "modifiedDate": "01/24/2017 13:54:05",
        "ownerProjectId": 1,
        "ownerProjectName": "Public Project",
        "shared": false,
        "sharedWith": 0,
        "wfId": 9,
        "wfName": "SQ0op_imp"
    },
    {
        "CSVValues": "13,sqoop_test,unassigned,admin,01/24/
↪2017 14:09:32,admin,01/24/2017 14:09:32,null,null",
        "category": "unassigned",
        "createdBy": "admin",
        "createdDate": "01/24/2017 14:09:32",
        "lastExecutionStatus": null,
        "lastExecutionTime": null,
        "modifiedBy": "admin",
        "modifiedDate": "01/24/2017 14:09:32",
        "ownerProjectId": 1,
        "ownerProjectName": "Public Project",
    }

```

```

        "shared": false,
        "sharedWith": 0,
        "wfId": 13,
        "wfName": "sqoop_test"
    },
    {
        "CSVValues": "20,Hive_Test999,unassigned,admin,01/24/
↪2017 14:27:22,admin,01/24/2017 14:27:22,null,null",
        "category": "unassigned",
        "createdBy": "admin",
        "createdDate": "01/24/2017 14:27:22",
        "lastExecutionStatus": null,
        "lastExecutionTime": null,
        "modifiedBy": "admin",
        "modifiedDate": "01/24/2017 14:27:22",
        "ownerProjectId": 1,
        "ownerProjectName": "Public Project",
        "shared": false,
        "sharedWith": 0,
        "wfId": 20,
        "wfName": "Hive_Test999"
    },
    {
        "CSVValues": "22,Hive_Test99,unassigned,admin,01/24/
↪2017 14:30:20,admin,01/24/2017 14:30:20,null,null",
        "category": "unassigned",
        "createdBy": "admin",
        "createdDate": "01/24/2017 14:30:20",
        "lastExecutionStatus": null,
        "lastExecutionTime": null,
        "modifiedBy": "admin",
        "modifiedDate": "01/24/2017 14:30:20",
        "ownerProjectId": 1,
        "ownerProjectName": "Public Project",
        "shared": false,
        "sharedWith": 0,
        "wfId": 22,
        "wfName": "Hive_Test99"
    },
    {
        "CSVValues": "24,Hive_Test9999,unassigned,admin,01/24/
↪2017 14:30:48,admin,01/24/2017 14:30:48,null,null",
        "category": "unassigned",
        "createdBy": "admin",
        "createdDate": "01/24/2017 14:30:48",
        "lastExecutionStatus": null,
        "lastExecutionTime": null,
        "modifiedBy": "admin",
        "modifiedDate": "01/24/2017 14:30:48",
        "ownerProjectId": 6,
        "ownerProjectName": "p4",
        "shared": false,
        "sharedWith": 0,
        "wfId": 24,
        "wfName": "Hive_Test9999"
    }
}
],

```

```
}
  "page": null
}
```

Response in Case of Failure:

- Scenario 1:

If no workflow exists:

```
no content
```

- Scenario 2:

If the logged in user does not have permission to the specified project (in the URL):

```
{
  "responseMessage": "User john doesn't have access to some of these_
↳permissions - wf_view_workflows, ingestion_manage_file_patterns",
  "restUri": "/bedrock-app/services/rest/workflows/search",
  "result": null,
  "page": null
}
```

1.5.2 Fetch Workflow Instance Details

This API allows you to fetch the details of a workflow instance within a project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/{workflow_Id}?
↳projectId={project_Id}&instanceId={instance_Id}&version={instance_version}
```

Note: You must pass the value for the *workflow_Id*, *project_Id*, *instance_Id*, and *instance_version* parameters in the API URL. To fetch the instance details for a workflow under the public project, set the *project_Id* parameter to *1*.

Method:

```
GET
```

Example Response:

```
{
  "responseMessage": "success",
  "restUri": "/bedrock-app/services/rest/workflows/5",
  "result": {
    "wfId": 15,
    "categoryId": 1,
    "wfName": "pique_Goal_Scores",
    "createdBy": "admin",
    "createdDate": "08/01/2017 16:49:58",
    "modifiedBy": "admin",
    "modifiedDate": "08/01/2017 20:23:10",
    "wfSeverity": "",
    "wfNotification": "N",
    "wfStepNotification": "N",
    "notificationDetails": [],
  }
}
```

```

    "globalParameter": ",,,,,,,,",
    "stepList": [
      {
        "stepId": 179,
        "actionId": 1,
        "stepName": "Start",
        "shapeCoordinates": "217,157",
        "stepSeverity": "",
        "stepNotification": "N",
        "yesStepId": 0,
        "yesStepName": null,
        "noStepId": 0,
        "noStepName": null,
        "pairConnectionId": 0,
        "pairConnectionName": null,
        "stepParamList": [],
        "stepConnectionList": [
          {
            "stepConnectionId": 122,
            "nextId": 181,
            "nextStepName": "dq_s1"
          }
        ],
        "isAlreadyExecuted": "",
        "stepSubList": null,
        "stepType": "Start",
        "subWorkflowId": 0,
        "subWorkFlow": null,
        "description": null,
        "skipped": false
      },
      {
        "stepId": 181,
        "actionId": 18,
        "stepName": "dq_s1",
        "shapeCoordinates": "333,144",
        "stepSeverity": "",
        "stepNotification": "N",
        "yesStepId": 0,
        "yesStepName": null,
        "noStepId": 0,
        "noStepName": null,
        "pairConnectionId": 0,
        "pairConnectionName": null,
        "stepParamList": [
          {
            "stepParamId": 788,
            "key": "invalidDataPath",
            "valueText": "/user/bedrock/praj/dq/
→report"

          },
          {
            "stepParamId": 789,
            "key": "entityTypeAndVersion",
            "valueText": "dq_par_test(25.1)"
          },
          {
            "stepParamId": 790,

```

```

↪":\\"p1\\",\\"value\\":\\"\\"]"
    },
    {
        "stepParamId": 791,
        "key": "description",
        "valueText": null
    },
    {
        "stepParamId": 792,
        "key": "createGoodDataEntity",
        "valueText": "false"
    },
    {
        "stepParamId": 793,
        "key": "inputFromEntity",
        "valueText": "false"
    },
    {
        "stepParamId": 794,
        "key": "badDataOPLocation",
        "valueText": "/user/bedrock/praj/dq/

↪bad"

    },
    {
        "stepParamId": 795,
        "key": "createBadDataEntity",
        "valueText": "false"
    },
    {
        "stepParamId": 796,
        "key": "dataFilePath",
        "valueText": "/user/bedrock/praj/dq/

↪input"

    },
    {
        "stepParamId": 797,
        "key": "proceedOnFailure",
        "valueText": "false"
    },
    {
        "stepParamId": 798,
        "key": "esReportRequired",
        "valueText": "true"
    },
    {
        "stepParamId": 799,
        "key": "fieldDelimiterDQ",
        "valueText": null
    },
    {
        "stepParamId": 800,
        "key": "validDataPath",
        "valueText": "/user/bedrock/praj/dq/

↪good"

    }
],

```

```

        "stepConnectionList": [
            {
                "stepConnectionId": 123,
                "nextId": 180,
                "nextStepName": "Stop"
            }
        ],
        "isAlreadyExecuted": "",
        "stepSubList": null,
        "stepType": "Data Quality Action",
        "subWorkflowId": 0,
        "subWorkFlow": null,
        "description": null,
        "skipped": false
    },
    {
        "stepId": 180,
        "actionId": 5,
        "stepName": "Stop",
        "shapeCoordinates": "444,150",
        "stepSeverity": "",
        "stepNotification": "N",
        "yesStepId": 0,
        "yesStepName": null,
        "noStepId": 0,
        "noStepName": null,
        "pairConnectionId": 0,
        "pairConnectionName": null,
        "stepParamList": [],
        "stepConnectionList": [],
        "isAlreadyExecuted": "",
        "stepSubList": null,
        "stepType": "Stop",
        "subWorkflowId": 0,
        "subWorkFlow": null,
        "description": null,
        "skipped": false
    }
],
"bedrockInstanceId": 55,
"executedBy": "admin",
"scheduledBy": null,
"wfLevelParameterList": [],
"wfType": "GEN",
"listCounterMap": {
    "totalSteps": "1"
},
"instanceVersion": 1,
"categoryName": "unassigned",
"bedrockURL": "http://192.168.1.39:9090/",
"wfNamespaceList": [
    {
        "id": 0,
        "key": "PROJECT_ID",
        "value": "1",
        "overridable": false
    }
],

```

```
{
  "description": "",
  "adminCapacityQueues": {
    "queueId": 0,
    "queueName": "default",
    "queueCapacity": null,
    "queueElasticity": 0,
    "parentQueueId": 0,
    "secUser": null
  },
  "logLevel": "DEBUG",
  "baseEncodedString": null,
  "overwritable": false,
  "clusterId": "1_39",
  "executionPriority": "NORMAL",
  "stepRestart": false
},
"page": null
}
```

Response in Case of Failure:

If no records are found for the request:

```
{
  "responseMessage": "NO_RECORD_FOUND",
  "restUri": "/bedrock-app/services/rest/workflows/instances/46/versions/2",
  "result": null,
  "page": null
}
```

1.5.3 Delete Workflow

This API allows you to delete multiple workflow(s).

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/bulk
```

Method:

```
DELETE
```

Input Parameters:

```
*[workflow_instance_Id1, workflow_instance_Id2,...]
```

Example Request:

```
[1,2]
```

Example Response:

- Scenario 1:

If multiple workflows are deleted:


```
{
  "responseMessage": "2 workflow(s) deleted successfully.",
  "restUri": "/bedrock-app/services/rest/workflows/bulk",
  "result": "2 workflow(s) deleted successfully.",
  "page": null
}
```

- Scenario 2:

If a single workflow is deleted:

```
{
  "responseMessage": " Workflow with Id: 6 deleted successfully",
  "restUri": "/bedrock-app/services/rest/workflows/bulk",
  "result": " Workflow with Id: 6 deleted successfully",
  "page": null
}
```

Response in Case of Failure:

- Scenario 1:

If you enter non-existent workflow_instance Ids:

```
{
  "responseMessage": "Missing resource associations for following artifact_
↳identifiers - 11,1001",
  "restUri": "/bedrock-app/services/rest/workflows/bulk",
  "result": null,
  "page": null
}
```

- Scenario 2:

If you enter a single workflow Id, such that this workflow is used as a sub-workflow in another workflow:

```
{
  "responseMessage": " Workflow 5 is already used as a sub workflow.",
  "restUri": "/bedrock-app/services/rest/workflows/bulk",
  "result": " Workflow 5 is already used as a sub workflow.",
  "page": null
}
```

- Scenario 3:

If you enter multiple workflow_instance Ids where one of the deleted workflow is used as a sub-workflow in another workflow:

```
{
  "responseMessage": "1 workflow(s) deleted successfully.1 workflow(s)_
↳could not be deleted",
  "restUri": "/bedrock-app/services/rest/workflows/bulk",
  "result": "1 workflow(s) deleted successfully.1 workflow(s) could not be_
↳deleted",
  "page": null
}
```

1.5.4 Execute Workflow

This API allows you to execute an existing workflow that belongs to a specific project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/{workflow_Id}/execute?  
↪projectIds={project_Id}
```

Note: You must provide the *project_Id* and *workflow_Id* parameters in the API URL. To execute an workflow under the public project, set the *project_Id* parameter to *1*.

Method:

POST

Request Payload:

```
{  
    *"wfName": "<This is the name of the workflow.>",  
    "globalParameter": "<This is the global parameter, represented as a_  
↪comma-separated list.>",  
    "wfLevelParameterList": "<This is the workflow-level parameter that is_  
↪sent as an array.>",  
    "wfNamespaceList": "<This is the workflow namespace variables that is_  
↪sent as an array.>",  
    "clusterId": "<This is the Id of the cluster where the job is_  
↪submitted. When you do not provide this value, the system uses the default cluster.>  
↪"  
    "executedBy": "<This is the user executing the workflow. When_  
↪undefined, this parameter is set as the logged-in user. When impersonation is_  
↪enabled, set the impersonated (or proxy) user as applicable.>",  
    "adminCapacityQueues": {  
        "queueName": "<This is the capacity queue-name (if any). If not, set_  
↪null as set in Example Input 2.>"}  
    "executionPriority": "<This is the priority of execution of the_  
↪workflow. This can be HIGH, NORMAL, or LOW. If you do not define this parameter,  
↪the system sets the value to NORMAL.>"  
}
```

Example Request:

- Scenario 1:

By using a capacity queue:

```
{  
    "wfName": "Data_Refine",  
    "globalParameter": "ex_pass,,,,,,,,",  
    "executedBy": "admin",  
    "wfLevelParameterList": [],  
    "wfNamespaceList": [],  
    "adminCapacityQueues": {  
        "queueName": "default"  
    },  
    "clusterId": "11_11_14_54"  
}
```

- Scenario 2:

Without using a capacity queue:

```
{
  "wfName": "Core_Data_Process",
  "globalParameter": "pass_in,,,,,,,,",
  "executedBy": "admin",
  "wfLevelParameterList": [],
  "wfNamespaceList": [],
  "clusterId": "10_1_3_59"
}
```

- Scenario 3:

By impersonating another user (when impersonation is enabled):

```
{
  "wfName": "Finance_YTD",
  "globalParameter": "year_id,,,,,,,,",
  "executedBy": "john_impersonate",
  "wfLevelParameterList": [],
  "wfNamespaceList": [],
  "adminCapacityQueues": {
    "queueName": "default"
  },
  "clusterId": "10_11_12_64"
}
```

Example Response:

```
{
  "responseMessage": "Workflow with instance Id - 259 has been queued. Will be_
↳ running shortly",
  "restUri": "/bedrock-app/services/rest/workflows/1/execute",
  "result": "Workflow with instance Id - 259 has been queued. Will be running shortly
↳ ",
  "page": null
}
```

Response in Case of Failure:

If the workflow name is not given or an invalid workflow name is given:

```
{
  "responseMessage": "Error occurred in {} with exception type: {}.",
  "restUri": "/bedrock-app/services/rest/workflows/24/execute",
  "result": null,
  "page": null
}
```

1.5.5 Stop Workflow

This API allows you to stop a workflow within a project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/{workflow_Id}/  
→instances/{instance_Id}/stop?projectIds={project_Id}
```

Note: You must pass the *workflow_Id*, *instance_Id*, *project_Id* parameters in the API URL. To stop a workflow under the public project, set the value for the *project_Id* parameter to *1*.

Method:

```
PUT
```

Example Response:

- Scenario 1:

If you stop a running workflow:

```
{  
  "responseMessage": "Instance id '70' has been stopped successfully",  
  "restUri": "/bedrock-app/services/rest/workflows/17/instances/70/stop",  
  "result": null,  
  "page": null  
}
```

- Scenario 2:

If you stop a workflow that is already stopped:

```
{  
  "responseMessage": "Workflow instance :70 has been already stopped",  
  "restUri": "/bedrock-app/services/rest/workflows/17/instances/70/stop",  
  "result": null,  
  "page": null  
}
```

Response in Case of Failure:

If the specified workflow does not exist in the project:

```
{  
  "responseMessage": "Artifact with Id 1 and type WORKFLOW cannot be accessed from_  
→project with Id 1",  
  "restUri": "/bedrock-app/services/rest/workflows/1/instances/1/stop",  
  "result": null,  
  "page": null  
}
```

1.5.6 Stop Workflows in Bulk

This API allows you to stop workflows, in bulk.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/instances/bulk/stop
```

Method:

PUT

Input Parameters:

```
*[workflow_instance_Id1, workflow_instance_Id2,...]
```

Example Request:

```
[1, 2]
```

Example Response:

```
{
  "responseMessage": "Bulk stopping of workflow instance(s) completed successfully",
  "restUri": "/bedrock-app/services/rest/workflows/instances/bulk/stop",
  "result": {
    "successIds": [
      361
    ],
    "failedIds": [],
    "failureCause": null
  },
  "page": null
}
```

Response in Case of Failure:

- Scenario 1:

If you enter non-existent *workflow_instance_Id* or if you try to stop completed workflows:

```
{
  "responseMessage": "Failed to bulk stop workflow instance(s)",
  "restUri": "/bedrock-app/services/rest/workflows/instances/bulk/stop",
  "result": {
    "successIds": [],
    "failedIds": [
      361
    ],
    "failureCause": null
  },
  "page": null
}
```

- Scenario 2:

If you specify *workflow_instance_Id* where some of the workflows are already executed:

```
{
  "responseMessage": "Bulk stopping of workflow instance completed with errors",
  "restUri": "/bedrock-app/services/rest/workflows/instances/bulk/stop",
  "result": {
    "successIds": [
      320
    ],
    "failedIds": [
      361
    ],
    "failureCause": null
  }
}
```

```
},  
"page": null  
}
```

1.5.7 Restart Failed Workflows in Bulk

This API allows you to restart the failed workflows, in bulk. Additionally, you can update the log level and execution priority for the specified workflows.

URL:

To restart multiple workflows in multiple projects:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/bulk/restart
```

Method:

POST

Request Payload:

```
{  
  *"instanceIds": "<This is the workflow instance Id(s) that are passed as an array.>",  
  "logLevel": "<This is the log level where you want to store the logs for the_<br>↳ restarted workflow.>",  
  "executionPriority": "<This is the execution priority for the restarted workflow.>"  
}
```

Example Request:

```
{  
  "instanceIds": [26,27],  
  "logLevel": "ERROR",  
  "executionPriority": "HIGH"  
}
```

Example Response:

```
{  
  "responseMessage": "Bulk rerun of workflow instance(s) completed successfully",  
  "restUri": "/bedrock-app/services/rest/workflows/bulk/restart",  
  "result": {  
    "successIds": [  
      43  
    ],  
    "failedIds": [],  
    "failureCause": {}  
  },  
  "page": null  
}
```

Response in Case of Failure:

- Scenario 1:

If some of the specified *workflow_instance_Id* correspond to workflows that have executed successfully:

```
{
  "responseMessage": "Bulk rerun of workflow instance(s) completed successfully_
  ↳with errors",
  "restUri": "/bedrock-app/services/rest/workflows/bulk/restart",
  "result": {
    "successIds": [
      43
    ],
    "failedIds": [
      65
    ],
    "failureCause": {
      "65": "Restarting of workflow instance failed for instance Id due to_
  ↳exception null"
    }
  },
  "page": null
}
```

- Scenario 2:

If you specify *workflow_instance_Id* for workflows that have successfully executed:

```
{
  "responseMessage": "Failed to bulk rerun workflow instance(s)",
  "restUri": "/bedrock-app/services/rest/workflows/bulk/restart",
  "result": {
    "successIds": [],
    "failedIds": [
      68,
      78
    ],
    "failureCause": {
      "68": "Restarting of workflow instance failed for instance Id due to_
  ↳exception Failed to restart/rerun workflow instance 68",
      "78": "Restarting of workflow instance failed for instance Id due to_
  ↳exception Failed to restart/rerun workflow instance 78"
    }
  },
  "page": null
}
```

1.5.8 Fetch Workflow Execution History

This API allows you to get the status of a specific workflow instance within a project or fetch all the workflows that have a specific status, within a project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/reports/execution?
  ↳projectIds={project_Id}
```

Note: You must pass the *project_Id* parameter in the request URL. To fetch the execution history of a workflow under the public project, set the value for the *project_Id* parameter to *1*.

Method:

POST

Request Payload:

```
{
  "instanceId": "<This is the workflow instance Id for which the execution_
↪history is requested.>",
  "sortBy": "<This is the field name over which the response result is sorted._
↪The default sorting is on the modified date so that the latest workflow execution_
↪instance comes on the top.>",
  "wfId": "<This is the workflow Id for which the execution history is_
↪requested. You must provide the value for this parameter to fetch the workflow_
↪execution history for a specific workflow (if 'wfName' is not provided). You may_
↪pass it as blank to retrieve the information about all the executed workflows.>",
  "chunkSize": "<This is the number of records required for each service_
↪request. Set this value to get the records corresponding to the specified value._
↪When set as '0' (or unspecified), all results are fetched.>",
  "wfName": "<This is the workflow name for which the execution history is requested._
↪You must provide the value for this parameter to fetch the workflow execution_
↪history for a specific workflow (if 'wfId' is not provided).>",
  "pageNo": "<This is the current page required for paginating the request_
↪depending on the total records and chunkSize.>",
  "orderBy": "<This is required to set the order of the search. The values can_
↪be ASC/DESC.>",
  "executionStatus": "<This is the workflow instance execution status. If you_
↪set this parameter to blank, the system returns all the results. This results can_
↪be narrowed down by setting this status to SUCCESS/FAILED/STOPPED/IN PROGRESS.>"
}
```

Note: Except *wfId*, *chunkSize* and *pageNo* that are of the *Integer* data type; all other parameters in the payload accepts *String* values.

Example Request:

- Scenario 1:

If you use the workflow name.

```
{
  "chunkSize": 10,
  "wfName": "Hive_file",
}
```

- Scenario 2:

If you use the workflow Id.

```
{
  "wfId": 21,
  "chunkSize": 5,
  "wfName": null,
  "pageNo": 2,
  "orderBy": "DESC",
  "executionStatus": "SUCCESS"
}
```


- Scenario 3:

If you use the workflow Id and instance Id.

```
{
  "instanceId": 55,
  "wfId": 21,
  "orderBy": "DESC",
}
```

Example Response:

```
{
  "responseMessage": "",
  "restUri": "/bedrock-app/services/rest/workflows/reports/execution",
  "result": {
    "currentPage": "0",
    "totalRecords": "6",
    "list": [
      {
        "clusterId": "192_168_2_150",
        "customStatus": null,
        "endDate": "11/09/2016 14:10:04",
        "executedBy": "wfuser",
        "firstInstanceStartDate": "",
        "instanceId": "67",
        "queueName": "",
        "scheduledBy": "",
        "startDate": "11/09/2016 14:09:58",
        "status": "SUCCESS",
        "version": 1,
        "wfDescription": "",
        "wfId": 21,
        "wfName": "Hive_file",
        "wfType": "GENERAL"
      },
      {
        "clusterId": "192_168_2_150",
        "customStatus": null,
        "endDate": "11/09/2016 12:32:58",
        "executedBy": "wfuser",
        "firstInstanceStartDate": "",
        "instanceId": "57",
        "queueName": "",
        "scheduledBy": "",
        "startDate": "11/09/2016 12:32:53",
        "status": "SUCCESS",
        "version": 1,
        "wfDescription": "",
        "wfId": 21,
        "wfName": "Hive_file",
        "wfType": "GENERAL"
      },
      {
        "clusterId": "192_168_2_150",
        "customStatus": null,
        "endDate": "11/09/2016 13:05:24",
        "executedBy": "admin",
        "firstInstanceStartDate": "",

```

```
    "instanceId": "55",
    "queueName": "",
    "scheduledBy": "",
    "startDate": "11/09/2016 13:05:18",
    "status": "SUCCESS",
    "version": 2,
    "wfDescription": "",
    "wfId": 21,
    "wfName": "Hive_file",
    "wfType": "GENERAL"
  },
  {
    "clusterId": "192_168_2_150",
    "customStatus": null,
    "endDate": "11/09/2016 12:28:58",
    "executedBy": "wfuser",
    "firstInstanceStartDate": "",
    "instanceId": "55",
    "queueName": "",
    "scheduledBy": "",
    "startDate": "11/09/2016 12:28:53",
    "status": "RESTARTED",
    "version": 1,
    "wfDescription": "",
    "wfId": 21,
    "wfName": "Hive_file",
    "wfType": "GENERAL"
  },
  {
    "clusterId": "192_168_2_150",
    "customStatus": null,
    "endDate": "11/09/2016 12:22:04",
    "executedBy": "wfuser",
    "firstInstanceStartDate": "",
    "instanceId": "54",
    "queueName": "",
    "scheduledBy": "",
    "startDate": "11/09/2016 12:21:58",
    "status": "SUCCESS",
    "version": 1,
    "wfDescription": "",
    "wfId": 21,
    "wfName": "Hive_file",
    "wfType": "GENERAL"
  },
  {
    "clusterId": "192_168_2_150",
    "customStatus": null,
    "endDate": "11/09/2016 12:08:46",
    "executedBy": "wfuser",
    "firstInstanceStartDate": "",
    "instanceId": "52",
    "queueName": "",
    "scheduledBy": "",
    "startDate": "11/09/2016 12:08:41",
    "status": "FAILED",
    "version": 1,
    "wfDescription": "",
```

```
        "wfId": 21,  
        "wfName": "Hive_file",  
        "wfType": "GENERAL"  
      }  
    ],  
    "page": null  
  }  
}
```

Response in Case of Failure:

If you send an invalid request payload:

```
{  
  "responseMessage": "Unexpected character (' ' (code 125)): expected a valid_  
↪value (number, String, array, object, 'true', 'false' or 'null')",  
  "restUri": "/bedrock-app/services/rest/workflows/reports/execution",  
  "result": null,  
  "page": null  
}
```

1.5.9 Fetch Workflow Status

This API allows you to get the status of a specific workflow instance within a project. The possible status options are: *SUCCESS*, *FAILED*, *IN PROGRESS*, and *STOPPED*.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/instances/{instance_Id}  
↪/status?projectId={project_Id}
```

Note: You must pass the *project_Id* and *instance_Id* parameters in the API URL. To fetch the status of a workflow under the public project, set the *project_Id* parameter to 1.

Method:

```
GET
```

Example Response:

```
{  
  "responseMessage": "Fetch successful for instance Id :46",  
  "restUri": "/bedrock-app/services/rest/workflows/instances/46/status",  
  "result": {  
    "wfId": 15,  
    "version": 1,  
    "wfName": "filemove_regex",  
    "instanceId": 46,  
    "wfStatus": "FAILED",  
    "customStatus": null,  
    "list": [  
      {  
        "stepExecutionHistoryKey": {  
          "id": 0,  
          "wfId": 15,  

```

```

        "stepId": 50,
        "instanceId": 46
    },
    "wfName": "filemove_regex",
    "stepName": "filemov_hdfs_local",
    "stepStatus": "SUCCESS",
    "startDate": "06/21/2016 17:12:58",
    "endDate": "06/21/2016 17:13:03",
    "stepType": "File Move Action",
    "initiatedBy": "admin",
    "version": 1,
    "parentBedrockInstanceId": 0,
    "parentStepId": 0,
    "stepId": 50,
    "customStatus": null
},
{
    "stepExecutionHistoryKey": {
        "id": 0,
        "wfId": 15,
        "stepId": 51,
        "instanceId": 46
    },
    "wfName": "filemove_regex",
    "stepName": "hdfs to hdfs",
    "stepStatus": "SUCCESS",
    "startDate": "06/21/2016 17:13:03",
    "endDate": "06/21/2016 17:13:09",
    "stepType": "File Move Action",
    "initiatedBy": "admin",
    "version": 1,
    "parentBedrockInstanceId": 0,
    "parentStepId": 0,
    "stepId": 51,
    "customStatus": null
},
{
    "stepExecutionHistoryKey": {
        "id": 0,
        "wfId": 15,
        "stepId": 52,
        "instanceId": 46
    },
    "wfName": "filemove_regex",
    "stepName": "hdfs to s3",
    "stepStatus": "SUCCESS",
    "startDate": "06/21/2016 17:13:09",
    "endDate": "06/21/2016 17:13:24",
    "stepType": "File Move Action",
    "initiatedBy": "admin",
    "version": 1,
    "parentBedrockInstanceId": 0,
    "parentStepId": 0,
    "stepId": 52,
    "customStatus": null
},
{
    "stepExecutionHistoryKey": {

```

```

        "id": 0,
        "wfId": 15,
        "stepId": 53,
        "instanceId": 46
    },
    "wfName": "filemove_regex",
    "stepName": "s3 to s3",
    "stepStatus": "SUCCESS",
    "startDate": "06/21/2016 17:13:24",
    "endDate": "06/21/2016 17:13:49",
    "stepType": "File Move Action",
    "initiatedBy": "admin",
    "version": 1,
    "parentBedrockInstanceId": 0,
    "parentStepId": 0,
    "stepId": 53,
    "customStatus": null
},
{
    "stepExecutionHistoryKey": {
        "id": 0,
        "wfId": 15,
        "stepId": 54,
        "instanceId": 46
    },
    "wfName": "filemove_regex",
    "stepName": "s3 to hdfs",
    "stepStatus": "SUCCESS",
    "startDate": "06/21/2016 17:13:49",
    "endDate": "06/21/2016 17:13:59",
    "stepType": "File Move Action",
    "initiatedBy": "admin",
    "version": 1,
    "parentBedrockInstanceId": 0,
    "parentStepId": 0,
    "stepId": 54,
    "customStatus": null
},
{
    "stepExecutionHistoryKey": {
        "id": 0,
        "wfId": 15,
        "stepId": 55,
        "instanceId": 46
    },
    "wfName": "filemove_regex",
    "stepName": "hdfs to s3 again",
    "stepStatus": "SUCCESS",
    "startDate": "06/21/2016 17:13:59",
    "endDate": "06/21/2016 17:14:15",
    "stepType": "File Move Action",
    "initiatedBy": "admin",
    "version": 1,
    "parentBedrockInstanceId": 0,
    "parentStepId": 0,
    "stepId": 55,
    "customStatus": null
},

```

```

{
  "stepExecutionHistoryKey": {
    "id": 0,
    "wfId": 15,
    "stepId": 56,
    "instanceId": 46
  },
  "wfName": "filemove_regex",
  "stepName": "s3 to local",
  "stepStatus": "SUCCESS",
  "startDate": "06/21/2016 17:14:15",
  "endDate": "06/21/2016 17:14:25",
  "stepType": "File Move Action",
  "initiatedBy": "admin",
  "version": 1,
  "parentBedrockInstanceId": 0,
  "parentStepId": 0,
  "stepId": 56,
  "customStatus": null
},
{
  "stepExecutionHistoryKey": {
    "id": 0,
    "wfId": 15,
    "stepId": 57,
    "instanceId": 46
  },
  "wfName": "filemove_regex",
  "stepName": "filemove local to local",
  "stepStatus": "FAILED",
  "startDate": "06/21/2016 17:14:25",
  "endDate": "06/21/2016 17:14:30",
  "stepType": "File Move Action",
  "initiatedBy": "admin",
  "version": 1,
  "parentBedrockInstanceId": 0,
  "parentStepId": 0,
  "stepId": 57,
  "customStatus": null
}
],
"instanceVersion": 0,
"queueName": "default",
"logList": [
  {
    "type": "FILE",
    "value": "/projects/bedrock-app/logs/WORKFLOW/
↪filemove_regex/46/46-DEBUG.log",
    "wfId": 0,
    "instanceId": 0,
    "logFilePath": null,
    "pageNo": 0,
    "chunkSize": 0
  }
]
},
"page": null
}

```

Response in Case of Failure:

If instance Id is invalid:

```
{
  "responseMessage": "For input string: \"123213s\"",
  "restUri": "/bedrock-app/services/rest/workflows/instances/123213s/status",
  "result": null,
  "page": null
}
```

1.5.10 Restart Workflow Instance

This API allows you to restart a *FAILED* workflow instance. Two types of payload can be provided based on the requirement.

- Modify workflow properties before restarting.
- Restart workflow directly by using *workflow_Id* only.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/{workflow_Id}/restart?
↪projectIds={project_Id}
```

Note: You must pass the *project_Id* and *workflow_Id* parameters in the API URL. To restart a workflow under the public project, set the *project_Id* parameter to *1*.

Method:

POST

Request Payload:

- Scenario 1:

To restart a *FAILED* workflow instance:

```
{
  *"instanceId": "<This is the FAILED workflow instance Id which you want to restart.
  ↪>"
}
```

- Scenario 2:

To modify the parameters of a *FAILED* workflow instance and restart, perform the following steps:

1. To get the details of a failed workflow instance, use the *Fetch Workflow Instance Details* API (for the *FAILED* workflow instance).
2. Modify the desired parameter(s) values accordingly.

Example Request:

- Scenario 1:

If you use the instance Id:

```
{
  "instanceId": 294
}
```

- Scenario 2:

If you modify parameters and restart a *FAILED* workflow instance:

```
{
  "wfId": 51,
  "categoryId": 1,
  "wfName": "p_dq",
  "createdBy": "admin",
  "createdDate": "08/01/2017 16:49:58",
  "modifiedBy": "admin",
  "modifiedDate": "08/01/2017 20:23:10",
  "wfSeverity": "",
  "wfNotification": "N",
  "wfStepNotification": "N",
  "notificationDetails": [],
  "globalParameter": ",,,,,,,,",
  "stepList": [
    {
      "stepId": 179,
      "actionId": 1,
      "stepName": "Start",
      "shapeCoordinates": "217,157",
      "stepSeverity": "",
      "stepNotification": "N",
      "yesStepId": 0,
      "yesStepName": null,
      "noStepId": 0,
      "noStepName": null,
      "pairConnectionId": 0,
      "pairConnectionName": null,
      "stepParamList": [],
      "stepConnectionList": [
        {
          "stepConnectionId": 122,
          "nextId": 181,
          "nextStepName": "dq_s1"
        }
      ],
      "isAlreadyExecuted": "",
      "stepSubList": null,
      "stepType": "Start",
      "subWorkflowId": 0,
      "subWorkFlow": null,
      "description": null,
      "skipped": false
    },
    {
      "stepId": 181,
      "actionId": 18,
      "stepName": "dq_s1",
      "shapeCoordinates": "333,144",
      "stepSeverity": "",
      "stepNotification": "N",

```



```

"yesStepId": 0,
"yesStepName": null,
"noStepId": 0,
"noStepName": null,
"pairConnectionId": 0,
"pairConnectionName": null,
"stepParamList": [
  {
    "stepParamId": 788,
    "key": "invalidDataPath",
    "valueText": "/user/bedrock/praj/dq/report
↪ "
  },
  {
    "stepParamId": 789,
    "key": "entityTypeAndVersion",
    "valueText": "dq_par_test(25.1)"
  },
  {
    "stepParamId": 790,
    "key": "partitionSpec",
    "valueText": "[{"partitionFieldName": "\
↪ "pl\"", "\"value\": \"\""}]"
  },
  {
    "stepParamId": 791,
    "key": "description",
    "valueText": null
  },
  {
    "stepParamId": 792,
    "key": "createGoodDataEntity",
    "valueText": "false"
  },
  {
    "stepParamId": 793,
    "key": "inputFromEntity",
    "valueText": "false"
  },
  {
    "stepParamId": 794,
    "key": "badDataOPLocation",
    "valueText": "/user/bedrock/praj/dq/bad"
  },
  {
    "stepParamId": 795,
    "key": "createBadDataEntity",
    "valueText": "false"
  },
  {
    "stepParamId": 796,
    "key": "dataFilePath",
    "valueText": "/user/bedrock/praj/dq/input"
  },
  {
    "stepParamId": 797,
    "key": "proceedOnFailure",
    "valueText": "false"
  }
]

```

```

        },
        {
            "stepParamId": 798,
            "key": "esReportRequired",
            "valueText": "true"
        },
        {
            "stepParamId": 799,
            "key": "fieldDelimiterDQ",
            "valueText": null
        },
        {
            "stepParamId": 800,
            "key": "validDataPath",
            "valueText": "/user/bedrock/praj/dq/good"
        }
    ],
    "stepConnectionList": [
        {
            "stepConnectionId": 123,
            "nextId": 180,
            "nextStepName": "Stop"
        }
    ],
    "isAlreadyExecuted": "",
    "stepSubList": null,
    "stepType": "Data Quality Action",
    "subWorkflowId": 0,
    "subWorkFlow": null,
    "description": null,
    "skipped": false
},
{
    "stepId": 180,
    "actionId": 5,
    "stepName": "Stop",
    "shapeCoordinates": "444,150",
    "stepSeverity": "",
    "stepNotification": "N",
    "yesStepId": 0,
    "yesStepName": null,
    "noStepId": 0,
    "noStepName": null,
    "pairConnectionId": 0,
    "pairConnectionName": null,
    "stepParamList": [],
    "stepConnectionList": [],
    "isAlreadyExecuted": "",
    "stepSubList": null,
    "stepType": "Stop",
    "subWorkflowId": 0,
    "subWorkFlow": null,
    "description": null,
    "skipped": false
},
    "bedrockInstanceId": 55,
    "executedBy": "admin",

```

```

    "scheduledBy": null,
    "wfLevelParameterList": [],
    "wfType": "GEN",
    "listCounterMap": {
        "totalSteps": "1"
    },
    "instanceVersion": 1,
    "categoryName": "unassigned",
    "bedrockURL": "http://192.168.1.39:9090/",
    "wfNamespaceList": [
        {
            "id": 0,
            "key": "PROJECT_ID",
            "value": "1",
            "overridable": false
        }
    ],
    "description": "",
    "adminCapacityQueues": {
        "queueId": 0,
        "queueName": "default",
        "queueCapacity": null,
        "queueElasticity": 0,
        "parentQueueId": 0,
        "secUser": null
    },
    "logLevel": "DEBUG",
    "baseEncodedString": null,
    "overwritable": false,
    "clusterId": "1_39",
    "executionPriority": "NORMAL",
    "stepRestart": false
}

```

Note: *bedrockInstanceId* and *instanceId* parameters are complimentary to each other. The usage across request payload can vary based on the service (or API) used.

Example Response:

```

{
    "responseMessage": "Workflow restarted successfully with the instance id - 294",
    "restUri": "/bedrock-app/services/rest/workflows/1/restart",
    "result": null,
    "page": null
}

```

Response in Case of Failure:

If a workflow instance with a *SUCCESS/RESTARTED* status is triggered:

```

{
    "responseMessage": "Restart is not applicable for instance id 323 and version_1, only latest version with failed status can be restarted.",
    "restUri": "/bedrock-app/services/rest/workflows/1/restart",
    "result": null,
}

```

```
}
  "page": null
}
```

1.5.11 Set Workflow Custom Status

This API allows you to set custom status to a specific version of a workflow instance, within a project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/customstatus?
↪projectIds={project_Id}
```

Note: You must pass the *project_Id* parameter in the API URL. To set the status for a workflow under the public project, set the *project_Id* parameter to *1*.

Method:

```
POST
```

Request Payload:

```
{
  *"instanceId": "<This is the workflow instance Id.>",
  *"version": "<This is the version of the workflow instance.>",
  "customStatus": "<This is the custom status message.>"
}
```

Note: If you do not pass *customStatus* in the payload, ZDP updates the parameter to *-*.

Example Request:

```
{
  "instanceId":257,
  "version":1,
  "customStatus":"wwd"
}
```

Example Response:

```
{
  "responseMessage": "Custom status for workflow with instanceId-257, version-1 has_
↪been modified successfully.",
  "restUri": "/bedrock-app/services/rest/workflows/customstatus",
  "result": "Custom status for workflow with instanceId-257, version-1 has been_
↪modified successfully.",
  "page": null
}
```

Response in Case of Failure:

If custom status exceeds 255 characters:

```
{
  "responseMessage": "Custom status cannot exceed 255 characters",
  "restUri": "/bedrock-app/services/rest/workflows/customstatus",
  "result": "Custom status cannot exceed 255 characters",
  "page": null
}
```

1.5.12 Set Workflow Step Custom Status

This API allows you to set custom status for a step, with a specific version belonging to a workflow instance, within a project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/step/customstatus?
↳projectIds={project_Id}
```

Note: You must pass the *project_Id* parameter in the API URL. To set the status for a workflow step under the public project, set the *project_Id* parameter to *1*.

Method:

```
POST
```

Request Payload:

```
{
  *"stepId": "<This is the step Id.>",
  *"instanceId": "<This is the workflow instance Id.>",
  *"version": "<This is the version of workflow instance.>",
  "customStatus": "<This is the custom status message.>"
}
```

Example Request:

```
{
  "stepId": 144,
  "instanceId": 96,
  "version": 1,
  "customStatus": "Dib ADI"
}
```

Note: If you do not pass *customStatus* in the payload, ZDP updates the parameter to *-*.

Example Response:

```
{
  "responseMessage": "Custom status for stepId-224,instanceId-257, version-1 has been_
↳modified successfully.",
  "restUri": "/bedrock-app/services/rest/workflows/step/customstatus",
  "result": "Custom status for stepId-224,instanceId-257, version-1 has been modified_
↳successfully.",
}
```

```
"page": null
}
```

Response in Case of Failure:

If the status for the workflow step cannot be set:

```
{
  "responseMessage": "Custom status for stepId-144,instanceId-257, version-1 could_
↪not be modified.",
  "restUri": "/bedrock-app/services/rest/workflows/step/customstatus",
  "result": "Custom status for stepId-144,instanceId-257, version-1 could not be_
↪modified.",
  "page": null
}
```

1.5.13 Get Workflow Schedule

This API fetches the workflow schedules in the JSON format for the workflows within a project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedules?projectId={project_Id}
↪&resourceType=WORKFLOW&resourceId={resource_Id}
```

URL Parameters:

- You must pass *project_Id*, *resource_type*, and *resource_Id* parameters in the API URL.
- Set the project Id against the *project_Id* parameter under which the resource (or workflow) exists.
- The *resource_Id* parameter indicates the workflow Id for which the existing schedules are fetched by this service.

Method:

```
GET
```

Example Response:

```
{
  "responseMessage": "Success",
  "restUri": null,
  "result": [
    {
      "scheduleInstanceId": 240,
      "scheduleName": "BEDROCK",
      "resourceId": 14,
      "resourceType": "WORKFLOW",
      "frequency": "RECURRING",
      "recurringDuration": "DAILY",
      "startDate": "11/09/2016",
      "startTime": "14:05",
      "minuteInterval": -1,
      "hourlyInterval": -1,
      "dailyInterval": 0,
      "weeklyInterval": "",
      "monthlyInterval": -1,
      "maxExecutions": 20,
    }
  ]
}
```

```

        "endDate": "11/10/2016",
        "endTime": "00:00",
        "started": "Y",
        "count": 1,
        "scheduledTime": "11/09/2016 13:58:47",
        "scheduledBy": "admin",
        "requestedBy": "admin",
        "nextTriggerTime": "",
        "resourceParams": null,
        "scheduleParameters": [
            {
                "scheduleParamId": 1170,
                "paramKey": "logLevel",
                "paramValue": "DEBUG"
            },
            {
                "scheduleParamId": 1173,
                "paramKey": "scheduleWfLevelParameters",
                "paramValue": "[]"
            },
            {
                "scheduleParamId": 1172,
                "paramKey": "scheduleGlobalParameters",
                "paramValue": "DJ_TEST,,,,,,,,,"
            },
            {
                "scheduleParamId": 1174,
                "paramKey": "scheduleWfNamespace",
                "paramValue": "[]"
            },
            {
                "scheduleParamId": 1171,
                "paramKey": "PROJECT_ID",
                "paramValue": "1"
            }
        ],
        "repeatBy": null,
        "scheduledStatus": "SCHEDULED"
    },
    "page": null
}

```

Response in Case of Failure:

If there is no workflow within a project:

```

{
    "responseMessage": "Workflow Id : 3 not available with project [1]",
    "restUri": "/bedrock-app/services/rest/schedules",
    "result": null,
    "page": null
}

```

1.5.14 Save Workflow Schedule

This API allows you to save a workflow schedule for the workflows within a project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/{project_Id}/schedules
```

Note: You must pass the *project_Id* parameter in the API URL. To save the workflow schedules under the public project, set the *project_Id* parameter to *1*.

Method:

```
POST
```

Request Payload:

```
{
  "scheduleInstanceId": "<This is the instance Id of the schedule. This value needs_
  ↳to be set as '0' while saving a schedule.>",
  * "scheduleName": "<This is the name of the schedule.>",
  * "resourceId": "<This is the Id of the workflow.>",
  * "resourceType": "<This is the resource type. This has to be WORKFLOW.>",
  "frequency": "<This is the frequency of execution. This can be ONCE, RECURRING, or_
  ↳CRONEXPRESSION.>",
  "cronExpression": "<This is the cron expresssion that you can use to run the_
  ↳schedule. This parameter is applicable only when the frequency parameter is set to_
  ↳CRONEXPRESSION. A sample value: 0 * ? * * 2018.>",
  * "recurringDuration": "<This is the recurring pattern. This parameter value can be_
  ↳DAILY, HOURLY, WEEKLY, MONTHLY, or MINUTE.>",
  * "startDate": "<This is the start date of schedule in the format- mm/dd/yyyy.>",
  * "startTime": "<This is the start time of the schedule in the 24 hr format- hh:mm.>",
  "minuteInterval": "<This is the minute interval. This parameter is applicable when
  ↳'recurringDuration' is set as MINUTE. Probable value can be any integer between 1_
  ↳to 60.>",
  "hourlyInterval": "<This is the hourly interval. This parameter is applicable when
  ↳'recurringDuration' is set as HOURLY. Probable value can be any integer between 1_
  ↳to 24.>",
  "weeklyInterval": "<This is the weekly interval. This parameter is applicable when
  ↳'recurringDuration' is set as WEEKLY. Probable values are: MON, TUE, WED, THU, FRI,_
  ↳or SAT or a combination of days like MON, WED, SAT.>",
  "monthlyInterval": "<This is the monthly interval. This parameter is applicable_
  ↳when 'recurringDuration' is set as MONTHLY. Probable value can be any integer_
  ↳between 1 to 30 (or 31).>",
  "maxExecutions": "<This is the maximum number of executions for the current_
  ↳schedule.>",
  * "endDate": "<This is the end date of schedule in the format- mm/dd/yyyy. The
  ↳'endDate' and 'endTime' is mandator only when the 'maxExecutions' is not set.>",
  * "endTime": "<This is the end time of the schedule in the 24 hr format- hh:mm.>",
  "started": "<This indicates if the schedule is started or not. The result as 'Y'_
  ↳ (Yes) or 'N' (No) indicating whether the existing schedule executed or not.>"
  "scheduledTime": "<Indicates the time when of the scheduled. This is a string value.
  ↳>",
  "scheduledBy": "<This field is applicable when impersonation is enabled and_
  ↳indicates the impersonated (or proxy) user executing the scheduled workflow. When_
  ↳unspecified, it picks up the logged in user (by default).>",
  "requestedBy": "<This is the user who wants to save the schedule.>",
  "repeatBy": "<This parameter is mandatory only when the 'recurringDuration' is set_
  ↳as 'MONTHLY'. Setting the value for this parameter as 'DAY_OF_MONTH' will schedule_
  ↳the workflow on a particular date (based on the start date) on monthly basis._
  ↳Settig this value as 'DAY_OF_WEEK' will schedule the workflow on a particular day_
  ↳ (based on the start day) of the week on monthly basis.>",
```



```

    "resourceParams": {
      "scheduleWfNamespace": [ "<Represents array of workflow namespaces_
↳that may be required during runtime.>"
        {
          "key": "<Set the namespace variable (if any) that needs to be_
↳called during execution.>",
          "value": "<Set the value for the namespace variable that must_
↳replaced during runtime.>",
          "overridable": "<Set true if you want to override the value_
↳for this namespace variable (when multiple namespaces variables with similar key_
↳exist across different namespace levels, such as 'CATEGORY', 'BEDROCK', or 'STEP')._
↳If not, set it 'false'.>"
        }
      ],
      "adminCapacityQueues": { "<This is a mandatory parameter if the_
↳Capacity Scheduler feature is enabled in ZDP.>"
        "queueName": "<This is the name of the capacity queue.>",
        "queueId": "<This is the Id of the capacity queue.>"
      }
      "logLevel": "<This is the log level for the workflow schedule being_
↳applied during runtime. The probable values can be 'INFO', 'DEBUG', 'ERROR'.>"
    }

    "scheduledStatus": "<This is the status for the workflow schedule. This parameter_
↳value can be 'SCHEDULED' or 'SUSPENDED'.>"
  }

```

Example Request:

- Scenario 1:

If the *frequency* parameter is set to *RECURRING*:

```

{
  "scheduleName": "CHIVAS_1",
  "resourceId": 239,
  "resourceType": "WORKFLOW",
  "frequency": "RECURRING",
  "recurringDuration": "DAILY",
  "startDate": "10/06/2017",
  "startTime": "00:00",
  "minuteInterval": -1,
  "hourlyInterval": -1,
  "dailyInterval": 0,
  "weeklyInterval": "",
  "monthlyInterval": -1,
  "maxExecutions": 3,
  "resourceParams": {
    "scheduleWfNamespace": [
      {
        "key": "myOutputLocation",
        "value": "/user/john/cfd226",
        "overridable": false
      },
      {
        "key": "input",
        "value": "/user/smith/incfd226",
        "overridable": true
      }
    ]
  }
}

```

```

    },
    "adminCapacityQueues": {
        "queueName": "high",
        "queueId": 7
    }
    "logLevel": "DEBUG"
}
}

```

- Scenario 2:

If the *frequency* parameter is set to *CRONEXPRESSION*:

```

{
  "scheduledBy": "admin",
  "endDate": "",
  "endTime": "",
  "frequency": "CRONEXPRESSION",
  "hourlyInterval": -1,
  "maxExecutions": null,
  "monthlyInterval": -1,
  "minuteInterval": -1,
  "recurringDuration": "",
  "scheduleName": "test_schedule",
  "startDate": "02/20/2018",
  "startTime": "00:00",
  "weeklyInterval": "",
  "scheduleInstanceId": 0,
  "resourceType": "WORKFLOW",
  "resourceParams": {
    "logLevel": "DEBUG",
    "scheduleWfNamespace": [

  ],
  "scheduleWfLevelParameters": [

  ],
  "scheduleGlobalParameters": ",,,,,,,,",
  "adminCapacityQueues": null
},
  "resourceId": 39,
  "cronExpression": "0 * ? * * 2018"
}

```

Example Response:

```

{
  "responseMessage": "Workflow scheduled successfully with schedule instance id_
↪:9",
  "restUri": "/projects/3/schedules",
  "result": null,
  "page": null
}

```

Response in Case of Failure:

- Scenario 1:

If the workflow schedule cannot be saved:

```
{
  "responseMessage": "Failed to save schedule information",
  "restUri": "/bedrock-app/services/rest/projects/1/schedules",
  "result": null,
  "page": null
}
```

- Scenario 2:

If the workflow start time provided is before the current time:

```
{
  "responseMessage": "Invalid scheduling parameters : Start time cannot be on or ↵
↵before current time",
  "restUri": "/bedrock-app/services/rest/projects/1/schedules",
  "result": null,
  "page": null
}
```

- Scenario 3:

If invalid scheduling parameters are used in the response code:

```
{
  "responseMessage": "Invalid scheduling parameters :",
  "restUri": "/bedrock-app/services/rest/projects/1/schedules",
  "result": null,
  "page": null
}
```

- Scenario 4:

If the same schedule name is used twice:

```
{
  "responseMessage": "Schedule details already exists for SCHEDULE_NAME ↵
↵:CHIVI",
  "restUri": "/bedrock-app/services/rest/projects/1/schedules",
  "result": null,
  "page": null
}
```

- Scenario 5:

If the *resourceId* (or the Workflow) does not exist:

```
{
  "responseMessage": "Workflow Details does not exist for workflow id :121",
  "restUri": "/bedrock-app/services/rest/projects/1/schedules",
  "result": null,
  "page": null
}
```

1.5.15 Resume/Suspend Workflow Schedule

This API allows you to suspend/resume a workflow schedule within a project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedules/{schedule_instance_Id}/  
→actions/{action_type}?projectId={project_Id}
```

Note: You must pass the *schedule_instance_Id*, *action_type*, and *project_Id* parameters in the API URL. To suspend/resume a workflow schedule under the public project, set the *project_Id* parameter to 1.

Method:

POST

Input Parameters:

- The value for *schedule_instance_Id* can be retrieved by using the *Get Workflow Schedule* API.
- The action placeholder in the URL decides the action of this service. The possible value of action can be:
 - *SUSPEND*: This action halts the schedule that was set earlier.

Note: A workflow schedule can be suspended only if it has executed at least once.

- *RESUME*: This action resumes the workflow schedule that was suspended earlier.

Example Response:

- Scenario 1:

If you suspend an existing workflow schedule:

```
{  
  "responseMessage": "SUSPEND action on scheduled workflow successfully_  
→executed",  
  "restUri": null,  
  "result": "SUSPEND",  
  "page": null  
}
```

- Scenario 2:

If you schedule a suspended workflow schedule:

```
{  
  "responseMessage": "SCHEDULED action on scheduled workflow successfully_  
→executed",  
  "restUri": null,  
  "result": "SCHEDULED",  
  "page": null  
}
```

Response in Case of Failure:

If a workflow cannot be unscheduled:

```
{  
  "responseMessage": "Workflow could not be unscheduled",  
  "restUri": "/bedrock-app/services/rest/schedules/1/actions/suspended",  
  "result": null,  
}
```

```
}
  "page": null
}
```

1.5.16 Resume/Suspend Workflow Schedules in Bulk

This API allows you to resume/suspend the workflow schedules, in bulk.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedules/bulk
```

Method:

```
PUT
```

Request Payload:

```
{
  * "scheduleInstanceIds": "<This indicates the schedule instance Ids for the
  ↳ schedules for which you want to set the scheduled status.>"
  * "property": "<This is the key for setting the scheduled status. For example,
  ↳ scheduledStatus.>"
  * "value": "<This is the new status for the workflow schedules. For example,
  ↳ SUSPENDED, SCHEDULED.>"
}
```

Example Request:

```
{
  "scheduleInstanceIds": [1,2],
  "property": "scheduledStatus",
  "value": "SUSPENDED"
}
```

Example Response:

```
{
  "responseMessage": "Selected schedules updated successfully.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [
      1
    ],
    "failedIds": [
    ],
    "failureCause": null
  },
  "page": null
}
```

Response in Case of Failure:

- Scenario 1:

If some of the specified *scheduleInstanceIds* do not exist:

```
{
  "responseMessage": "Selected schedules updated, with errors.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [
      1
    ],
    "failedIds": [
      2
    ],
    "failureCause": null
  },
  "page": null
}
```

- Scenario 2:

If *scheduledStatus* is not specified as the value for the *property* parameter in the request payload:

```
{
  "responseMessage": "An Error Occurred While Serving Request. Check Logs For More_↵
  Details",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": null,
  "page": null
}
```

- Scenario 3:

If the specified *scheduleInstanceIds* do not exist:

```
{
  "responseMessage": "Failed to update selected schedules.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [],
    "failedIds": [
      2
    ],
    "failureCause": null
  },
  "page": null
}
```

1.5.17 Unschedule Workflow

This API removes/unschedules an existing workflow schedule within a project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedules/{schedule_instance_Id}?
↵projectIds={project_Id}
```

Before using this service, note the following points:

- You must pass the *schedule_instance_Id* and *project_Id* parameters in the API URL.
- To retrieve the *schedule_instance_Id*, you must first use the *Get Workflow Schedule* API.


```
{
  "artifacts": [{
    "page": {
      "chunkSize": 20,
      "currentPage": 1,
      "sortBy": "modifiedTime",
      "sortOrder": "DESC"
    },
    "searchTerm": "",
    "artifactType": "WORKFLOW",
    "facetFilter": [{
      "facetName": "projectIds",
      "facetValue": [1]
    }]
  }]
}
```

Example Response:

```
{
  "page": {
    "chunkSize": 20,
    "currentPage": 1,
    "sortBy": "modifiedTime",
    "sortOrder": "DESC"
  },
  "searchTerm": "",
  "artifactType": "WORKFLOW",
  "facetFilter": [
    {
      "facetName": "projectIds",
      "facetValue": [
        1
      ]
    }
  ]
}
```

1.5.19 Update Log Level for Workflow Schedules in Bulk

This API allows you to modify the log level for the workflow schedules, in bulk.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedules/bulk
```

Method:

```
PUT
```

Request Payload

```
{
  * "scheduleInstanceIds": "<This indicates the schedule instance Ids for the_
  ↳ schedules for which you want to update the log level.>"
  * "property": "<This is the key for setting the log level for the workflow schedules.
  ↳ >"
  * "value": "<This is the value of the key for setting the log level for the workflow_
  ↳ schedules. For example, ERROR, INFO, DEBUG.>"
}
```


Example Request:

```
{
  "scheduleInstanceIds": [1,2],
  "property": "logLevel",
  "value": "DEBUG"
}
```

Example Response:

```
{
  "responseMessage": "Selected schedules updated successfully.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [
      1
    ],
    "failedIds": [],
    "failureCause": null
  },
  "page": null
}
```

Response in Case of Failure:

- Scenario 1:

If you do not provide *value* as a parameter in the request payload:

```
{
  "responseMessage": "Invalid log level provided",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": null,
  "page": null
}
```

- Scenario 2:

If you do not provide *property/scheduleInstanceIds* as a parameter in the request payload:

```
{
  "responseMessage": "An Error Occurred While Serving Request. Check Logs For More_↵  
↵Details",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": null,
  "page": null
}
```

- Scenario 3:

If some of the specified *scheduleInstanceIds* do not exist:

```
{
  "responseMessage": "Selected schedules updated, with errors.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [
      216
    ]
  }
}
```

```
    ],
    "failedIds": [
      1
    ],
    "failureCause": null
  },
  "page": null
}
```

- Scenario 4:

If the specified *scheduleInstanceIds* do not exist:

```
{
  "responseMessage": "Failed to update selected schedules.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [],
    "failedIds": [
      1,
      2
    ],
    "failureCause": null
  },
  "page": null
}
```

1.5.20 Update Cluster Id for Workflow Schedules in Bulk

This API allows you to modify the cluster Id for the workflow schedules, in bulk.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedules/bulk
```

Method:

```
PUT
```

Request Payload

```
{
  * "scheduleInstanceIds": "<This indicates the schedule instance Ids for the_
  ↳ schedules for which you want to update the cluster Id.>"
  * "property": "<This is the key for setting the hadoop cluster for schedule workflow_
  ↳ execution. For example, HADOOP_CLUSTER_ID.>"
  * "value": "<This is the updated cluster Id for the schedules.>"
}
```

Example Request:

```
{
  "scheduleInstanceIds": [1,2],
  "property": "HADOOP_CLUSTER_ID",
  "value": "cluster_id"
}
```

Example Response:

```
{
  "responseMessage": "Selected schedules updated successfully.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [
      215
    ],
    "failedIds": [],
    "failureCause": null
  },
  "page": null
}
```

Response in Case of Failure:

• Scenario 1:

If *HADOOP_CLUSTER_ID* is not specified as the value for the *property* parameter in the request payload:

```
{
  "responseMessage": "An Error Occurred While Serving Request. Check Logs For More_↵  
↵Details",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": null,
  "page": null
}
```

• Scenario 2:

If the specified *scheduleInstanceIds* do not exist:

```
{
  "responseMessage": "Failed to update selected schedules.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [],
    "failedIds": [
      1,
      2
    ],
    "failureCause": null
  },
  "page": null
}
```

• Scenario 3:

If some of the specified *scheduleInstanceIds* do not exist:

```
{
  "responseMessage": "Selected schedules updated, with errors.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [
      216
    ],
    "failedIds": [
      1
    ]
  }
}
```

```
    ],
    "failureCause": null
  },
  "page": null
}
```

1.5.21 Update the Execution Priority for Workflow Schedules in Bulk

This API allows you to set the execution priority for the workflow schedules, in bulk.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedules/bulk
```

Method:

```
PUT
```

Request Payload:

```
{
  *"scheduleInstanceIds": "<This indicates the schedule instance Ids for the schedules_
  ↳for which you want to set the execution priority.>"
  *"property": "<This is the key for setting the priority for execution for the_
  ↳selected workflow schedules. For example, EXECUTION_PRIORITY.>"
  *"value": "<This is the value of the key for setting the priority for execution for_
  ↳the workflow schedules. For example, HIGH, NORMAL, LOW.>"
}
```

Example Request:

```
{
  "scheduleInstanceIds": [1,2],
  "property": "EXECUTION_PRIORITY",
  "value": "HIGH"
}
```

Example Response:

```
{
  "responseMessage": "Selected schedules updated successfully.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [
      1
    ],
    "failedIds": [],
    "failureCause": null
  },
  "page": null
}
```

Response in Case of Failure:

- Scenario 1:

If *EXECUTION_PRIORITY* is not set as the value for the *property* parameter in the request payload:

```
{
  "responseMessage": "An Error Occurred While Serving Request. Check Logs For More_↵
  Details",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": null,
  "page": null
}
```

- Scenario 2:

If the specified *scheduleInstanceIds* do not exist:

```
{
  "responseMessage": "Failed to update selected schedules.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [],
    "failedIds": [
      2
    ],
    "failureCause": null
  },
  "page": null
}
```

- Scenario 3:

If some of the specified *scheduleInstanceIds* do not exist:

```
{
  "responseMessage": "Selected schedules updated, with errors.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [
      216
    ],
    "failedIds": [
      1
    ],
    "failureCause": null
  },
  "page": null
}
```

1.5.22 Update the Run As User for Workflow Schedules in Bulk

This API allows you to update the user who runs the workflow schedules, in bulk. To use this API, impersonation must be enabled in the ZDP instance.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedules/bulk
```

Method:

```
PUT
```

Request Payload:

```
{
  * "scheduleInstanceIds": "<This indicates the schedule instance Ids for the
  ↳ schedules for which you want to update the run as user.>"
  * "property": "<This is the key for setting the run as user. For example,
  ↳ scheduledBy.>"
  * "value": "<This is the new impersonated user. For example, doAsUser10.>"
}
```

Example Request:

```
{
  "scheduleInstanceIds": [1,2],
  "property": "scheduledBy",
  "value": "doAsUser10"
}
```

Example Response:

```
{
  "responseMessage": "Selected schedules updated successfully.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [
      222
    ],
    "failedIds": [],
    "failureCause": null
  },
  "page": null
}
```

Response in Case of Failure:

- Scenario 1:

If some of the specified *scheduleInstanceIds* do not exist:

```
{
  "responseMessage": "Selected schedules updated, with errors.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [
      222
    ],
    "failedIds": [
      0
    ],
    "failureCause": null
  },
  "page": null
}
```

- Scenario 2:

If the specified *scheduleInstanceIds* do not exist:

```
{
  "responseMessage": "Failed to update selected schedules.",
}
```

```

"restUri": "/bedrock-app/services/rest/schedules/bulk",
"result": {
  "successIds": [],
  "failedIds": [
    2
  ],
  "failureCause": null
},
"page": null
}

```

- Scenario 3:

If *scheduledBy* is not specified as a value for the *property* parameter in the request payload:

```

{
  "responseMessage": "An Error Occurred While Serving Request. Check Logs For More_↵
  Details",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": null,
  "page": null
}

```

- Scenario 4:

If impersonation is not enabled in the ZDP instance:

```

{
  "responseMessage": "Impersonation is not enabled, hence scheduled by user cannot_↵
  be updated",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": null,
  "page": null
}

```

1.5.23 Delete Workflow Schedules in Bulk

This API allows you to delete workflow schedules, in bulk.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedules/bulk
```

Method:

```
DELETE
```

Request Payload:

```
*[schedule_instance_Id, schedule_instance_Id,...]
```

Example Request:

```
[1,2]
```

Example Response:

```
{
  "responseMessage": "Selected schedules deleted successfully",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [
      210
    ],
    "failedIds": [],
    "failureCause": null
  },
  "page": null
}
```

Response in Case of Failure:

- Scenario 1:

If the specified *schedule_instance_Id* does not exist:

```
{
  "responseMessage": "Failed to delete selected schedules.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [],
    "failedIds": [
      10,
      11
    ],
    "failureCause": null
  },
  "page": null
}
```

- Scenario 2:

If some of the specified *schedule_instance_Id* exist while some do not:

```
{
  "responseMessage": "Selected schedules deleted, with errors.",
  "restUri": "/bedrock-app/services/rest/schedules/bulk",
  "result": {
    "successIds": [
      7
    ],
    "failedIds": [
      10
    ],
    "failureCause": null
  },
  "page": null
}
```

1.5.24 Fetch the Status of the Workflow Scheduler

This API allows you to fetch the status of the workflow scheduler.

URL:


```
http://<bedrock-host>:port/bedrock-app/services/rest/schedules/globalStatus/  
↪resourceType/{resource_type}
```

Note: You must pass the value for the *resource_type* parameter in the API URL as *WORKFLOW*.

Method:

```
GET
```

Example Response:

Note: The *operationalStatus* parameter is displayed as either *SUSPENDED* or *SCHEDULED*.

```
{  
  "responseMessage": "SUCCESS",  
  "restUri": "/bedrock-app/services/rest/schedules/globalStatus/resourceType/WORKFLOW",  
  "result": {  
    "resourceType": "WORKFLOW",  
    "operationalStatus": "SCHEDULED",  
    "lastUpdatedDate": "Tue Feb 06 15:54:12 IST 2018",  
    "lastUpdatedBy": "admin"  
  },  
  "page": null  
}
```

Response in Case of Failure:

If the *resource_type* parameter is not defined as *WORKFLOW*:

```
{  
  "responseMessage": "Index: 0, Size: 0",  
  "restUri": "/bedrock-app/services/rest/schedules/globalStatus/resourceType/WORKFLOWS  
↪",  
  "result": null,  
  "page": null  
}
```

1.5.25 Pause/Unpause Workflow Schedules Globally

This API allows you to pause/unpause the workflow scheduler globally. Thus, all the scheduled jobs are suspended or scheduled.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedules/globalStatus/  
↪resourceType/{resource_type}
```

Note: You must pass the value for the *resource_type* parameter in the API URL as *WORKFLOW*.

Method:

PUT

Request Payload:

```
{
  "property": "<This is the status for the workflow schedules. Specify this key as_
↳ operationalStatus.>",
  "value": "<This is the status for the workflow schedules that you want it to be._
↳ Based on this value (SUSPENDED/SCHEDULED), the system pauses/unpauses all the_
↳ workflow schedules globally.>"
}
```

Example Request:

```
{
  "property": "operationalStatus",
  "value": "SUSPENDED"
}
```

Example Response:

```
{
  "responseMessage": "Schedule global operational status updated to SUSPENDED_
↳ successfully for resource type: WORKFLOW",
  "restUri": "/bedrock-app/services/rest/schedules/globalStatus/resourceType/WORKFLOW",
  "result": {
    "resourceType": "WORKFLOW",
    "operationalStatus": "SUSPENDED",
    "lastUpdatedDate": "Fri Feb 09 18:55:48 IST 2018",
    "lastUpdatedBy": "admin"
  },
  "page": null
}
```

Response in Case of Failure:

If the *resource_type* parameter is not defined as *WORKFLOW*:

```
{
  "responseMessage": "Please enter valid resource type - WORKFLOW",
  "restUri": "/bedrock-app/services/rest/schedules/globalStatus/resourceType/WORKFLOWS
↳ ",
  "result": null,
  "page": null
}
```

1.5.26 Export Workflow

This API exports a workflow that belongs to a specific project in a JSON format.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/export?projectIds=
↳ {project_Id}
```

Note: You must pass the *project_Id* parameter in the API URL. To export a workflow under the public project, set the *project_Id* parameter to 1.

Method:

```
POST
```

Request Payload:

```
{
  * "wfName": "<This is the workflow name.>"
}
```

Note: You can export multiple workflows by entering comma-separated workflow names in the request payload.

Example Request:

```
{
  * "wfName": "Hive_test,Watermark_workflow"
}
```

Example Response:

The workflow details are in the JSON format. This can later be imported to other ZDP instances by using the *Import Workflow* API. The JSON response can be extensive depending on the size of the workflow.

For more information, refer to the *Export-Import* section.

Response in Case of Failure:

- Scenario 1:

If the workflow does not exist:

```
{
  "responseMessage": "Error while exporting workflow details : No records found",
  "restUri": "/bedrock-app/services/rest/workflows/export",
  "result": null,
  "page": null
}
```

- Scenario 2:

If the workflow does not belong to the specific project:

```
{
  "responseMessage": "Error while exporting workflow details : Missing resource_
↪ associations for following artifact identifiers - 6",
  "restUri": "/bedrock-app/services/rest/workflows/export",
  "result": null,
  "page": null
}
```

This API imports an exported workflow within a project. You can set the *overwritable* parameter to *true* to overwrite any existing workflow name in the target instance.

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/import?projectIds=
↳ {project_Id}
```

Note: You must pass the *project_Id* parameter in the API URL. To import a workflow under the public project, set the *project_Id* parameter to 1.

POST

```
{
  "jobList": "<Specify an array of the workflow objects.>"
}
```

```
{
  "jobList": [
    {
      "description": "",
      "createdBy": "john",
      "createdDate": "2014-02-21 12:26:49",
      "wfId": 6047,
      "overwritable": false,
      "globalParameter": "Hive_Test,,,,,,,,",
      "wfNamespaceList": [],
      "stepRestart": false,
      "categoryId": 1,
      "wfName": "Hive_Test99",
      "stepList": [
        {
          "description": null,
          "actionId": 1,
          "stepId": 21831,
          "yesStepId": 0,
          "noStepId": 0,
          "pairConnectionId": 0,
          "stepConnectionList": [
            {
              "nextId": 21833,
              "stepConnectionId": 14898,
              "lineCoordinates": ""
            }
          ],
          "stepParamList": [],
          "subWorkFlow": null,
          "stepName": "Start",
          "stepSeverity": "",

```

```

        "subWorkflowId": 0,
        "wfActionVO": null,
        "stepType": null,
        "shapeCoordinates": "172,71",
        "stepNotification": "N",
        "isAlreadyExecuted": "",
        "stepSubList": null
    },
    {
        "description": null,
        "actionId": 5,
        "stepId": 21832,
        "yesStepId": 0,
        "noStepId": 0,
        "pairConnectionId": 0,
        "stepConnectionList": [],
        "stepParamList": [],
        "subWorkFlow": null,
        "stepName": "Stop",
        "stepSeverity": "",
        "subWorkflowId": 0,
        "wfActionVO": null,
        "stepType": null,
        "shapeCoordinates": "430,229",
        "stepNotification": "N",
        "isAlreadyExecuted": "",
        "stepSubList": null
    },
    {
        "description": null,
        "actionId": 7,
        "stepId": 21833,
        "yesStepId": 0,
        "noStepId": 0,
        "pairConnectionId": 0,
        "stepConnectionList": [
            {
                "nextId": 21832,
                "stepConnectionId": 14899,
                "lineCoordinates": ""
            }
        ],
        "stepParamList": [
            {
                "key": "hiveQuery",
                "valueText": "show databases;
",
                "stepParamId": 38333
            },
            {
                "key": "hiveFilePath",
                "valueText": "",
                "stepParamId": 38334
            },
            {
                "key": "description",
                "valueText": "",
                "stepParamId": 38335
            }
        ]
    }
],
"stepParamList": [
    {
        "key": "hiveQuery",
        "valueText": "show databases;
",
        "stepParamId": 38333
    },
    {
        "key": "hiveFilePath",
        "valueText": "",
        "stepParamId": 38334
    },
    {
        "key": "description",
        "valueText": "",
        "stepParamId": 38335
    }
]

```

```

    },
    "subWorkFlow": null,
    "stepName": "H1",
    "stepSeverity": "",
    "subWorkflowId": 0,
    "wfActionVO": null,
    "stepType": null,
    "shapeCoordinates": "273,139",
    "stepNotification": "N",
    "isAlreadyExecuted": "",
    "stepSubList": null
  },
  ],
  "modifiedDate": "2014-04-28 08:54:24",
  "instanceVersion": 0,
  "wfType": "GEN",
  "wfLevelParameterList": [],
  "bedrockInstanceId": 0,
  "modifiedBy": "john",
  "categoryName": null,
  "bedrockURL": null,
  "wfSeverity": "",
  "executedBy": null,
  "listCounterMap": {},
  "wfNotification": "N",
  "wfStepNotification": "N"
}
]
}

```

Example Response:

```

{
  "result": null,
  "status": {
    "responseCode": 200,
    "result": null,
    "responseMessage": " Workflow 'Hive_Test99' imported successfully",
    "responseType": "INFO"
  }
}

```

Response in Case of Failure:

- Scenario 1:

If a workflow already exists with the given workflow name:

```

{
  "responseMessage": " Workflow name 'watermarked134' already exist,
  ↳ import skipped ",
  "restUri": "/bedrock-app/services/rest/workflows/import",
  "result": null,
  "page": null
}

```

- Scenario 2:

If JSON is edited and made invalid:

```
{
    "responseMessage": "Error while importing watermarked134",
    "restUri": "/bedrock-app/services/rest/workflows/import",
    "result": null,
    "page": null
}
```

1.5.28 Fetch List of Instance Logs

This API allows you to fetch all the Instance logs for a particular workflow under a specific project.

URL:

```
http://<bedrock-host:port>/bedrock-app/services/rest/workflows/{workflow_Id}/
instances/{instance_Id}/logs?projectIds={project_Id}
```

Note: You must pass the *workflow_Id*, *instance_Id*, and *project_Id* parameters in the API URL. To fetch all the workflow instance logs under the public project, set the *project_Id* parameter to *1*.

Method:

POST

Request Payload:

```
{
    "chunkSize": "<This is the number of records that must be fetched per page.␣
↳When set as '0' (or unspecified), all results are fetched.>",
    "pageNo": "<This is the current page required for paginating the request␣
↳depending on the total records and chunkSize.>",
}
```

Example Request:

```
{
    "chunkSize": 10,
    "pageNo": 1
}
```

Example Response:

```
{
    "responseMessage": null,
    "restUri": null,
    "result": {
        "currentPage": 1,
        "totalRecords": 2,
        "list": [
            {
                "type": "FILE",
                "value": "/projects/zaloni/Bedrock_44/bedrock/bedrock-
↳app/logs/WORKFLOW/fm_hdfstohdfs/107/fm_hdfstohdfs-107-DEBUG.log",
                "logFileName": "fm_hdfstohdfs-107-DEBUG.log"
            }
        ]
    }
}
```

```

        },
        {
            "type": "FILE",
            "value": "/projects/zaloni/Bedrock_44/bedrock/bedrock-
↪app/logs/WORKFLOW/fm_hdfstohdfs/107/FM_Output_32_107_100.log",
            "logFileName": "FM_Output_32_107_100.log"
        }
    ]
},
"page": null
}

```

Response in Case of Failure:

- Scenario 1:

If the workflow does not exist for the specific instance:

```

{
    "responseMessage": "Workflow id 5 does not exist for this instance",
    "restUri": null,
    "result": null,
    "page": null
}

```

- Scenario 2:

If the workflow cannot be accessed from the specified project:

```

{
    "responseMessage": "Artifact with Id 52 and type WORKFLOW cannot be_
↪accessed from project with Id 1",
    "restUri": "/bedrock-app/services/rest/workflows/52/instances/161/logs",
    "result": null,
    "page": null
}

```

1.5.29 Download Workflow Log File

This API allows you to download the workflow log file for a specific workflow under the project.

URL:

```

http://<bedrock-host:port>/bedrock-app/services/rest/workflow/{workflow_Id}/instance/
↪{instance_Id}/download/{log_file_name}?projectIds={project_Id}

```

Before using this service, ensure to note the following points:

- You must pass the *workflow_Id*, *instance_Id*, and *project_Id* parameters in the API URL.
- To get the Workflow Id, you can use the [Search and Fetch Workflows](#) API
- To retrieve the Instance Id of the workflow (whose logs need to be downloaded), refer to the [Fetch Workflow Execution History](#) API.
- To get the *log_file_name* that needs to be passed in the url, use the [Fetch List of Instance Logs](#) API.
- To download the workflow logs for a workflow under the public project, set the *project_Id* parameter to *1*.

Method:

GET

Example Response:

The log file is displayed.

Response in Case of Failure:

If no workflow exists for the specified instance:

```
{
  "responseMessage": "Workflow id 5 does not exist for this instance",
  "restUri": "/bedrock-app/services/rest/workflow/5/instance/1/download/wf_
↪hiveAction1-16-DEBUG.log",
  "result": null,
  "page": null
}
```

1.5.30 List Transformation

This API allows you to view all the transformations under the specified project.

URL:

```
http://<bedrock-host:port>/bedrock-app/services/rest/transformation/mappings/search?
↪projectIds={project_Id}
```

Note: You must pass the *project_Id* parameter in the API URL. To fetch the transformations under the public project, set the *project_Id* parameter to *1*.

Method:

POST

Request Payload:

You can pass anyone of the following parameters to fetch the desired output:

```
{
  "chunkSize": "<This is the number of records required for each service_
↪request. Set this parameter to a value to get specific number of the records per_
↪page. When set as '0' (or unspecified), all results are fetched.>",
  "currentPage": "<This is the current page required for paginating the request_
↪depending on the total records and chunkSize.>",
  "searchCriteria": [
    {
      "searchMode": "<This is the type of search that must be_
↪performed by using the value set for the searchValue parameter. Set LIKE to perform_
↪a like search, EXACT to match the exact value, NOT to perform an invert search, or_
↪RANGE to perform a search using a defined range.>",
      "searchAttribute": "<This is the name of the attribute based_
↪on which the search is performed. Leave it blank if not applicable. A probable_
↪value is trMappingName.>",
      "searchValue": "<This is the value for which search is_
↪performed.>"
    }
  ]
}
```

```
    ],
    "sortBy": "<This is the field name over which the response result will be_
↪sorted. The default sorting is on the modified date and hence, the latest workflow_
↪execution instance comes on the top.>"
    "orderBy": "<This is the parameter by which you want to order the_
↪transformations.>"
}
```

Example Request:

```
{
  "chunkSize": "50",
  "currentPage": 1,
  "searchCriteria": [
    {
      "searchMode": "LIKE",
      "searchAttribute": "trMappingName",
      "searchValue": ""
    }
  ],
  "sortBy": "trModifiedDate",
  "orderBy": "DESC"
}
```

Example Response:

```
{
  "responseMessage": "info",
  "restUri": "/bedrock-app/services/rest/transformation/mappings/search",
  "result": {
    "resultList": [
      {
        "trMappingId": 3,
        "trMappingName": "trans2_projection_08102016",
        "trStartingNodeId": 13,
        "trCreatedBy": "john",
        "trCreatedDate": 1470816491000,
        "trCreatedDateString": "08/10/2016 13:38:11",
        "trModifiedBy": "john",
        "trModifiedDate": 1470816491000,
        "trModifiedDateString": "08/10/2016 13:38:11",
        "isEditable": true,
        "trWfId": 8,
        "trWfName": "trans2_projection_08102016"
      },
      {
        "trMappingId": 2,
        "trMappingName": "clone1",
        "trStartingNodeId": 9,
        "trCreatedBy": "admin",
        "trCreatedDate": 1470813290000,
        "trCreatedDateString": "08/10/2016 12:44:50",
        "trModifiedBy": "admin",
        "trModifiedDate": 1470813290000,
        "trModifiedDateString": "08/10/2016 12:44:50",
        "isEditable": true,
        "trWfId": 7,
        "trWfName": "clone1"
      }
    ]
  }
}
```

```

    },
    {
      "trMappingId": 1,
      "trMappingName": "trans1_john_08102016",
      "trStartingNodeId": 5,
      "trCreatedBy": "john",
      "trCreatedDate": 1470812000000,
      "trCreatedDateString": "08/10/2016 12:23:20",
      "trModifiedBy": "admin",
      "trModifiedDate": 1470813269000,
      "trModifiedDateString": "08/10/2016 12:44:29",
      "isEditable": true,
      "trWfId": 6,
      "trWfName": "trans1_john_08102016"
    }
  ],
  "totalRecords": 3,
  "currentPage": 1,
  "chunkSize": 0
},
"page": null
}

```

Response in Case of Failure:

If no transformation is present:

```
no content
```

1.5.31 Create User-Defined Action

This is a preview feature.

This API allows you to create custom action.

URL:

```
http://<bedrock-host:port>/bedrock-app/services/rest/workflows/userDefinedActions
```

Method:

```
POST
```

Request Payload:

```

{
  "userDefinedActionId": "<This is the user-defined action Id. You can use_
↪zero for creating new UDA. Alternatively, you may not pass any value.>"
  * "userDefinedActionBusinessName": "<This is the business name of the user-
↪defined action.>",
  * "userDefinedActionTechnicalName": "<This is the technical name of the user-
↪defined action.>",
  "description": "<This is the description of the user-defined action.>",
  "createdBy": "<This is the user who creates the action.>",
  "createdDate": "<This is the time that has elapsed since the Unix Epoch, in_
↪seconds, till the time when the action was created.>",
  "modifiedBy": "<This is the user who updates the user-defined action.>",

```

```

    "modifiedDate": "<This is the time that has elapsed since the Uniz Epoch,
↳in seconds, till the time when the action was updated.>",
    * "status": "<This is the status of user-defined action. The possible values
↳are designed, deployed, or deprecated.>",
    "wfId": "<This is the Id of the workflow where you want to use the user-
↳defined action.>",
    "wfName": "<This is the name of the workflow where you want to use the user-
↳defined action.>",
    * "wfTarPath": "<This is the absolute path of the tar file uploaded into the
↳bedrock-app server. It is generated by exporting a single workflow to be used as a
↳template for the user-defined action.>",
    "wfDetails": "<This is the description for the workflow.>",
    * "userDefinedParamsMapping": "<This is the serialized JSON string of the
↳user-defined action param mapping object. For more information, refer to the
↳Details about the userDefinedParamsMapping parameter section.>",
    "actionIcon": "<This is the icon of the action.>",
    "actionColor": "<This is the color of the action icon.>",
    "actionHaveConsoleOutput": "<This indicates if you want to have console
↳output.>",
    * "wfUserDefinedActionParamsList":
    [
        {
            "paramKey": "<This is the technical name of the user-defined
↳action parameter.>",
            "paramLabel": "<This is the business name of the user-
↳defined parameter.>",
            "paramOrder": "<This is the order to be used to display the
↳user-defined action parameter.>",
            "paramDataType": "<This is the data type of the parameter.
↳Possible values for this parameter are listed in a separate section below.>",
            "paramRegex": "<This is the regular expression for
↳validating text input of this user-defined action parameter.>",
            "paramHelperText": "<This is the helper text to be used for
↳this user-defined action parameter.>",
            "paramMaxChar": "<This is the maximum number of character to
↳be allowed in case of text input of this user-defined action parameter.>",
            "defaultValue": "<This is the default value to be used.>",
            "paramToolTipDisplay": "<This is the boolean value
↳indicating if tooltip need to be displayed for this user-defined action.>",
            "required": "<This indicates if the parameter is required or
↳not. 1 and 0 indicates if the parameter is required and not required respectively.>"
        }
    ]
}

```

Details about the userDefinedParamsMapping parameter:

You must send the serialized version of the below JSON for the *userDefinedParamsMapping* parameter, in the request payload.

```

{
    "stepParamsMapping":
    [
        {
            "stepName": "<This is the user-defined step name of the step in the template
↳workflow, which you want to overwrite.>",
            "stepParams":
            [{

```

```

        "key": "<This is the technical name of the step parameter of the
↪ workflow step.>",
        "valueText": "<This is the value that needs to be passed to overwrite
↪ the step-parameter value of the template workflow. ${UDA.technicalNameOfTheUDAParam}
↪ can be used here.>"
    }}
  },
  "namespaceMapping":
  [{
    "namespaceVariable": "<This is the namespace identifier that you want to
↪ overwrite.>",
    "userDefinedParam": "<This is the value for the namespace identifier that
↪ you want to overwrite. ${UDA.technicalNameOfTheUDAParam} can be used here.>"
  }]
}

```

Possible values for the paramDataType parameter:

```

emailBean.from, emailBean.to, emailBean.strSubject, emailBean.strBody, emailBean.
↪ description, emailBean.proceedOnFailure, fileMoveAction.
↪ isDistributedFileMoveEnabled, fileMoveAction.isManifestFile, fileMoveAction.
↪ srcDesPairs, fileMoveAction.regex, fileMoveAction.description, fileMoveAction.
↪ proceedOnFailure, shellAction.scriptContent, shellAction.scriptFile, shellAction.
↪ strArgument, shellAction.environmentParams, shellAction.description, shellAction.
↪ executeOnCluster, shellAction.proceedOnFailure, customJava.mainClass, customJava.
↪ jarNames, customJava.classPathw, customJava.javaOpts, customJava.strArgs,
↪ customJava.description, customJava.proceedOnFailure, hiveAction.hiveQuery,
↪ hiveAction.hiveFilePath, hiveAction.hiveQueryFileNode, hiveAction.hiveVar,
↪ hiveAction.hiveConfVar,

```

```

hiveAction.description, hiveAction.proceedOnFailure, ifAction.Conditional, ifAction.
↪ description, mapReduceAction.jarFilePath, mapReduceAction.driverClass,
↪ mapReduceAction.stringArguments, mapReduceAction.classPath, mapReduceAction.
↪ description, mapReduceAction.extraJobConfig, mapReduceAction.proceedOnFailure,
↪ subWorkflowAction.workflowName, subWorkflowAction.workflowId, subWorkflowAction.
↪ overrideGlobalParam, subWorkflowAction.overrideWfLevelParam, subWorkflowAction.
↪ description, subWorkflowAction.overrideWfNamespace, subWorkflowAction.
↪ proceedOnFailure, waterMarkAction.dfsInputPath, waterMarkAction.dfsOutputPath,
↪ waterMarkAction.entityTypeAndVersion, waterMarkAction.numReduceTaskWatermark,
↪ waterMarkAction.effectiveDate, waterMarkAction.dataFormatWatermark, waterMarkAction.
↪ fieldDelim, waterMarkAction.description, waterMarkAction.extraJobConfig,
↪ waterMarkAction.proceedOnFailure, lookupAction.dataFilePath,

```

```

lookupAction.dataFileEntityType, lookupAction.lookupEntityInstanceIdVersion,
↪ lookupAction.outputFileLocation, lookupAction.numReduceTaskLookup, lookupAction.
↪ dataFileDelimiter, lookupAction.description, lookupAction.proceedOnFailure,
↪ tokenMaskingAction.entityTypeIdAndVersion, tokenMaskingAction.inputDirectory,
↪ tokenMaskingAction.storeOutputDirectory, tokenMaskingAction.secureVault,
↪ tokenMaskingAction.secureOutputDirectory, tokenMaskingAction.vaultArguments,
↪ tokenMaskingAction.numReduceTokenMasking, tokenMaskingAction.description,
↪ tokenMaskingAction.fieldDelimiterTokenMasking, tokenMaskingAction.extraJobConfig,
↪ tokenMaskingAction.proceedOnFailure, fileWatchAction.pathsFileWatch,
↪ fileWatchAction.delayFileWatch, fileWatchAction.timeOutFileWatch, fileWatchAction.
↪ description, fileWatchAction.proceedOnFailure, dataQualityAction.
↪ entityTypeAndVersion, dataQualityAction.fieldDelimiterDQ, dataQualityAction.
↪ inputFromEntity, dataQualityAction.dataFilePath,

```

```

dataQualityAction.partitionSpec, dataQualityAction.description, dataQualityAction.
→validDataPath, dataQualityAction.badDataOPLocation, dataQualityAction.
→invalidDataPath, dataQualityAction.esReportRequired, dataQualityAction.
→createGoodDataEntity, dataQualityAction.createBadDataEntity, dataQualityAction.
→proceedOnFailure, cdcAction.hdfsInputPath, cdcAction.entityTypeAndVersion,
→cdcAction.fieldDelimiterCDC, cdcAction.deleteFilePath, cdcAction.deleteFileSuffix,
→cdcAction.cdcTempPath, cdcAction.archivePath, cdcAction.cdcBadRecords, cdcAction.
→errorThresholdValue, cdcAction.noOfInputRecords, cdcAction.falsePositiveRate,
→cdcAction.skipEmptyRecordsFlag, cdcAction.description, cdcAction.extraJobConfig,
→cdcAction.proceedOnFailure, sparkAction.classPath, sparkAction.master, sparkAction.
→deployMode, sparkAction.conf, sparkAction.jarFilePath, sparkAction.arguments,
→sparkAction.sparkOptions, sparkAction.description, sparkAction.extraJobConfig,
→sparkAction.proceedOnFailure, sparkSQLAction.sql, sparkSQLAction.description,
→sparkSQLAction.master, sparkSQLAction.deployMode, sparkSQLAction.sparkOptions,
→sparkSQLAction.extraJobConfig, sparkSQLAction.proceedOnFailure, entityLevelDQAction.
→dataFilePath,

```

```

entityLevelDQAction.entityTypeAndVersion, entityLevelDQAction.badRecordPath,
→entityLevelDQAction.badRecordThresholdCount, entityLevelDQAction.description,
→entityLevelDQAction.extraJobConfig, entityLevelDQAction.proceedOnFailure,
→loadPartitionAction.loadPartitionsBy, loadPartitionAction.dataFilePath,
→loadPartitionAction.entityTypeAndVersion, loadPartitionAction.description,
→loadPartitionAction.partitionKeyValuePairs, loadPartitionAction.overwriteData,
→loadPartitionAction.hiveVar, loadPartitionAction.hiveConfVar, loadPartitionAction.
→proceedOnFailure, fileEncodeAction.sourcePath, fileEncodeAction.destinationPath,
→fileEncodeAction.sourceEncoding, fileEncodeAction.numReducers, fileEncodeAction.
→extraJobConfig, fileEncodeAction.proceedOnFailure, qeAction.queryName, qeAction.
→outputDirectory, qeAction.headerRequired, qeAction.delimiter, qeAction.description,
→qeAction.proceedOnFailure, textToAvroConversionAction.entityTypeAndVersion,
→textToAvroConversionAction.targetEntityName, textToAvroConversionAction.
→dataFilePath, textToAvroConversionAction.targetDataPath,

```

```

textToAvroConversionAction.targetFieldSchemaPath, textToAvroConversionAction.
→targetSchema, textToAvroConversionAction.actionDescription,
→textToAvroConversionAction.proceedOnFailure, sqoopImportAction.optionsFile,
→sqoopImportAction.arguments, sqoopImportAction.targetDir, sqoopImportAction.
→chooseEnt, sqoopImportAction.description, sqoopImportAction.proceedOnFailure,
→sqoopExportAction.optionsFile, sqoopExportAction.arguments, sqoopExportAction.
→description, sqoopExportAction.proceedOnFailure, sparkTransformationAction.
→description, sparkTransformationAction.mapping, sparkTransformationAction.
→sparkOptions, sparkTransformationAction.extraJobConfig, sparkTransformationAction.
→proceedOnFailure, dataInventoryAction.userInputDirectories, dataInventoryAction.
→exclusionFilters, dataInventoryAction.profileName, dataInventoryAction.scanTillDate,
→dataInventoryAction.description, dataInventoryAction.extraJobConfig,
→dataInventoryAction.proceedOnFailure, textToParquetConversionAction.
→entityTypeAndVersion, textToParquetConversionAction.targetEntityName,

```

```

textToParquetConversionAction.dataFilePath, textToParquetConversionAction.
→targetDataPath, textToParquetConversionAction.targetSchema,
→textToParquetConversionAction.actionDescription, textToParquetConversionAction.
→proceedOnFailure, xsltAction.inputFS, xsltAction.input, xsltAction.inputxslt,
→xsltAction.outputFS, xsltAction.outputExt, xsltAction.outputloc, xsltAction.
→description, xsltAction.proceedOnFailure, emlAction.inputFS, emlAction.
→localInputPath, emlAction.outputFS, emlAction.hdfsOutputPath, emlAction.description,
→emlAction.proceedOnFailure, restAction.httpMethod, restAction.url, restAction.
→header, restAction.payload, restAction.outputLocation, restAction.
→headerInfoRequired, restAction.description, restAction.proceedOnFailure, ftpAction.
→operation, ftpAction.protocol, ftpAction.host, ftpAction.port, ftpAction.uname,
→ftpAction.passwd, ftpAction.keyFile, ftpAction.remoteDir, ftpAction.localDir,
→ftpAction.fileFolder, ftpAction.globOp, ftpAction.continueOp, ftpAction.noRetries,
→ftpAction.fileSystem, ftpAction.uploadSrcURI, ftpAction.deleteSourceAfterTransfer,
→ftpAction.overwriteExisting, ftpAction.description, ftpAction.proceedOnFailure,

```

```
dataProfilingAction.timeOfLastUpdated, dataProfilingAction.selectEntityTypes, ↵
↵dataProfilingAction.description, dataProfilingAction.extraJobConfig, ↵
↵dataProfilingAction.incremental, dataProfilingAction.proceedOnFailure, pythonAction.
↵strCommand, pythonAction.scriptPath, pythonAction.strArgument, pythonAction.
↵environmentParams, pythonAction.description, pythonAction.executeOnCluster, ↵
↵pythonAction.proceedOnFailure, pySparkAction.master, pySparkAction.deployMode, ↵
↵pySparkAction.conf, pySparkAction.jarFilePath, pySparkAction.arguments, ↵
↵pySparkAction.sparkOptions, pySparkAction.description, pySparkAction.extraJobConfig,
↵ pySparkAction.proceedOnFailure, dbImportAction.connectionName, dbImportAction.
↵sourcePlatform, dbImportAction.sourceSchema, dbImportAction.userName, ↵
↵dbImportAction.userPassword, dbImportAction.testConnection, dbImportAction.
↵tableName, dbImportAction.createBedrockEnt, dbImportAction.targetSchema, ↵
↵dbImportAction.targetDirPath, dbImportAction.additionalOptions,
```

```
dbImportAction.dbimportAdditionalOptions, dbImportAction.description, dbImportAction.
↵proceedOnFailure, vsamParseAction.inputFilePath, vsamParseAction.
↵entityTypeAndVersion, vsamParseAction.fileEncodingType, vsamParseAction.charSet, ↵
↵vsamParseAction.fileFormat, vsamParseAction.description, vsamParseAction.
↵skipErrorRecords, vsamParseAction.extraJobConfig, vsamParseAction.proceedOnFailure, ↵
↵hiveProjection.sourceEntityTypeAndVersion, hiveProjection.providedDataPath, ↵
↵hiveProjection.sourceFilter, hiveProjection.targetEntityTypeAndVersion, ↵
↵hiveProjection.fieldKeyValuePairs, hiveProjection.partitionKeyValuePairs, ↵
↵hiveProjection.overwrite, hiveProjection.hiveVar, hiveProjection.hiveConfVar, ↵
↵hiveProjection.description, hiveProjection.proceedOnFailure, hiveInventoryAction.
↵hiveDatabase, hiveInventoryAction.tablePattern, hiveInventoryAction.
↵additionalOptions, hiveInventoryAction.description, hiveInventoryAction.
↵proceedOnFailure, xmlConverterAction.inputFilePath, xmlConverterAction.
↵entityTypeAndVersion, xmlConverterAction.rootTag, xmlConverterAction.description, ↵
↵xmlConverterAction.proceedOnFailure
```

Example Request:

```
{
  "userDefinedActionId": 0,
  "userDefinedActionBusinessName": "HiveSample10",
  "userDefinedActionTechnicalName": "HiveSample10",
  "description": "Run hive query",
  "status": "designed",
  "wfTarPath": "/home/hudson/rupjit/UDAtars/workflow-20170726144012584.tar",
  "userDefinedParamsMapping": "{\"stepParamsMapping\": [{\"stepName\": \"wff\", ↵
↵ \"stepParams\": [{\"key\": \"hiveQuery\", \"valueText\": \"${UDA.HiveQuery}\"}]}], ↵
↵ \"namespaceMapping\": [{\"namespaceVariable\": \"WORKFLOW.some_key1\", ↵
↵ \"userDefinedParam\": \"hard coded value\"}]}",
  "wfUserDefinedActionParamsList": [{
    "paramKey": "HiveQuery",
    "paramLabel": "Hive Query",
    "paramOrder": 1,
    "paramDataType": "hiveAction.hiveQuery",
    "paramRegex": "",
    "paramHelperText": "Enter hive Query",
    "paramMaxChar": null,
    "defaultValue": "Select * from bedrock.abc",
    "paramToolTipDisplay": true,
    "required": 1
  }]
}
```

Example Response:

```
{
  "responseMessage": "The user defined action: HiveSample10 saved successfully.",
  "restUri": "/bedrock-app/services/rest/workflows/userDefinedActions",
  "result": {
    "userDefinedActionId": 31,
    "userDefinedActionBusinessName": "HiveSample10",
    "userDefinedActionTechnicalName": "HiveSample10",
    "description": "Run hive query",
    "createdBy": "admin",
    "createdDate": 1501247383000,
    "modifiedBy": "admin",
    "modifiedDate": 1501247383000,
    "status": "designed",
    "actionId": 0,
    "wfId": 0,
    "wfName": "uda_a719ddd1_9c99_405f_9ada_257504eb1975",
    "wfTarPath": "/home/bedrock/rupjit/UDA/workflow-20170728183827595.tar",
    "wfDetails": "{\n\"wfId\":0,\n\"categoryId\":1,\n\"wfName\":\n\"uda_a719ddd1_9c99_405f_9ada_257504eb1975\",\n\"createdBy\":\n\"admin\",\n\"createdDate\":\n\"07/28/2017 18:39:43\",\n\"modifiedBy\":\n\"admin\",\n\"modifiedDate\":\n\"07/28/2017 18:39:43\",\n\"wfSeverity\":\n\"\",\n\"wfNotification\":\n\"N\",\n\"wfStepNotification\":\n\"N\",\n\"notificationDetails\":\n[],\n\"globalParameter\":\n\"\",\n\"stepList\":\n[\n{\n\"stepId\":0,\n\"actionId\":1,\n\"stepName\":\n\"Start\",\n\"shapeCoordinates\":\n\"303,138\",\n\"stepSeverity\":\n\"\",\n\"stepNotification\":\n\"N\",\n\"yesStepId\":0,\n\"yesStepName\":\nnull,\n\"noStepId\":0,\n\"noStepName\":\nnull,\n\"pairConnectionId\":0,\n\"pairConnectionName\":\nnull,\n\"stepParamList\":\n[],\n\"stepConnectionList\":\n[\n{\n\"stepConnectionId\":0,\n\"nextId\":0,\n\"nextStepName\":\n\"s1\"\n}\n],\n\"isAlreadyExecuted\":\nfalse,\n\"stepSubList\":\nnull,\n\"stepType\":\n\"Start\",\n\"subWorkflowId\":0,\n\"subWorkflow\":\nnull,\n\"description\":\nnull,\n\"skipped\":\nfalse,\n{\n\"stepId\":0,\n\"actionId\":5,\n\"stepName\":\n\"Stop\",\n\"shapeCoordinates\":\n\"673,190\",\n\"stepSeverity\":\n\"\",\n\"stepNotification\":\n\"N\",\n\"yesStepId\":0,\n\"yesStepName\":\nnull,\n\"noStepId\":0,\n\"noStepName\":\nnull,\n\"pairConnectionId\":0,\n\"pairConnectionName\":\nnull,\n\"stepParamList\":\n[],\n\"stepConnectionList\":\n[],\n\"isAlreadyExecuted\":\nfalse,\n\"stepSubList\":\nnull,\n\"stepType\":\n\"Stop\",\n\"subWorkflowId\":0,\n\"subWorkflow\":\nnull,\n\"description\":\nnull,\n\"skipped\":\nfalse,\n{\n\"stepId\":0,\n\"actionId\":3,\n\"stepName\":\n\"s1\",\n\"shapeCoordinates\":\n\"504,164\",\n\"stepSeverity\":\n\"\",\n\"stepNotification\":\n\"N\",\n\"yesStepId\":0,\n\"yesStepName\":\nnull,\n\"noStepId\":0,\n\"noStepName\":\nnull,\n\"pairConnectionId\":0,\n\"pairConnectionName\":\nnull,\n\"stepParamList\":\n[\n{\n\"stepParamId\":0,\n\"key\":\n\"description\",\n\"valueText\":\nnull\n},\n{\n\"stepParamId\":0,\n\"key\":\n\"srcDesPairs\",\n\"valueText\":\n\"/user/bedrock/Jeet/mInput/manifest;HDFS;/user/bedrock/himangshu/fm/out10;HDFS;true;3;3\"\n},\n{\n\"stepParamId\":0,\n\"key\":\n\"proceedOnFailure\",\n\"valueText\":\n\"false\"\n},\n{\n\"stepParamId\":0,\n\"key\":\n\"regex\",\n\"valueText\":\n\"false\"\n},\n{\n\"stepParamId\":0,\n\"key\":\n\"isManifestFile\",\n\"valueText\":\n\"true\"\n},\n{\n\"stepParamId\":0,\n\"key\":\n\"isDistributedFileMoveEnabled\",\n\"valueText\":\n\"true\"\n}\n],\n\"stepConnectionList\":\n[\n{\n\"stepConnectionId\":0,\n\"nextId\":0,\n\"nextStepName\":\n\"Stop\"\n}\n],\n\"isAlreadyExecuted\":\nfalse,\n\"stepSubList\":\nnull,\n\"stepType\":\n\"File Move Action\",\n\"subWorkflowId\":0,\n\"subWorkflow\":\nnull,\n\"description\":\nnull,\n\"skipped\":\nfalse,\n{\n\"bedrockInstanceId\":0,\n\"executedBy\":\nnull,\n\"scheduledBy\":\nnull,\n\"wfLevelParameterList\":\n[],\n\"wfType\":\n\"UDAWORKFLOW\",\n\"listCounterMap\":\n{\n},\n\"instanceVersion\":0,\n\"categoryName\":\n\"unassigned\",\n\"bedrockURL\":\nnull,\n\"wfNamespaceList\":\n[],\n\"description\":\n\"\",\n\"adminCapacityQueues\":\nnull,\n\"logLevel\":\n\"DEBUG\",\n\"baseEncodedString\":\nnull,\n\"overwritable\":\ntrue,\n\"clusterId\":\nnull,\n\"executionPriority\":\nnull,\n\"stepRestart\":\nfalse\n}\n}",
    "userDefinedParamsMapping": "{\n\"stepParamsMapping\":\n[\n{\n\"stepName\":\n\"wff\",\n\"stepParams\":\n[\n{\n\"key\":\n\"hiveQuery\",\n\"valueText\":\n\"${UDA.HiveQuery}\"\n}\n]\n},\n{\n\"namespaceMapping\":\n[\n{\n\"namespaceVariable\":\n\"WORKFLOW.some_key1\",\n\"userDefinedParam\":\n\"hard coded value\"\n}\n]\n}\n]"
  }
}
```

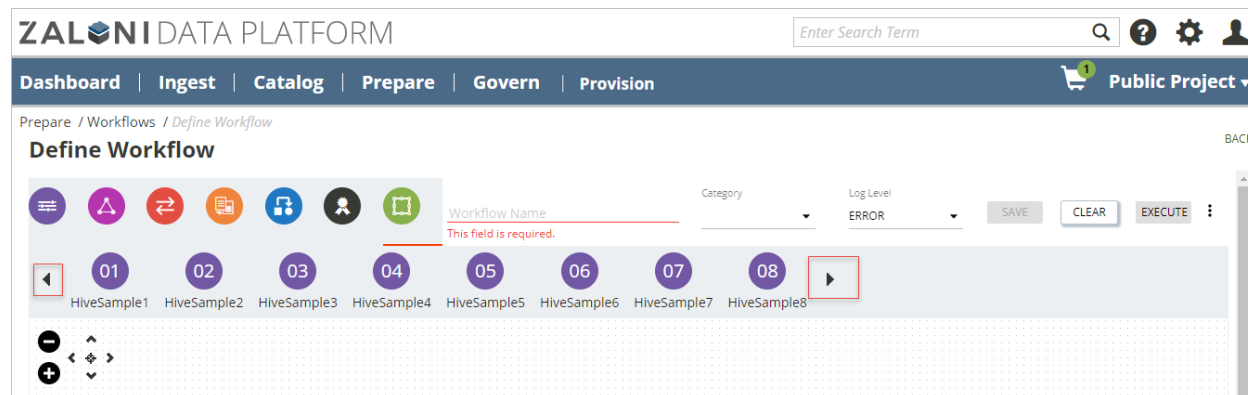


```

    "actionIcon": null,
    "actionColor": null,
    "actionHaveConsoleOutput": null,
    "wfUserDefinedActionParamsList": [
      {
        "userDefinedActionParamId": 24,
        "userDefinedActionId": 31,
        "paramKey": "HiveQuery",
        "paramLabel": "Hive Query",
        "paramOrder": 1,
        "paramDataType": "hiveAction.hiveQuery",
        "paramRegex": "",
        "paramHelperText": "Enter hive Query",
        "paramMaxChar": null,
        "paramFieldControl": null,
        "paramFieldControlOptions": null,
        "errorMessage": null,
        "defaultValue": "Select * from bedrock.abc",
        "paramValue": null,
        "paramToolTipDisplay": true,
        "required": 1,
        "lineageType": null
      }
    ],
    "page": null
  }
}

```

Once the user-defined action is created, you can view the action in the Define Workflow page and leverage the same, for creating workflows.



ZDP displays a maximum of 8 user-defined actions in the horizontal pane. If you create more than 8 user-defined actions, you can navigate to the user-defined actions (which are not displayed), by using the > or < icons.

ZALONI DATA PLATFORM Enter Search Term ? ⚙️ 👤

Dashboard | Ingest | Catalog | Prepare | Govern | Provision 🛒 1 **Public Project**

Prepare / Workflows / Define Workflow BACK

Define Workflow

Workflow Name This field is required. Category Log Level ⋮

01 HiveSample1

Step Name

User Defined Action

Hive Query

Select * from bedrock.abd ☒ Use Hive Query

Description

☐ Proceed to Next Step on Failure

01 02 03

HiveSample1 HiveSample2 HiveSample

Workflow Diagram: A sequence of steps 01, 02, and 03. Step 01 (HiveSample1) is highlighted with a red circle and a play button icon. Step 02 is HiveSample2 and Step 03 is HiveSample.

Response in Case of Failure:

- Scenario 1:

If the tar file could not be located:

```
{
  "responseMessage": "User Defined Action template workflow not found in_
  ↳the tar file: /home/hudson/rupjit/UDAtars/workflow-hiveUDA.tar",
  "restUri": "/bedrock-app/services/rest/workflows/userDefinedActions",
  "result": null,
  "page": null
}
```

- Scenario 2:

If the tar file does not exist:

```
{
  "responseMessage": "User Defined Action template workflow tar File does_
  ↳not exists.",
  "restUri": "/bedrock-app/services/rest/workflows/userDefinedActions",
  "result": null,
  "page": null
}
```

- Scenario 3:

If the tar file contains multiple workflows:

```
{
  "responseMessage": "User Defined Action multiple template Workflows found_
  ↳in the tar file: /home/hudson/rupjit/UDAtars/workflow-20170726143526176.tar",
  "restUri": "/bedrock-app/services/rest/workflows/userDefinedActions",
  "result": null,
  "page": null
}
```

- Scenario 4:

If the action already exists:

```
{
  "responseMessage": "User defined action already exist with name ↪HiveSample5",
  "restUri": "/bedrock-app/services/rest/workflows/userDefinedActions",
  "result": null,
  "page": null
}
```

1.5.32 Fetch all the User-Defined Action Details

This is a preview feature.

This API allows you to fetch the details about the all the user-defined actions.

URL:

`http://<bedrock-host:port>/bedrock-app/services/rest/workflows/userDefinedActions/`
[↩ search](#)

Method:

POST

Example Response:

```

{
  "responseMessage": null,
  "restUri": "/bedrock-app/services/rest/workflows/userDefinedActions/search",
  "result": {
    "resultList": [
      {
        "userDefinedActionId": 1,
        "userDefinedActionBusinessName": "HiveSample9",
        "userDefinedActionTechnicalName": "HiveSample9",
        "description": "Run hive query",
        "createdBy": "admin",
        "createdDate": 1501060300000,
        "modifiedBy": "admin",
        "modifiedDate": 1501060300000,
        "status": "designed",
        "actionId": 0,
        "wfId": 0,
        "wfName": "uda_634d2b27_3fdc_43a8_a09e_465a04e73247",
        "wfTarPath": "/home/hudson/rupjit/UDAtars/workflow-
20170726144012584.tar",
        "wfDetails": {"wfId": 0, "categoryId": 1, "wfName":
"\uda_634d2b27_3fdc_43a8_a09e_465a04e73247", "createdBy": "admin",
"createdDate": "\07/26/2017 14:41:40", "modifiedBy": "admin", "modifiedDate":
"\07/26/2017 14:41:40", "wfSeverity": "", "wfNotification": "N",
"wfStepNotification": "N", "notificationDetails": [], "globalParameter": "",
, "\", "stepList": [{"stepId": 0, "actionId": 1, "stepName": "Start",
"shapeCoordinates": "\158,169", "stepSeverity": "", "stepNotification": "N",
"yesStepId": 0, "yesStepName": null, "noStepId": 0, "noStepName": null,
"pairConnectionId": 0, "pairConnectionName": null, "stepParamList": [],
"stepConnectionList": [{"stepConnectionId": 0, "nextId": 0, "nextStepName":
"asfads"}], "isAlreadyExecuted": "", "stepSubList": null, "stepType": "Start",
"subWorkflowId": 0, "subWorkflow": null, "description": null, "skipped": false}, {
"stepId": 0, "actionId": 5, "stepName": "Stop", "shapeCoordinates": "\649,221",
"stepSeverity": "", "stepNotification": "N", "yesStepId": 0, "yesStepName":
null, "noStepId": 0, "noStepName": null, "pairConnectionId": 0,
"pairConnectionName": null, "stepParamList": [], "stepConnectionList": [],

```

```

        "userDefinedParamsMapping": "{\\"stepParamsMapping\\":[
↪{\\"stepName\\":\\"wff\\",\\"stepParams\\":[{\\"key\\":\\"hiveQuery\\",\\"valueText\\":\\"${UDA.
↪HiveQuery}\\\"}]]},\\"namespaceMapping\\":[{\\"namespaceVariable\\":\\"WORKFLOW.some_key1\\
↪\",\\"userDefinedParam\\":\\"hard coded value\\\"}]]}",
        "actionIcon": null,
        "actionColor": null,
        "actionHaveConsoleOutput": null,
        "wfUserDefinedActionParamsList": [
            {
                "userDefinedActionParamId": 1,
                "userDefinedActionId": 1,
                "paramKey": "HiveQuery",
                "paramLabel": "Hive Query",
                "paramOrder": 1,
                "paramDataType": "hiveAction.hiveQuery

↪",
                "paramRegex": "",
                "paramHelperText": "Enter hive Query",
                "paramMaxChar": null,
                "paramFieldControl": null,
                "paramFieldControlOptions": null,
                "errorMessage": null,
                "defaultValue": "Select * from_

↪bedrock.abc",
                "paramValue": null,
                "paramToolTipDisplay": true,
                "required": 1,
                "lineageType": null
            }
        ]
    },
    {
        "userDefinedActionId": 2,
        "userDefinedActionBusinessName": "HiveSample1",
        "userDefinedActionTechnicalName": "HiveSample1",
        "description": "Run hive query",
        "createdBy": "admin",
        "createdDate": 1501060936000,
        "modifiedBy": "admin",
        "modifiedDate": 1501060936000,
        "status": "designed",
        "actionId": 0,
        "wfId": 0,
        "wfName": "uda_53cff665_7879_46a2_8df5_809d3c074aa1",
        "wfTarPath": "/home/hudson/rupjit/UDAtars/workflow-

↪20170726144012584.tar",
        "wfDetails": "{\\"wfId\\":0,\\"categoryId\\":1,\\"wfName\\
↪\\":\\"uda_53cff665_7879_46a2_8df5_809d3c074aa1\\",\\"createdBy\\":\\"admin\\",\
↪\"createdDate\\":\\"07/26/2017 14:52:16\\",\\"modifiedBy\\":\\"admin\\",\\"modifiedDate\\":\
↪\"07/26/2017 14:52:16\\",\\"wfSeverity\\":\\"\\",\\"wfNotification\\":\\"N\\",\
↪\"wfStepNotification\\":\\"N\\",\\"notificationDetails\\":[],\\"globalParameter\\":\\"\\",\
↪\",\\"stepList\\":[{\\"stepId\\":0,\\"actionId\\":1,\\"stepName\\":\\"Start\\",\
↪\"shapeCoordinates\\":\\"158,169\\",\\"stepSeverity\\":\\"\\",\\"stepNotification\\":\\"N\\",\
↪\"yesStepId\\":0,\\"yesStepName\\":null,\\"noStepId\\":0,\\"noStepName\\":null,\
↪\"pairConnectionId\\":0,\\"pairConnectionName\\":null,\\"stepParamList\\":[],\
↪\"stepConnectionList\\":[{\\"stepConnectionId\\":0,\\"nextId\\":0,\\"nextStepName\\":\
↪\"asfads\\\"}],\\"isAlreadyExecuted\\":\\"\\",\\"stepSubList\\":null,\\"stepType\\":\\"Start\\",\
↪\"subWorkflowId\\":0,\\"subWorkFlow\\":null,\\"description\\":null,\\"skipped\\":false},{\
↪\"stepId\\":0,\\"actionId\\":5,\\"stepName\\":\\"Stop\\",\\"shapeCoordinates\\":\\"649,221\\",\
↪\"stepSeverity\\":\\"\\",\\"stepNotification\\":\\"N\\",\\"yesStepId\\":0,\\"yesStepName\\
↪\\":null,\\"noStepId\\":0,\\"noStepName\\":null,\\"pairConnectionId\\":0,\
↪\"pairConnectionName\\":null,\\"stepParamList\\":[],\\"stepConnectionList\\":[],\
↪\"isAlreadyExecuted\\":\\"\\",\\"stepSubList\\":null,\\"stepType\\":\\"Stop\\",\

```

```

        "userDefinedParamsMapping": "{\\"stepParamsMapping\\":[
↪{\\"stepName\\":\\"wff\\",\\"stepParams\\":[{\\"key\\":\\"hiveQuery\\",\\"valueText\\":\\"${UDA.
↪HiveQuery}\\\"}]]},\\"namespaceMapping\\":[{\\"namespaceVariable\\":\\"WORKFLOW.some_key1\\
↪\",\\"userDefinedParam\\":\\"hard coded value\\\"}]]}",
        "actionIcon": null,
        "actionColor": null,
        "actionHaveConsoleOutput": null,
        "wfUserDefinedActionParamsList": [
            {
                "userDefinedActionParamId": 2,
                "userDefinedActionId": 2,
                "paramKey": "HiveQuery",
                "paramLabel": "Hive Query",
                "paramOrder": 1,
                "paramDataType": "hiveAction.hiveQuery

↪",
                "paramRegex": "",
                "paramHelperText": "Enter hive Query",
                "paramMaxChar": null,
                "paramFieldControl": null,
                "paramFieldControlOptions": null,
                "errorMessage": null,
                "defaultValue": "Select * from_

↪bedrock.abc",
                "paramValue": null,
                "paramToolTipDisplay": true,
                "required": 1,
                "lineageType": null
            }
        ]
    },
    "totalRecords": 15,
    "currentPage": 0,
    "chunkSize": 0
},
"page": null
}

```

1.5.33 Fetch the Specific User-Defined Action Details

This is a preview feature.

This API allows you to fetch the details about the specific user-defined action.

URL:

```

http://<bedrock-host:port>/bedrock-app/services/rest/workflows/userDefinedActions/
↪{user_defined_action_Id}

```

Note: You must pass the value for the `user_defined_action_Id` parameter in the API URL.

Method:

GET

Example Response:

```
{
  "responseMessage": null,
  "restUri": "/bedrock-app/services/rest/workflows/userDefinedActions/1",
  "result": {
    "userDefinedActionId": 1,
    "userDefinedActionBusinessName": "HiveSample9",
    "userDefinedActionTechnicalName": "HiveSample9",
    "description": "Run hive query",
    "createdBy": "admin",
    "createdDate": 1501060300000,
    "modifiedBy": "admin",
    "modifiedDate": 1501060300000,
    "status": "designed",
    "actionId": 0,
    "wfId": 0,
    "wfName": "uda_634d2b27_3fdc_43a8_a09e_465a04e73247",
    "wfTarPath": "/home/hudson/rupjit/UDAtars/workflow-20170726144012584.
    tar",
    "wfDetails": {"wfId":0,"categoryId":1,"wfName":"uda_634d2b27_
    3fdc_43a8_a09e_465a04e73247","createdBy":"admin","createdDate":"07/26/2017
    14:41:40","modifiedBy":"admin","modifiedDate":"07/26/2017 14:41:40","
    wfSeverity":"","wfNotification":"","wfStepNotification":"","notificationDetails":[],"globalParameter":"","stepList":[{"stepId":0,"actionId":1,"stepName":"Start","shapeCoordinates":"158,169","stepSeverity":"","stepNotification":"","yesStepId":0,"yesStepName":null,"noStepId":0,"noStepName":null,"pairConnectionId":0,"pairConnectionName":null,"stepParamList":[],"stepConnectionList":[{"stepConnectionId":0,"nextId":0,"nextStepName":"asfads"}],"isAlreadyExecuted":"","stepSubList":null,"stepType":"Start","subWorkflowId":0,"subWorkflow":null,"description":null,"skipped":false},{"stepId":0,"actionId":5,"stepName":"Stop","shapeCoordinates":"649,221","stepSeverity":"","stepNotification":"","yesStepId":0,"yesStepName":null,"noStepId":0,"noStepName":null,"pairConnectionId":0,"pairConnectionName":null,"stepParamList":[],"stepConnectionList":[],"isAlreadyExecuted":"","stepSubList":null,"stepType":"Stop","subWorkflowId":0,"subWorkflow":null,"description":null,"skipped":false},{"stepId":0,"actionId":4,"stepName":"asfads","shapeCoordinates":"382,196","stepSeverity":"","stepNotification":"","yesStepId":0,"yesStepName":null,"noStepId":0,"noStepName":null,"pairConnectionId":0,"pairConnectionName":null,"stepParamList":[{"stepParamId":0,"key":"environmentParams","valueText":null},{stepParamId":0,"key":"proceedOnFailure","valueText":"false"},{"stepParamId":0,"key":"strArgument","valueText":null},{stepParamId":0,"key":"scriptFile","valueText":"","stepParamId":0,"key":"scriptContent","valueText":"a=10\\nb=\\\"/home/bedrock/rupjit/\\\"necho $a\\necho $b\\nsh /projects/zaloni/BEDROCK_APP_HOME/44_25jul2017_270/bedrock/bedrock-app/scripts/output_handlers/bedrock_output_handler.sh username=admin password=password KEY1=$b\"},{stepParamId":0,\"key\":\"executeOnCluster\",\"valueText\":\"false\"},{stepParamId\":0,\"key\":\"description\",\"valueText\":null}]},\"stepConnectionList":[{"stepConnectionId\":0,\"nextId\":0,\"nextStepName\":\"dsss\"}],\"isAlreadyExecuted":"","stepSubList":null,\"stepType\":\"Shell Action\",\"subWorkflowId\":0,\"subWorkflow\":null,\"description\":null,\"skipped\":false},{stepId\":0,\"actionId\":4,\"stepName\":\"dsss\",\"shapeCoordinates\":\"537,226\",\"stepSeverity":"","stepNotification":"","yesStepId":0,"yesStepName":null,"noStepId":0,"noStepName":null,"pairConnectionId":0,"pairConnectionName":null,"stepParamList":[{"stepParamId":0,\"key\":\"environmentParams\",\"valueText\":null},{stepParamId\":0,\"key\":\"proceedOnFailure\",\"valueText\":\"false\"},{stepParamId\":0,\"key\":\"strArgument\",\"valueText\":null},{stepParamId\":0,\"key\":\"scriptFile\",
```

```

        "userDefinedParamsMapping": "{ \"stepParamsMapping\": [{ \"stepName\": \"wff\", \"stepParams\": [{ \"key\": \"hiveQuery\", \"valueText\": \"${UDA.HiveQuery}\" } ] }, { \"namespaceMapping\": [{ \"namespaceVariable\": \"WORKFLOW.some_key1\", \"userDefinedParam\": \"hard coded value\" } ] } }",
        "actionIcon": null,
        "actionColor": null,
        "actionHaveConsoleOutput": null,
        "wfUserDefinedActionParamsList": [
            {
                "userDefinedActionParamId": 1,
                "userDefinedActionId": 1,
                "paramKey": "HiveQuery",
                "paramLabel": "Hive Query",
                "paramOrder": 1,
                "paramDataType": "hiveAction.hiveQuery",
                "paramRegex": "",
                "paramHelperText": "Enter hive Query",
                "paramMaxChar": null,
                "paramFieldControl": null,
                "paramFieldControlOptions": null,
                "errorMessage": null,
                "defaultValue": "Select * from bedrock.abc",
                "paramValue": null,
                "paramToolTipDisplay": true,
                "required": 1,
                "lineageType": null
            }
        ],
        "page": null
    }

```

Response in Case of Failure:

```
204 No Content
```

1.5.34 Update the User-Defined Action Details

This is a preview feature.

This API allows you to update the details about the user-defined action.

URL:

```

http://<bedrock-host:port>/bedrock-app/services/rest/workflows/userDefinedActions/
↳ {user_defined_action_Id}

```

Note: You must pass the value for the *user_defined_action_Id* parameter in the API URL.

Method:

```
PUT
```

Example Request:

- Scenario 1:

```
{
  "userDefinedActionId": 24,
  "userDefinedActionBusinessName": "LLLLL1",
  "userDefinedActionTechnicalName": "LLLLL1",
  "description": "Use HQL to convert data format",
  "createdBy": "admin",
  "createdDate": 1498661819000,
  "modifiedBy": "admin",
  "modifiedDate": 1498661819000,
  "status": "designed",
  "wfId": null,
  "wfName": null,
  "wfTarPath": "/home/bedrock/template-workflow.tar",
  "wfDetails": null,
  "userDefinedParamsMapping": "{ \"namespaceMapping\": [{ \"namespaceVariable\" : \"WORKFLOW.WORD_COUNT_INPUT\", \"userDefinedParam\": \"${UDA.inputDataPath}\" } ] }",
  "actionIcon": null,
  "actionColor": null,
  "actionHaveConsoleOutput": null,
  "wfUserDefinedActionParamsList": [
    {
      "paramKey": "sourceEntity",
      "paramLabel": "Source Entity",
      "paramOrder": 1,
      "paramDataType": "entity",
      "paramRegex": "(^s)",
      "paramHelperText": "Source Entity Type Id and Version",
      "paramMaxChar": null,
      "paramFieldControl": "entitySearch",
      "paramFieldControlOptions": null,
      "errorMessage": "Invalid target entity",
      "defaultValue": "",
      "paramValue": null,
      "paramToolTipDisplay": false,
      "required": 1,
      "lineageType": ""
    },
    {
      "paramKey": "targetEntity",
      "paramLabel": "Target Entity",
      "paramOrder": 2,
      "paramDataType": "entity",
      "paramRegex": "(^s)",
      "paramHelperText": "Target Entity Type Id and Version",
      "paramMaxChar": null,
      "paramFieldControl": "entitySearch",
      "paramFieldControlOptions": null,
      "errorMessage": "Invalid target entity",
      "defaultValue": "",
      "paramValue": null,
      "paramToolTipDisplay": false,
      "required": 1,
      "lineageType": ""
    }
  ]
}
```


}

- Scenario 2:

If the status is already *DEPLOYED*, you cannot update the details for the `wfUserDefinedActionParamsList` parameter:

```
{
  "userDefinedActionId": 24,
  "userDefinedActionBusinessName": "LLLLL1",
  "userDefinedActionTechnicalName": "LLLLL1",
  "description": "Use HQL to convert data format",
  "createdBy": "admin",
  "createdAt": 1498661819000,
  "modifiedBy": "admin",
  "modifiedDate": 1498661819000,
  "status": "deployed",
  "wfId": null,
  "wfName": null,
  "wfTarPath": "/home/bedrock/template-workflow.tar",
  "wfDetails": null,
  "userDefinedParamsMapping": "{ \"stepParamsMapping\": [{ \"stepName\": \"step_mr_wordcount\", \"stepParams\": [{ \"key\": \"scriptContent\", \"valueText\": \"hadoop jar /usr/hdp/2.5.3.0-37/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount ${UDA.inputDataPath} ${WORKFLOW.WORD_COUNT_OUTPUT}\" } ] } ] }",
  "actionIcon": null,
  "actionColor": null,
  "actionHaveConsoleOutput": null,
  "wfUserDefinedActionParamsList": []
}
```

Example Response:

```
{  
    "responseMessage": "The user defined action: HiveSample11 updated_  
→successfully.",  
    "restUri": "/bedrock-app/services/rest/workflows/userDefinedActions/35",  
    "result": {  
        "userDefinedActionId": 35,  
        "userDefinedActionBusinessName": "HiveSample12",  
        "userDefinedActionTechnicalName": "HiveSample11",  
        "description": "Run hive query",  
        "createdBy": null,  
        "createdAt": null,  
        "modifiedBy": "admin",  
        "modifiedDate": 1501488787000,  
        "status": "designed",  
        "actionId": 0,  
        "wfId": 0,  
        "wfName": "uda_fald9222_42ff_4ee8_985f_e0beab92a8f7",  
        "wfTarPath": "/home/bedrock/rupjit/UDA/workflow-20170731133710840.tar  
→",  
  
        "wfDetails": "{\n\"wfId\":101,\n\"categoryId\":1,\n\"wfName\\\":\"\nuda_\n→fald9222_42ff_4ee8_985f_e0beab92a8f7\",\n\n\"createdBy\\\":\"\nadmin\",\n\n\"createdAt\\\":\"\n07/\n→31/2017 13:43:08\",\n\n\"modifiedBy\\\":\"\nadmin\",\n\n\"modifiedDate\\\":\"\n07/31/2017 13:43:08\"\n→\", \n\n\"wfSeverity\\\":\"\nN\",\n\n\"wfNotification\\\":\"\nN\",\n\n\"wfStepNotification\\\":\"\nN\",\n\n\n→\"notificationDetails\\\":[],\n\n\"globalParameter\\\":{\n\n},\n\n\"stepList\\\":[{\n\n\"stepId\\\":0,\n\n\"actionId\\\":1,\n\n\"stepName\\\":\"\nStart\",\n\n\"shapeCoordinates\\\":[\n\n],\n\n\"stepSeverity\\\":\"\nN\",\n\n\"stepNotification\\\":\"\nN\",\n\n\"yesStepId\\\":0,\n\n\"yesStepName\\\":\n→\"null\",\n\n\"noStepId\\\":0,\n\n\"noStepName\\\":\n\n\"pairConnectionId\\\":0,\n\n\"pairConnectionName\\\":\n→null,\n\n\"stepParamList\\\":[],\n\n\"stepConnectionList\\\":[{\n\n\"stepConnectionId\\\":0,\n\n\"nextId\\\":0,\n\n\"nextStepName\\\":\"\ne3e\"",
```

```

        "userDefinedParamsMapping": "{\\"stepParamsMapping\\": [{\\"stepName\\": \\"wff\\", \\"stepParams\\": [{\\"key\\": \\"hiveQuery\\", \\"valueText\\": \\"${UDA.HiveQuery}\\"}]}], \\"namespaceMapping\\": [{\\"namespaceVariable\\": \\"WORKFLOW.some_key1\\", \\"userDefinedParam\\": \\"hard coded value\\"}]}",
        "actionIcon": null,
        "actionColor": null,
        "actionHaveConsoleOutput": null,
        "wfUserDefinedActionParamsList": [
            {
                "userDefinedActionParamId": 27,
                "userDefinedActionId": 35,
                "paramKey": "HiveQuery",
                "paramLabel": "Hive Query",
                "paramOrder": 1,
                "paramDataType": "hiveAction.hiveQuery",
                "paramRegex": "",
                "paramHelperText": "Enter hive Query",
                "paramMaxChar": null,
                "paramFieldControl": null,
                "paramFieldControlOptions": null,
                "errorMessage": null,
                "defaultValue": "Select * from bedrock.abc",
                "paramValue": null,
                "paramToolTipDisplay": true,
                "required": 1,
                "lineageType": null
            }
        ],
        "page": null
    }

```

Response in Case of Failure:

- Scenario 1:

If the user-defined action Id in the request payload does not match with the one in the URL:

```

{
    "responseMessage": "User Defined Actions id is not match with url.",
    "restUri": "/bedrock-app/services/rest/workflows/userDefinedActions/3",
    "result": null,
    "page": null
}

```

- Scenario 2:

If you try to update the technical name:

```

{
    "responseMessage": "User Defined Actions Technical Name cannot be altered.",
    ↪,
    "restUri": "/bedrock-app/services/rest/workflows/userDefinedActions/3",
    "result": null,
    "page": null
}

```

1.5.35 Delete the User-Defined Action

This is a preview feature.

This API allows you to delete user-defined action specific to the defined action Id.

URL:

```
http://<bedrock-host:port>/bedrock-app/services/rest/workflows/userDefinedActions/  
↪{user_defined_action_Id}
```

Note: You must pass the value for the *user_defined_action_Id* parameter in the API URL.

Method:

DELETE

Example Response:

```
{  
  "responseMessage": "SUCCESS",  
  "restUri": "/bedrock-app/services/rest/workflows/userDefinedActions/12",  
  "result": "The user Defiend action: 12 deleted successfully.",  
  "page": null  
}
```

Response in Case of Failure:

If the user-defined action does not exist with the specified Id:

```
{  
  "responseMessage": "No user defined action exist with id 123",  
  "restUri": "/bedrock-app/services/rest/workflows/userDefinedActions/123",  
  "result": null,  
  "page": null  
}
```

1.5.36 Rename the Workflow Name

This is a preview feature.

This API allows you to rename the name for a single workflow within a specific project.

URL:

```
http://<bedrock-host:port>/bedrock-app/services/rest/workflows/name/{workflow_Id}?  
↪projectId={project_Id}
```

Note: You must pass the values for the *workflow_Id* and *project_Id* parameters in the API URL. To rename the workflows under the public project, set the value for the *project_Id* parameter to *1*.

Method:

PUT

Input Parameters:

```
*"new_workflow_name"
```

Example Request:

```
"Token_masking_workflow"
```

Example Response:

```
{
  "responseMessage": "Workflow name updated successfully",
  "restUri": "/bedrock-app/services/rest/workflows/name/2",
  "result": "Workflow name updated successfully",
  "page": null
}
```

Response in Case of Failure:

- Scenario 1:

If the specified *project_Id* does not exist:

```
{
  "responseMessage": "Project with Id 10 doesn't exist",
  "restUri": "/bedrock-app/services/rest/workflows/name/2",
  "result": null,
  "page": null
}
```

- Scenario 2:

If the specified *workflow_Id* does not exist or does not belong to specified *project_Id*:

```
{
  "responseMessage": "Artifact with Id 222 and type WORKFLOW cannot be accessed_
→from project with Id 1",
  "restUri": "/bedrock-app/services/rest/workflows/name/222",
  "result": null,
  "page": null
}
```

- Scenario 3:

If the updated workflow name already exists:

```
{
  "responseMessage": "Workflow with same name already exist",
  "restUri": "/bedrock-app/services/rest/workflows/name/1",
  "result": null,
  "page": null
}
```

1.5.37 Update the Execution Priority of Workflows in Queue

This API allows you to update the execution priority for all the workflows in queue.

URL:

```
http://<bedrock-host:port>/bedrock-app/services/rest/workflows/instances/priority/  
↪ {execution_priority}
```

Note: You must pass the value for the *execution_priority* parameter in the API URL. The values for the *execution_priority* parameter can be *HIGH/NORMAL/LOW*.

Method:

```
PUT
```

Input Parameters:

```
*[workflow_instance_Id1, workflow_instance_Id2, ...]
```

Example Request:

```
[1, 2, 3]
```

Example Response:

```
{  
  "responseMessage": "7 instance(s) moved to NORMAL priority",  
  "restUri": "/workflows/instances/priority/NORMAL",  
  "result": null,  
  "page": null  
}
```

Response in Case of Failure:

```
{  
  "responseMessage": "Invalid input provided in Instance list [113, 110]",  
  "restUri": "/bedrock-app/services/rest/workflows/instances/priority/HIGH",  
  "result": null,  
  "page": null  
}
```

1.5.38 Delete Workflow Instances in Queue

This API allows you to delete the workflow instances that are in queue from multiple projects.

URL:

```
http://<bedrock-host:port>/bedrock-app/services/rest/workflows/queue/instances
```

Method:

```
DELETE
```

Input Parameters:

```
*[workflow_instance_Id1, workflow_instance_Id2, ....]
```

Example Request:

```
[1, 2, 3]
```

Example Response:

```
{
  "responseMessage": "1 instance(s) deleted successfully",
  "restUri": null,
  "result": null,
  "page": null
}
```

Response in Case of Failure:

```
{
  "responseMessage": "Invalid input provided in Instance list [111, 110]",
  "restUri": "/bedrock-app/services/rest/workflows/queue/instances",
  "result": null,
  "page": null
}
```

1.6 Metadata API

The following APIs can be leveraged to manage the metadata module in ZDP.

1.6.1 Search Entities

This API allows you to perform an entity search for a specific project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/entities/search?projectIds=
↳ {project_Id}
```

Note: You must pass the *project_Id* parameter in the API URL. To search an entity under the public project, set the *project_Id* parameter to 1.

Method:

```
POST
```

Request Payload:

```
{
  *"entityTypeName": "<This indicates the various entity information, such as_
↳ business name, data format. For example, if you specify delimited as the value for_
↳ this parameter, the system returns entities with names that match delimited along_
↳ with the delimited data format entities. Partial string search is supported only if_
↳ you append the * wildcard to the string.>",
  "startDate": "<This is the start date of the date range. If the end date is_
↳ provided, the start date is mandatory. The date format should be in the format: mm/_
↳ dd/yyyy HH:MM:SS.>",
  "endDate": "<This is the end date of the date range. The date format should_
↳ be in the format: mm/dd/yyyy HH:MM:SS.>",
```

```

    "currentPage": "<This is the number according to which the system fetches the
    ↪result of the specified page. The records fetched are based on the set chunk size.>
    ↪",
    "chunkSize": "<This is the value that limits the number of records to be
    ↪fetched per page. When set as '0' (or unspecified), all results are fetched.>",
    "sortBy": "<This is the criterion according to which the fetched results are
    ↪sorted. A few probable values are: createdBy, modifiedBy, createTime, modifiedTime.
    ↪>",
    "orderBy": "<This is the order in which the result needs to be fetched.
    ↪Probable values can be DESC(default) or ASC.>",
    "sourcePlatform": "<This is the source platform.>",
    "sourceSchema": "<This is the source schema.>"
}

```

Example Request:

- Scenario 1:

To search for all entity types whose business name starts with a specific name:

```

{
    "entityTypeName": "Zone*"
}

```

- Scenario 2:

To search for all entity types by using the combination of business name, source platform, source schema, and created within a specific interval:

```

{
    "entityTypeName": "Zone",
    "startDate": "05/10/2016 00:00:00",
    "endDate": "06/04/2016 00:00:00",
    "sourcePlatform": "sp",
    "sourceSchema": "ss"
}

```

Example Response:

The response provides minimal information about the entities.

```

{
    "responseMessage": "Success",
    "restUri": null,
    "result": {
        "resultList": [
            {
                "entityTypeId": 27,
                "version": 1,
                "ruleSetId": 0,
                "ruleSetName": null,
                "businessName": "Zone2",
                "technicalName": "Zone2",
                "description": null,
                "subjectArea": "",
                "label": null,
                "dataStoreId": 0,
                "dataStoreName": "HDFS",
                "frequencyOfUpdate": null,
            }
        ]
    }
}

```

```
"sourcePlatform": "spz2",
"sourceSchema": "ssz2",
"loadType": null,
"preQueryStatements": null,
"dataFileFormatId": 0,
"dataFileFormat": "DELIMITED",
"metadata": null,
"recordLength": null,
"createdBy": "admin",
"createdTime": "06/03/2016 11:58:34",
"customTable": false,
"modifiedBy": "admin",
"modifiedTime": "06/03/2016 11:58:34",
"ingestionTime": null,
"newVersionRequired": false,
"securityContext": {
  "loggedInUser": null,
  "doAsUser": null,
  "projectId": null,
  "artifactPermissions": null,
  "impersonationOn": false,
  "powerUser": false,
  "validImpersonationRequest": false
},
"isDelete": null,
"derived": false,
"fields": [],
"partitionFields": [],
"ruleSetEntityMapping": [],
"complexMetadataEntityMapping": [],
"complexEntityVersion": null,
"hcatTableId": 0,
"tableName": null,
"targetSchema": null,
"targetSchemaValue": null,
"delimiter": null,
"lineDelimiter": null,
"external": false,
"externalDataPath": "",
"externalDataPathValue": "",
"fileFormat": null,
"serdeLib": null,
"libClasspath": null,
"inputFormatLib": null,
"outputFormatLib": null,
"tableProperties": {
  "skip.header.line.count": "0",
  "myKey": "myValue"
},
"storageDescriptorProperties": {},
"serdeInfoProperties": {},
"cdcActionable": false,
"hasInActiveTable": false,
"entityClusterLocation": [],
"validateRequest": false,
"latestVersion": true,
"migrate": false,
```



```

    },
    {
      "syncJob": false,
      "entityTypeId": 26,
      "version": 1,
      "ruleSetId": 0,
      "ruleSetName": null,
      "businessName": "Zone1",
      "technicalName": "Zone1",
      "description": null,
      "subjectArea": "",
      "label": null,
      "dataStoreId": 0,
      "dataStoreName": "HDFS",
      "frequencyOfUpdate": null,
      "sourcePlatform": "spz1",
      "sourceSchema": "ssz1",
      "loadType": null,
      "preQueryStatements": null,
      "dataFileFormatId": 0,
      "dataFileFormat": "DELIMITED",
      "metadata": null,
      "recordLength": null,
      "createdBy": "admin",
      "createdTime": "06/03/2016 11:56:53",
      "customTable": false,
      "modifiedBy": "admin",
      "modifiedTime": "06/03/2016 11:56:53",
      "ingestionTime": null,
      "newVersionRequired": false,
      "securityContext": {
        "loggedInUser": null,
        "doAsUser": null,
        "projectId": null,
        "artifactPermissions": null,
        "impersonationOn": false,
        "powerUser": false,
        "validImpersonationRequest": false
      },
      "isDelete": null,
      "derived": false,
      "fields": [],
      "partitionFields": [],
      "ruleSetEntityMapping": [],
      "complexMetadataEntityMapping": [],
      "complexEntityVersion": null,
      "hcatTableId": 0,
      "tableName": null,
      "targetSchema": null,
      "targetSchemaValue": null,
      "delimiter": null,
      "lineDelimiter": null,
      "external": false,
      "externalDataPath": "",
      "externalDataPathValue": "",
      "fileFormat": null,
      "serdeLib": null,
      "libClasspath": null,
    }
  ]
}

```

```

        "inputFormatLib": null,
        "outputFormatLib": null,
        "tableProperties":
        {
            "skip.header.line.count": "0",
            "myKey": "myValue"
        },
        "storageDescriptorProperties": {},
        "serdeInfoProperties": {},
        "cdcActionable": false,
        "hasInActiveTable": false,
        "entityClusterLocation": [],
        "validateRequest": false,
        "latestVersion": true,
        "migrate": false,
        "syncJob": false
    },
    "totalRecords": 2,
    "currentPage": 0,
    "chunkSize": 0
},
"page": null
}

```

Response in Case of Failure:

Returns the 204 response code indicating *no content*.

1.6.2 Create/Update an Entity

The following two methods can be used to create an entity or update an entity/field(s) in ZDP by defining various parameters.

Methods:

- To create entity:

```
POST
```

For more information, refer to the *Create an Entity* section.

- To update entity:

```
PUT
```

For more information, refer to the *Update an Entity* section.

Request Payload:

```

{
    "entityTypeId": "<This is the entity Id. While creating a new entity_
    ↪type, this parameter is not mandatory (or must be set '0'). For updating an_
    ↪existing entity, this parameter is mandatory and must be set correctly.>",
    "version": "<This is the entity type version Id. While creating a new_
    ↪entity type, this parameter is not mandatory (or must be set '0'). For updating an_
    ↪existing entity type, this value represents the version of the entity being updated.
    ↪ For complex entity types, this value is always '0'.>",

```

```

        *"businessName": "<This is the business name of the entity type.>",
        *"technicalName": "<This is the technical name of the entity type. It can
↳only have letters, numbers, or underscores.>",
            "targetSchema": "<This is the name of the Hive database under which
↳the HCatalog table needs to be created. 'BEDROCK' level namespace can also be
↳passed in this field.>",
            "tableType": "<Indicates the type of Hive table corresponding to the
↳Entity that needs to be created.
Setting 'VIEW' creates 'VIRTUAL_VIEW' type table in Hive indicating a
↳view query needs to be defined for the Entity.
Setting 'EXTERNAL' indicates that the table in Hive for the Entity
↳needs to be maintained externally.
Setting 'INTERNAL' indicates the the table in Hive for the Entity
↳will be managed by Hive.>",
            "entityClusterLocation": ["<Represents the cluster information under
↳which the entity needs to be created. For multiple clusters, an array of cluster
↳information needs to be provided. For a disconnected cluster, do not send any
↳values for this parameter.>"
            {
                "clusterId": "<This is the cluster Id within which the
↳entity needs to be created.>"
            }
        ],
        "external": "<This is a Boolean value indicating whether the Hive
↳table is external (or not) and comes into play only when 'tableType' parameter is
↳undefined. The default value is false. This parameter is deprecated and now
↳corresponds to the 'tableType'.>",
        "externalDataPath": "<When the Hive table is external, this parameter
↳represents the path where the entity is created. This parameter is deprecated and
↳now corresponds to the 'entityDataPath'. When 'tableType' is 'EXTERNAL' and
↳'entityDataPath' is undefined, the value set against this parameter is considered.>
↳",
        "newVersionRequired": "<This indicates if a new version is required,
↳while updating an entity. Possible values are true and false. If a new entity is
↳being created, it defaults to true.>",
        "complexEntityVersion": "<This is required for complex entity. For
↳example, when dataFileFormat is either HL7 or X12.>",
        "tableName": "<This is the Hive table name. While creating a new
↳entity type, this can be set by the user or can be left undefined (so that ZDP
↳names the Hive table using the combination of 'sourcePlatform', 'sourceSchema',
↳'technicalName', and 'targetSchema' values). While updating an entity type, the
↳correct value must be provided.>",
        "tableProperties": "<This is the optional table properties that must
↳be passed as key-value pairs. This property can also include storage descriptor
↳properties. For AVRO entity types, the table property: 'avro.schema.url' must be
↳present containing the value of the HDFS location of the AVRO schema file. For
↳example, /tmp/schemas/avro/person.avsc.>",
        "validateRequest": "<Set this parameter to true to validate the
↳entity structure before creating/updating the entity. Alternatively, set this
↳parameter to false to avoid validating the entity structure before creating/
↳updating the entity.>",
        "description": "<This is the description of the entity type.>",
        "subjectArea": "<This is any string describing the subject areas.>",
        "label": "<This is any string describing identification labels.>",
        *"sourcePlatform": "<This is any string that identifies the source
↳platform. The value can have have letters, numbers, dots, and underscores.>",
        *"sourceSchema": "<This is a string that identifies the source schema. It
↳can only have letters, numbers, or underscores.>",

```

```

        "delimiter": "<Indicates the field delimiter. This is applicable only
        ↳if the Data File Format is Delimited. The default value is a comma (,). If the
        ↳delimiter is non-printable, provide the corresponding unicode value. For example,
        ↳the unicode value for Ctrl+A delimiter corresponds to \\u0001 (\\ to escape).>",
        *"dataStoreId": "<This is the data storage type. '1' for 'BEDROCK' and '2'
        ↳for 'HDFS'. Optional, if 'dataStoreName' is provided.>",
        "dataStoreName": "<This is the data store name. This can be 'BEDROCK'
        ↳or 'HDFS'. Optional, if 'dataStoreId' is already provided.>",
        "dataFileFormat": "<This is the file format. For example, DELIMITED,
        ↳FIXED_LENGTH, HL7, X12, XML, JSON, CUSTOM, AVRO, ORC, PARQUET, RC, SEQUENCE FILE.>",
        *"dataFileFormatId": "<This is the file format Id. 1 for DELIMITED, 2 for
        ↳FIXED_LENGTH, 3 for HL7, 4 for X12, 6 for JSON, 7 for CUSTOM, 8 for AVRO, 9 for ORC,
        ↳10 for PARQUET, 11 for RC, 12 for SEQUENCE. This parameter is mandatory only if
        ↳the 'dataFileFormat' is not defined. While creating an entity of VIEW tableType,
        ↳this value must be set to '12'>",
        "view": {
            "query": "<This is the query that is applicable while
            ↳creating or updating an entity of 'VIEW' tableType. Provide the select query
            ↳aplicable for the VIEW entity. A simple example can be - 'SELECT * FROM [TARGET_
            ↳SCHEMA].[TABLE_NAME]'>"
        },
        "owner": "<This parameter is applicable when impersonation is enabled
        ↳and indicates the impersonated (or proxy) user owning the entity that is being
        ↳created. When Impersonation is enabled and this parameter is undefined, the logged-
        ↳in user is automatically set while saving.>",
        "frequencyOfUpdate": "<This is a string for the frequency of update.>
        ↳",
        "recordLength": "<This is the length of single record. This is
        ↳required only if the dataFileFormat is Fixed Length.>",
        "loadType": "<This is a string that indicates the load type. For
        ↳example, Incremental or Full.>",
        "fields": [
            {
                "primary": "<This indicates if this field is a
                ↳primary key. Possible values are true and false.>",
                "fieldId": "<This is the Id of a field. For a new
                ↳field, it should be 0.>",
                "position": "<This is the position (or order) of the
                ↳field in a data row.>",
                *"fieldBusinessName": "<This is the business name of the
                ↳field.>",
                "searchable": "<This indicates if the field should be
                ↳displayed in searches. Possible values are true and false. This is only applicable
                ↳if the data store is Bedrock.>",
                "fieldDataLength": "<This is the maximum possible
                ↳length of the field data.>",
                *"fieldDataType": "<This is the data type of the field.
                ↳The following are the possible values: ARRAY, BIGINT, BINARY, BOOLEAN, DOUBLE,
                ↳FLOAT, INT, MAP, RECORD, SMALLINT, STRING, TEXT, TINYINT.>",
                "ruleSetName": "<This is the entity rule set name to be associated to
                ↳the entity type.>",
                "ruleSetId": "<This is the Id of the rule set to be associated to the
                ↳entity type.>",
                "hasInActiveTable": "<This is a boolean flag to indicate if the
                ↳entity type needs to have inactive table if CDC is configured. This flag needs to
                ↳be true only if any field in the entity type is configured to be a CDC field.>",
                "serdeLib": "<This is the Serde library. For example, for XML type,
                ↳it is com.ibm.spss.hive.serde2.xml.XmlSerDe. It applies to CUSTOM data format.>",

```

```

        "libClasspath": "<This is the semicolon-separated list of HDFS paths_
↳to jar files. For example, /user/bedrock/lib/hivexmlserde-1.0.5.2.jar;/user/bedrock/
↳tmp/lib/hcatalog-core-1.2.jar. It applies to CUSTOM data format.>",
        "inputFormatLib": "<This is the library for input format. For example,
↳ for XML type, it is com.ibm.spss.hive.serde2.xml.XmlInputFormat. This is required_
↳for CUSTOM data format.>",
        "outputFormatLib": "<This is the library for output format. For_
↳example, for XML type, it is org.apache.hadoop.hive.ql.io.
↳HiveIgnoreKeyTextOutputFormat. This is required for CUSTOM data format.>",
        "serdeInfoProperties": "<This is the Serde properties in the form of_
↳key-value pairs. It applies to the CUSTOM data format.>",
        "fieldDataTypeschema": "<This is the schema definition of the entity_
↳field defined as a JSON string. See Example 1 for more details on input format. The_
↳structure of the schema is the following:"
        simple
↳ := <simple_type>
        array
↳ := { <array_type> }
        map
↳ := { <map_type> }
        record
↳ := { <record_type> }
        simple_type
↳ "type" : <primitive_type> :=
        array_type := "type
↳ " : "array", "items" : <simple> | <complex>
        map_type
↳ := "type" : "map", "values" : <simple> | <complex>
        record_type :=
↳ "type" : "record", "fields" : [ <record_simple_type> | <record_complex_type> , ... ]
        record_simple_type := "name" : <fieldname> ,
↳<simple_type>
        record_complex_type := "name" :
↳<fieldname>, <complex_type>
        complex
↳ := { <complex_type> }
        complex_type :=
↳<array_type> | <map_type> | <record_type>
        primitive_type := "int" | "tinyint" |
↳"smallint" | "bigint" | "boolean" | "float" | "double" | "string" | "binary"
        fieldname
↳ := "<field name as a string containing only numbers, letters, and underscores._
↳See Example 1 of field schemas below for creating and updating entity types. Use_
↳either 'type' or 'id' in a payload (or both). The recommendation is to use 'type'_
↳instead.>"
        "dataScale": "<The data scale i.e. the number of digits after decimal_
↳for decimal values>",
        "fieldTechnicalName": "<This is the technical name of the field. It_
↳can only have letters, numbers, or underscores. To capture data quality rules and_
↳business names of fields within a complex structure, the fieldTechnicalName must_
↳have the path to the field name within the complex structure. Fields that start_
↳with / (e.g. /location/address/city) are special fields that capture metadata_
↳information about a leaf node within a complex field. These types of fields are_
↳used mainly for storing DQ information and/or business field name changes of_
↳HCatalog-integrated fields. Hence, these special fields are not HCatalog integrated.
↳ See Example 2 for more details.>",
        "dataFormat": "<This is the format of the data. This is meaningful_
↳for date values.>",

```

```

        "sensitivityId": "<This is the Id of the sensitivity attribute. If
↪sensitivityValue is provided, this is not required. If a new sensitivity value is
↪being added, set this parameter to 0.>",
        "sensitivityValue": "<This is the value of the sensitivity attribute.
↪If a positive sensitivityId is provided, this is not required. Set to N/A if you do
↪not know the sensitivity value or you do not want to set it.>",
        "startOffset": "<This is the start offset of the field. Required only
↪if dataFileFormat of the entity type is Fixed Length.>",
        "endOffset": "<This is the end offset of the field. This is required
↪only if dataFileFormat of the entity type is Fixed Length. If startOffset and
↪fieldDataLength are provided, the endOffset is not required.>",
        "tokenizerAlgorithmId": "<This is the Id of the tokenizer algorithm.
↪This property is mutually exclusive with tokenizerAlgorithmName.>",
        "tokenizerAlgorithmName": "<This is the name of the tokenizer
↪algorithm. This property is mutually exclusive with tokenizerAlgorithmId.>",
        "maskingPattern": "<This is the masking pattern with which the field
↪data is to be masked with.>",
        "cdcColumn": "<This is the boolean value that identifies if the field
↪is a change data capture field. This has to be true for only one field only and
↪cannot be true for field with primary key.>",
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "<This is the Id of the custom
↪attribute, which is a String parameter.>",
                        "value": "<This the value of the custom
↪attribute.>",
                        "managed": "<This is the type of the
↪attribute: managed (true) or non-managed (false). The system always assumes this
↪value as false.>"
                    },
                    {
                        "id": "<This is the Id of the custom
↪attribute.>",
                        "value": "<This the value of the custom
↪attribute.>",
                        "managed": "<This is the type of the
↪attribute: managed (true) or non-managed (false). The system always assumes this
↪value as false.>"
                    }
                ],
                "source": {
                    "type": "<The type of the source. For ZDP,
↪you must specify the type as BR.>,"
                    "name": "<This is the name of the source. For
↪ZDP, you must specify the name as Zaloni Bedrock.>"
                }
            }
        ],
        "complexType" : "<This indicates if the data type is complex. If
↪field data type is primitive type. set as false. For array, map, record, set this
↪parameter as true.>"
    },
    "ruleMappings": [
        {
            "ruleName": "<This is an existing rule name.>",
            "ruleSetName": "<This is an existing rule set name. Either of
↪ruleName or ruleSetName can be defined but not both.>",

```

```

        "ruleId": "<This is the Id of the rule.>",
        "ruleSetId": "<This is the Id of the rule set.>",
        "executionOrder": "<This is an optional value to indicate the
↪order in which the rule needs to be executed.>",
        "stopExecution": "<This is a boolean flag to indicate if you
↪want to prevent execution of rules down the execution order if the current rule
↪fails.>",
        *"variableMapping": [
            {
                "variableName": "<This is the name of the variable
↪associated with the field.>",
                "fieldId": "<This is the Id for the field associated with
↪the variable.>",
                "fieldTechnicalName": "<This is the technical name of the
↪field associated with the variable.>",
                "varType": "<This is the type of the variable.>"
            }
        ],
        "partitionFields": [
            {
                "fieldDataType": "<This is the data type of the partition
↪field.>",
                "partitionFieldName": "<This is the unique name for the
↪partition field.>",
                "partitionFieldPosition": "<This is the partition field
↪position.>",
                "partitionFieldId": "<This is the Id of the partition field.
↪If new partition is added, it must be 0.>"
            }
        ],
        "cdcActionable": "<This is a boolean flag to indicate if the entity
↪type has been configured for CDC action. This must be set to true only if any of
↪the fields is marked as CDC Column field.>",
        "bucketing": {
            *"numberOfBuckets": "<This is the number (n) of buckets that need
↪to be created (number decided on the basis n-1).>",
            *"bucketingFields": [
                {
                    "fieldId": "<This is a mandatory while
↪updating an entity and indicates the Id of the field that needs to be bucketed.
↪While creating an Entity, this parameter is not required.>",
                    *"fieldTechnicalName": "<This is the technical
↪name of the field that needs to be bucketed.>",
                    "fieldBusinessName": "<This is the business
↪name of the field that needs to be bucketed.>",
                    *"fieldDataType": "<This is the data type of the
↪entity field that needs to be bucketed.>",
                    *"bucketingFieldOrder": "<This is an integer value
↪indicating the order of the fields that are defined for bucketing. While defining
↪multiple entity fields for bucketing, the order is mandatory as it impacts the
↪internal evaluation logic.>"
                }
            ],
            "sortingFields": [
                {
                    "fieldId": "<This is a mandatory only while
↪updating an entity and indicates the Id of the field that needs to be sorted. While
↪creating an entity, this parameter is not required.>",

```

```

        *"fieldTechnicalName": "<This is the technical_
↪name of the field that needs to be sorted.>",
        "fieldBusinessName": "<This is the business_
↪name of the field that needs to be sorted.>",
        *"fieldDataType": "<This is the data type of the_
↪entity field that needs to be bucketed.>",
        "sortingFieldOrder": "<This is an integer_
↪value indicating the order of the sort field defined. For example, setting 1_
↪indicates the first sorting preference.>",
        *"sortType": "<This is the sorting mode for the_
↪entity field defined. The probable values can be ASCENDING or DESCENDING.>"
    }
},
"skewing": {
    "skewingFields": [
        {
            "fieldId": "<This is a mandatory while_
↪updating an entity and indicates the Id of the field that needs to be skewed. While_
↪creating an entity, this parameter is not required>",
            *"fieldTechnicalName": "<This is the technical_
↪name of the field that needs to be skewed.>",
            "fieldBusinessName": "<This is the business_
↪name of the field that needs to be skewed.>",
            *"fieldDataType": "<This is the data type of the_
↪entity field that needs to be skewed.>",
            *"skewingFieldOrder": "<This is the integer value_
↪indicating the order of the entity field defined for skewing.>",
            *"skewingFieldValues": [
                "<This is the comma-separated list of_
↪column values for the entity field defined for skewing.>"
            ]
        }
    ],
    "storedAsDirectories": "<This is a boolean value indicating_
↪whether the skewed files needs to be managed within directories or not. Probable_
↪value true/false.>"
},
}

```

Note: There is no validation in place for configuring the entity type for the CDC functionality. So any configurations made to the entity type with invalid value, leads to failure in the CDC action and generation of inactive table.

1.6.2.1 Create an entity

This API allows you to create HDFS managed entity.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/{project_Id}/entities
```

Note: You must pass the *project_Id* parameter in the API URL. To create an entity under the public project, set the *project_Id* parameter to 1.

Method:

POST

Example Request:

- Scenario 1:

To create a *DELIMITED* entity:

```
{
  "entityTypeId": 0,
  "version": 1,
  "newVersionRequired": false,
  "createdBy": "admin",
  "createdTime": "06/06/2016 11:23:19",
  "modifiedBy": "admin",
  "fields": [
    {
      "fieldId": -1,
      "position": 1,
      "fieldTechnicalName": "Id",
      "dataTypeId": 10,
      "fieldDataType": "INT",
      "fieldDataLength": "",
      "dataScale": 0,
      "dataFormat": "",
      "fieldBusinessName": "Id",
      "description": null,
      "primary": false,
      "searchable": false,
      "masked": false,
      "tokenized": false,
      "nonTopLevel": false,
      "managedListId": 0,
      "managedListName": null,
      "tokenizerAlgorithmId": 0,
      "tokenizerAlgorithmName": null,
      "maskingPattern": "",
      "startOffset": 0,
      "endOffset": 0,
      "sensitivityId": 0,
      "sensitivityValue": null,
      "ruleMappings": [
        {
          "ruleSetId": null,
          "ruleId": 8,
          "ruleName": "SimpleRule_API",
          "ruleSetName": null,
          "ruleTag": "",
          "ruleDataType": "STRING",
          "ruleSetDataType": null,
          "ruleSetTag": null,
          "stopExecution": false,
          "executionOrder": 2,
          "variableMapping": [
            {
              "variableName": "primaryVar",
              "fieldId": 11675,
```

```

        "fieldTechnicalName": "/Add/STATE",
        "varType": null
    },
    ],
    "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\
↪"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\\\"}\\),\\"firstGroup\\
↪":true}}}"
    },
    ],
    "cdcColumn": false,
    "customAttributes": [
        {
            "attributes": [
                {
                    "id": "privRating",
                    "value": "0",
                    "managed": false
                },
                {
                    "id": "alias",
                    "value": "Uniqueid",
                    "managed": false
                }
            ],
            "source": {
                "type": "BR",
                "name": "Zaloni Bedrock"
            }
        },
        {
            "fieldId": -2,
            "position": 2,
            "fieldTechnicalName": "EFID",
            "dataTypeId": 7,
            "fieldDataType": "STRING",
            "fieldDataLength": "",
            "dataScale": 0,
            "dataFormat": "",
            "fieldBusinessName": "EFID",
            "description": null,
            "primary": false,
            "searchable": false,
            "masked": false,
            "tokenized": false,
            "nonTopLevel": false,
            "managedListId": 0,
            "managedListName": null,
            "tokenizerAlgorithmId": 0,
            "tokenizerAlgorithmName": null,
            "maskingPattern": "",
            "startOffset": 0,
            "endOffset": 0,
            "sensitivityId": 0,
        }
    ],
    "complexType": false
},
{
    "fieldId": -2,
    "position": 2,
    "fieldTechnicalName": "EFID",
    "dataTypeId": 7,
    "fieldDataType": "STRING",
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "EFID",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": null,
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,
    "maskingPattern": "",
    "startOffset": 0,
    "endOffset": 0,
    "sensitivityId": 0,
}

```

```

        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleSetId": null,
                "ruleId": 8,
                "ruleName": "SimpleRule_API",
                "ruleSetName": null,
                "ruleTag": "",
                "ruleDataType": "STRING",
                "ruleSetDataType": null,
                "ruleSetTag": null,
                "stopExecution": false,
                "executionOrder": 2,
                "variableMapping": [
                    {
                        "variableName": "primaryVar",
                        "fieldId": 11675,
                        "fieldTechnicalName": "/Add/STATE",
                        "varType": null
                    }
                ],
                "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"
↪"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\\\"}\\),\\"firstGroup\\
↪":true}}}"
            }
        ],
        "cdcColumn": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "privRating",
                        "value": "0",
                        "managed": false
                    },
                    {
                        "id": "alias",
                        "value": "Uniqueid",
                        "managed": false
                    }
                ],
                "source": {
                    "type": "BR",
                    "name": "Zaloni Bedrock"
                }
            }
        ],
        "complexType": false
    },
    {
        "fieldId": -3,
        "position": 3,
        "fieldTechnicalName": "Target",
        "dataTypeId": 7,
        "fieldDataType": "STRING",
        "fieldDataLength": "",

```

```

        "dataScale": 0,
        "dataFormat": "",
        "fieldBusinessName": "Target",
        "description": null,
        "primary": false,
        "searchable": false,
        "masked": false,
        "tokenized": false,
        "nonTopLevel": false,
        "managedListId": 0,
        "managedListName": null,
        "tokenizerAlgorithmId": 0,
        "tokenizerAlgorithmName": null,
        "maskingPattern": "",
        "startOffset": 0,
        "endOffset": 0,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleSetId": null,
                "ruleId": 8,
                "ruleName": "SimpleRule_API",
                "ruleSetName": null,
                "ruleTag": "",
                "ruleDataType": "STRING",
                "ruleSetDataType": null,
                "ruleSetTag": null,
                "stopExecution": false,
                "executionOrder": 2,
                "variableMapping": [
                    {
                        "variableName": "primaryVar",
                        "fieldId": 11675,
                        "fieldTechnicalName": "/Add/STATE",
                        "varType": null
                    }
                ],
                "expressionPayload": "{\"group\":{\"operator\":\"AND\",\\
↪\"rules\": [{\\\"data\":\\\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\\\"}],\\\"firstGroup\\
↪\":true}}\"
            }
        ],
        "cdcColumn": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "privRating",
                        "value": "0",
                        "managed": false
                    },
                    {
                        "id": "alias",
                        "value": "Uniqueid",
                        "managed": false
                    }
                ]
            }
        ]
    },
    {
        "id": "alias",
        "value": "Uniqueid",
        "managed": false
    }
]

```

```

        },
        "source": {
            "type": "BR",
            "name": "Zaloni Bedrock"
        }
    },
    "complexType": false
},
{
    "partitionFields": [],
    "complexMetadataEntityMapping": [],
    "validateRequest": false,
    "businessName": "Zone4",
    "technicalName": "Zone4",
    "description": "",
    "subjectArea": "",
    "label": "",
    "dataStoreId": 2,
    "dataFileFormatId": 1,
    "delimiter": "\\u0001",
    "entityCreateOptions": 1,
    "complexEntityVersion": null,
    "recordLength": null,
    "frequencyOfUpdate": "",
    "sourcePlatform": "spz4",
    "sourceSchema": "ssz4",
    "loadType": "",
    "hasInActiveTable": false,
    "ruleSetId": 0,
    "isComplexEntity": false,
    "serdeInfoProperties": {
        "": ""
    },
    "libClasspath": "",
    "serdeLib": "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
    "inputFormatLib": "org.apache.hadoop.mapred.TextInputFormat",
    "outputFormatLib": "org.apache.hadoop.hive.ql.io.
↪HiveIgnoreKeyTextOutputFormat",
    "tableName": "spz4_ssz4_Zone4_1",
    "targetSchema": "zonesp",
    "external": true,
    "externalDataPath": "",
    "tableType": "EXTERNAL",
    "entityClusterLocation": [
        {
            "clusterId": "192_168_1_38"
        },
        {
            "clusterId": "192_168_1_39"
        }
    ],
    "fileFormat": "TextFile",
    "tableProperties": {
        "skip.header.line.count": "0",
        "myKey": "myValue"
    }
}

```

```
    },
    "owner": "do2user1",
    "derived": false,
    "position": 1,
    "cdcActionable": true
}
```

Example Property:

To define the *fieldDataTypeSchema* property:

```
{
  "type": "record",
  "fields": [
    {
      "name": "id",
      "type": "int"
    },
    {
      "name": "username",
      "type": "string"
    },
    {
      "name": "name",
      "type": "string"
    },
    {
      "name": "shippingaddress",
      "type": "record",
      "fields": [
        {
          "name": "address1",
          "type": "string"
        },
        {
          "name": "address2",
          "type": "string"
        },
        {
          "name": "city",
          "type": "string"
        },
        {
          "name": "state",
          "type": "string"
        }
      ]
    },
    {
      "name": "countries",
      "type": "map",
      "values": {
        "type": "record",
        "fields": [
          {
            "name": "name",
            "type": "string"
          }
        ]
      }
    }
  ]
}
```

```

        {
            "name": "location",
            "type": "string"
        }
    ]
},
{
    "name": "aliases",
    "type": "array",
    "items": {
        "type": "string"
    }
},
{
    "name": "orders",
    "type": "array",
    "items": {
        "type": "record",
        "fields": [
            {
                "name": "itemid",
                "type": "int"
            },
            {
                "name": "orderdate",
                "type": "string"
            },
            {
                "name": "status",
                "type": "string"
            }
        ]
    }
}
]
}

```

- Scenario 2:

To create a custom entity:

```

{
    "entityTypeId": 0,
    "version": 0,
    "newVersionRequired": false,
    "createdBy": "admin",
    "modifiedBy": "admin",
    "fields": [
        {
            "fieldId": 0,
            "position": 0,
            "fieldTechnicalName": "arr_of_int",
            "dataTypeId": 20,
            "fieldDataType": "ARRAY",
            "startOffset": null,
            "endOffset": null,
            "fieldDataLength": "",

```

```

        "dataScale": 0,
        "dataFormat": "",
        "fieldBusinessName": "arr_of_int",
        "fieldDataTypeSchema": "{\"id\":20,\"type\":\"array\",\\
↪\"items\":{\"id\":14,\"type\":\"INT\"}}",
        "primary": false,
        "searchable": false,
        "masked": false,
        "maskingPattern": "",
        "tokenized": false,
        "tokenizerAlgorithmId": 0,
        "managedListId": 0,
        "managedListName": "",
        "tokenizerAlgorithmName": null,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleSetId": null,
                "ruleId": 8,
                "ruleName": "SimpleRule_API",
                "ruleSetName": null,
                "ruleTag": "",
                "ruleDataType": "STRING",
                "ruleSetDataType": null,
                "ruleSetTag": null,
                "stopExecution": false,
                "executionOrder": 2,
                "variableMapping": [
                    {
                        "variableName": "primaryVar",
                        "fieldId": 11675,
                        "fieldTechnicalName": "/Add/STATE",
                        "varType": null
                    }
                ],
                "expressionPayload": "{\"group\":{\"operator\":\"AND\",\\
↪\"rules\":[{\"data\":\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\"}],\"firstGroup\\
↪\":true}}"
            }
        ],
        "cdcColumn": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "privRating
↪",
                        "value": "0",
                        "managed": false
                    },
                    {
                        "id": "alias",
                        "value": "Uniqueid
↪",
                        "managed": false
                    }
                ]
            }
        ],

```



```

        "source": {
            "type": "BR",
            "name": "Zaloni Bedrock"
        }
    },
    "complexType": false
},
{
    "partitionFields": [],
    "businessName": "Custom_Xml",
    "technicalName": "Custom_Xml",
    "description": "",
    "subjectArea": "",
    "label": "",
    "dataStoreId": 2,
    "dataFileFormatId": 7,
    "delimiter": "",
    "complexEntityVersion": "1",
    "recordLength": null,
    "frequencyOfUpdate": "",
    "sourcePlatform": "SP",
    "sourceSchema": "SS",
    "loadType": "",
    "hasInActiveTable": false,
    "ruleSetId": 0,
    "isComplexEntity": true,
    "serdeInfoProperties": {
        "column.xpath.item": "/arr_of_int/item",
        "serialization.format": "1"
    },
    "libClasspath": "/tmp/hive_xml/libclasspath/hivexmlserde-1.0.5.2.jar",
    "serdeLib": "com.ibm.spss.hive.serde2.xml.XmlSerDe",
    "inputFormatLib": "com.ibm.spss.hive.serde2.xml.XmlInputFormat",
    "outputFormatLib": "org.apache.hadoop.hive.ql.io.
↪HiveIgnoreKeyTextOutputFormat",
    "storageDescriptorProperties": {
        "": ""
    },
    "hcatTableId": 0,
    "tableName": "",
    "targetSchema": "aa_xml",
    "tableType": "INTERNAL",
    "entityClusterLocation": [
        {
            "clusterId": "192_168_1_39"
        }
    ],
    "fileFormat": "TextFile",
    "tableProperties": {
        {
            "skip.header.line.count": "0",
            "myKey": "myValue"
        },
        "owner": "jsmith",
        "fileFormat": "TextFile",
        "tableProperties": {
            "xmlinput.start": "arr_of_int"
        }
    }
}

```

```

    },
    "cdcActionable": true
}

```

- Scenario 3:

To create an entity (such as *DELIMITED*) with *Bucketing*, *Skewing* as well as *Sorting* definitions.

Note: Bucketing, skewing, and sorting definitions for an entity are independent configurations and has no dependencies with each other.

```

{
    "businessName": "Data_Sheet",
    "technicalName": "datasheet",
    "description": "Entity generated from file [bucketSeat.txt]",
    "subjectArea": null,
    "label": null,
    "dataStoreId": 2,
    "tableType": "EXTERNAL",
    "hcatIntegrated": true,
    "dataFileFormatId": 1,
    "delimiter": ",",
    "complexEntityVersion": null,
    "recordLength": null,
    "frequencyOfUpdate": "",
    "sourcePlatform": "undefined",
    "sourceSchema": "undefined",
    "loadType": "",
    "hasInactiveTable": false,
    "ruleSetId": 0,
    "isComplexEntity": false,
    "serdeInfoProperties": {
        "": ""
    },
    "libClasspath": "",
    "serdeLib": "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
    "inputFormatLib": "org.apache.hadoop.mapred.TextInputFormat",
    "outputFormatLib": "org.apache.hadoop.hive ql.io.
↪HiveIgnoreKeyTextOutputFormat",
    "hcatTableId": 0,
    "tableName": "datasheet_1",
    "targetSchema": "2304",
    "fileFormat": "TextFile",
    "owner": "do2user1",
    "cdcActionable": true,
    "entityTypeId": 0,
    "version": 0,
    "newVersionRequired": false,
    "createdBy": "admin",
    "modifiedBy": "admin",
    "fields": [
        {
            "fieldId": -1,
            "position": 1,
            "fieldTechnicalName": "id",
            "dataTypeId": 19,

```

```

"fieldDataType": "INTEGER",
"fieldDataLength": "",
"dataScale": 0,
"dataFormat": null,
"fieldBusinessName": "ID",
"description": null,
"primary": false,
"searchable": false,
"masked": false,
"tokenized": false,
"nonTopLevel": false,
"managedListId": 0,
"managedListName": null,
"tokenizerAlgorithmId": 0,
"tokenizerAlgorithmName": "",
"algoArguments": "",
"maskingPattern": null,
"startOffset": 0,
"endOffset": 0,
"sensitivityId": 0,
"sensitivityValue": "",
"ruleMappings": [
  {
    "ruleSetId": null,
    "ruleId": 8,
    "ruleName": "SimpleRule_API",
    "ruleSetName": null,
    "ruleTag": "",
    "ruleDataType": "STRING",
    "ruleSetDataType": null,
    "ruleSetTag": null,
    "stopExecution": false,
    "executionOrder": 2,
    "variableMapping": [
      {
        "variableName": "primaryVar",
        "fieldId": 11675,
        "fieldTechnicalName": "/Add/STATE",
        "varType": null
      }
    ],
    "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"
↪"rules\\": [{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\
↪":true}}"
  }
],
"cdcColumn": false,
"customAttributes": [
  {
    "attributes": [
      {
        "id": "",
        "value": "",
        "managed": false
      }
    ],
    "source": {
      "type": "BR",

```

```

        "name": "Zaloni Bedrock"
      }
    }
  ],
  "complexType": false,
  "isPartitionField": false
},
{
  "fieldId": -2,
  "position": 2,
  "fieldTechnicalName": "firstname",
  "dataTypeId": 7,
  "fieldDataType": "STRING",
  "fieldDataLength": "",
  "dataScale": 0,
  "dataFormat": null,
  "fieldBusinessName": "First Name",
  "description": null,
  "primary": false,
  "searchable": false,
  "masked": false,
  "tokenized": false,
  "nonTopLevel": false,
  "managedListId": 0,
  "managedListName": null,
  "tokenizerAlgorithmId": 0,
  "tokenizerAlgorithmName": "",
  "algoArguments": "",
  "maskingPattern": null,
  "startOffset": 0,
  "endOffset": 0,
  "sensitivityId": 0,
  "sensitivityValue": "",
  "ruleMappings": [
    {
      "ruleSetId": null,
      "ruleId": 8,
      "ruleName": "SimpleRule_API",
      "ruleSetName": null,
      "ruleTag": "",
      "ruleDataType": "STRING",
      "ruleSetDataType": null,
      "ruleSetTag": null,
      "stopExecution": false,
      "executionOrder": 2,
      "variableMapping": [
        {
          "variableName": "primaryVar",
          "fieldId": 11675,
          "fieldTechnicalName": "/Add/STATE",
          "varType": null
        }
      ],
      "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"
    }
  ],
  "rules\\": [{"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\":true}}"
}
],

```

```

"cdcColumn": false,
"customAttributes": [
  {
    "attributes": [
      {
        "id": "",
        "value": "",
        "managed": false
      }
    ],
    "source": {
      "type": "BR",
      "name": "Zaloni Bedrock"
    }
  }
],
"complexType": false,
"isPartitionField": false
},
{
  "fieldId": -3,
  "position": 3,
  "fieldTechnicalName": "lastname",
  "dataTypeId": 7,
  "fieldDataType": "STRING",
  "fieldDataLength": "",
  "dataScale": 0,
  "dataFormat": null,
  "fieldBusinessName": "Last Name",
  "description": null,
  "primary": false,
  "searchable": false,
  "masked": false,
  "tokenized": false,
  "nonTopLevel": false,
  "managedListId": 0,
  "managedListName": null,
  "tokenizerAlgorithmId": 0,
  "tokenizerAlgorithmName": "",
  "algoArguments": "",
  "maskingPattern": null,
  "startOffset": 0,
  "endOffset": 0,
  "sensitivityId": 0,
  "sensitivityValue": "",
  "ruleMappings": [
    {
      "ruleSetId": null,
      "ruleId": 8,
      "ruleName": "SimpleRule_API",
      "ruleSetName": null,
      "ruleTag": "",
      "ruleDataType": "STRING",
      "ruleSetDataType": null,
      "ruleSetTag": null,
      "stopExecution": false,
      "executionOrder": 2,
      "variableMapping": [

```

```

        {
            "variableName": "primaryVar",
            "fieldId": 11675,
            "fieldTechnicalName": "/Add/STATE",
            "varType": null
        },
        ],
        "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"
↪"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\
↪":true}}"
    },
    ],
    "cdcColumn": false,
    "customAttributes": [
        {
            "attributes": [
                {
                    "id": "",
                    "value": "",
                    "managed": false
                }
            ],
            "source": {
                "type": "BR",
                "name": "Zaloni Bedrock"
            }
        }
    ],
    "complexType": false,
    "isPartitionField": false
},
],
"partitionFields": [],
"complexMetadataEntityMapping": [],
"validateRequest": false,
"entityClusterLocation": [
    {
        "clusterId": "192_168_1_39"
    }
],
"bucketing": {
    "numberOfBuckets": 2,
    "bucketingFields": [
        {
            "fieldId": -2,
            "fieldTechnicalName": "firstname",
            "fieldBusinessName": "First Name",
            "fieldDataType": "STRING",
            "bucketingFieldOrder": 1
        }
    ],
    "sortingFields": [
        {
            "fieldId": -1,
            "fieldTechnicalName": "id",
            "fieldBusinessName": "ID",
            "fieldDataType": "INTEGER",
            "sortingFieldOrder": 1,

```

```

        "sortType": "ASCENDING"
      }
    ],
    "skewing": {
      "skewingFields": [
        {
          "fieldId": -2,
          "fieldTechnicalName": "lastname",
          "fieldBusinessName": "Last Name",
          "fieldDataType": "STRING",
          "skewingFieldOrder": 1,
          "skewingFieldValues": [
            "Dutta",
            "Goswami"
          ]
        }
      ],
      "storedAsDirectories": true
    }
  },
}

```

- Scenario 4:

To create a *JSON* entity:

```

{
  "entityTypeId": 0,
  "version": 0,
  "fields": [
    {
      "fieldId": -1,
      "position": 1,
      "fieldTechnicalName": "username",
      "dataTypeId": 4,
      "fieldDataType": "STRING",
      "startOffset": null,
      "endOffset": null,
      "fieldDataLength": "",
      "dataScale": 0,
      "dataFormat": "",
      "fieldBusinessName": "username",
      "fieldDataTypeSchema": "{\"id\":4,\"type\":\"string\"}",
      "primary": false,
      "searchable": false,
      "masked": false,
      "maskingPattern": "",
      "tokenized": false,
      "tokenizerAlgorithmId": 0,
      "managedListId": 0,
      "managedListName": "",
      "tokenizerAlgorithmName": null,
      "sensitivityId": 0,
      "sensitivityValue": null,
      "ruleMappings": [
        {
          "ruleSetId": null,
          "ruleId": 8,

```

```

"ruleName": "SimpleRule_API",
"ruleSetName": null,
"ruleTag": "",
"ruleDataType": "STRING",
"ruleSetDataType": null,
"ruleSetTag": null,
"stopExecution": false,
"executionOrder": 2,
"variableMapping": [
  {
    "variableName": "primaryVar",
    "fieldId": 11675,
    "fieldTechnicalName": "/Add/STATE",
    "varType": null
  }
],
"expressionPayload": "{\"group\":{\"operator\":\"AND\",\\
↪\"rules\": [{\\\"data\":\\\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\\\"}],\\\"firstGroup\\
↪\":true}}\"
}

],
"cdcColumn": false,
"customAttributes": [
  {
    "attributes": [
      {
        "id": "privRating",
        "value": "0",
        "managed": false
      },
      {
        "id": "alias",
        "value": "Uniqueid",
        "managed": false
      }
    ],
    "source": {
      "type": "BR",
      "name": "Zaloni Bedrock"
    }
  }
],
"complexType": false
},
{
  "fieldId": -2,
  "position": 2,
  "fieldTechnicalName": "/who_tweets/category",
  "dataTypeId": 4,
  "fieldDataType": "STRING",
  "startOffset": null,
  "endOffset": null,
  "fieldDataLength": "",
  "dataScale": 0,
  "dataFormat": "",
  "fieldBusinessName": "category",

```



```

        "fieldDataTypeSchema": "",
        "primary": false,
        "searchable": false,
        "masked": false,
        "maskingPattern": "",
        "tokenized": false,
        "tokenizerAlgorithmId": 0,
        "managedListId": 0,
        "managedListName": "",
        "tokenizerAlgorithmName": null,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleSetId": null,
                "ruleId": 8,
                "ruleName": "SimpleRule_API",
                "ruleSetName": null,
                "ruleTag": "",
                "ruleDataType": "STRING",
                "ruleSetDataType": null,
                "ruleSetTag": null,
                "stopExecution": false,
                "executionOrder": 2,
                "variableMapping": [
                    {
                        "variableName": "primaryVar",
                        "fieldId": 11675,
                        "fieldTechnicalName": "/Add/STATE",
                        "varType": null
                    }
                ],
                "expressionPayload": "{\"group\":{\"operator\":\"AND\",\\
↪\"rules\": [{\"data\": \"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\"}], \"firstGroup\\
↪\": true}}\"
            }
        ],
        "cdcColumn": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "privRating",
                        "value": "0",
                        "managed": false
                    },
                    {
                        "id": "alias",
                        "value": "Uniqueid",
                        "managed": false
                    }
                ],
                "source": {
                    "type": "BR",
                    "name": "Zaloni Bedrock"
                }
            }
        ]
    },
    "source": {
        "type": "BR",
        "name": "Zaloni Bedrock"
    }
}

```

```

    },
    "complexType": false
  },
  {
    "fieldId": -3,
    "position": 3,
    "fieldTechnicalName": "/who_tweets/tweet_types",
    "dataTypeId": 4,
    "fieldDataType": "STRING",
    "startOffset": null,
    "endOffset": null,
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "tweet_types",
    "fieldDataTypeSchema": "",
    "primary": false,
    "searchable": false,
    "masked": false,
    "maskingPattern": "",
    "tokenized": false,
    "tokenizerAlgorithmId": 0,
    "managedListId": 0,
    "managedListName": "",
    "tokenizerAlgorithmName": null,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "ruleMappings": [
      {
        "ruleSetId": null,
        "ruleId": 8,
        "ruleName": "SimpleRule_API",
        "ruleSetName": null,
        "ruleTag": "",
        "ruleDataType": "STRING",
        "ruleSetDataType": null,
        "ruleSetTag": null,
        "stopExecution": false,
        "executionOrder": 2,
        "variableMapping": [
          {
            "variableName": "primaryVar",
            "fieldId": 11675,
            "fieldTechnicalName": "/Add/STATE",
            "varType": null
          }
        ],
        "expressionPayload": "{\"group\":{\"operator\":\"AND\", \"
↪\"rules\": [{\"data\": \"EQUALS_STRING(primaryVar, \\\"ZPL\\\")\"}], \"firstGroup\"
↪\": true}}\"
      }
    ],
    "cdcColumn": false,
    "customAttributes": [
      {
        "attributes": [

```

```

    "id": "privRating",
    "value": "0",
    "managed": false
  },
  {
    "id": "alias",
    "value": "Uniqueid",
    "managed": false
  }
],
"source": {
  "type": "BR",
  "name": "Zaloni Bedrock"
}
},
"complexType": false
},
{
  "fieldId": -4,
  "position": 4,
  "fieldTechnicalName": "who_tweets",
  "dataTypeId": 22,
  "fieldDataType": "RECORD",
  "startOffset": null,
  "endOffset": null,
  "fieldDataLength": "",
  "dataScale": 0,
  "dataFormat": "",
  "fieldBusinessName": "who_tweets",
  "fieldDataTypeSchema": "{\"id\":22,\"name\":\"who_tweets\\",
  "\": \"type\\\": \"record\\\", \"fields\\\": [{\"id\\\":4, \"name\\\": \"category\\\", \"type\\\": \"
  \"STRING\\\"}, {\"id\\\":4, \"name\\\": \"tweet_types\\\", \"type\\\": \"STRING\\\"}]}\",
  "primary": false,
  "searchable": false,
  "masked": false,
  "maskingPattern": "",
  "tokenized": false,
  "tokenizerAlgorithmId": 0,
  "managedListId": 0,
  "managedListName": "",
  "tokenizerAlgorithmName": null,
  "sensitivityId": 0,
  "sensitivityValue": null,
  "ruleMappings": [
    {
      "ruleSetId": null,
      "ruleId": 8,
      "ruleName": "SimpleRule_API",
      "ruleSetName": null,
      "ruleTag": "",
      "ruleDataType": "STRING",
      "ruleSetDataType": null,
      "ruleSetTag": null,
      "stopExecution": false,
      "executionOrder": 2,

```

```

        "variableMapping": [
            {
                "variableName": "primaryVar",
                "fieldId": 11675,
                "fieldTechnicalName": "/Add/STATE",
                "varType": null
            }
        ],
        "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"
↪"rules\\": [{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\
↪":true}}"
    },
    "cdcColumn": false,
    "customAttributes": [
        {
            "attributes": [
                {
                    "id": "privRating
↪",
                    "value": "0",
                    "managed": false
                },
                {
                    "id": "alias",
                    "value": "Uniqueid
↪",
                    "managed": false
                }
            ],
            "source": {
                "type": "BR",
                "name": "Zaloni Bedrock"
            }
        }
    ],
    "complexType": false
},
"partitionFields": [],
"businessName": "JSON_T",
"technicalName": "JSON_B",
"description": "",
"subjectArea": "",
"label": "",
"dataStoreId": 2,
"dataFileFormatId": 6,
"delimiter": "",
"owner": "regular_user",
"complexEntityVersion": null,
"recordLength": null,
"frequencyOfUpdate": "",
"sourcePlatform": "JSON_SP",
"sourceSchema": "JSON_SS",
"loadType": "",
"hasInActiveTable": false,
"ruleSetId": 0,
"isComplexEntity": true,

```

```

    "serdeInfoProperties": {
        "": ""
    },
    "libClasspath": "",
    "serdeLib": "",
    "inputFormatLib": "",
    "outputFormatLib": "",
    "storageDescriptorProperties": {
        "": ""
    },
    "hcatTableId": 0,
    "entityClusterLocation": [
        {
            "clusterId": "192_168_1_39"
        }
    ],
    "externalDataPath": "",
    "tableType": "EXTERNAL",
    "tableName": "",
    "targetSchema": "aa",
    "fileFormat": "TextFile",
    "tableProperties":
        {
            "skip.header.line.count": "0",
            "myKey": "myValue"
        },
    "cdcActionable": true
}

```

- Scenario 5:

To create a *PARQUET* entity:

```

{
    "entityTypeId": 0,
    "version": 0,
    "newVersionRequired": false,
    "createdBy": "admin",
    "modifiedBy": "admin",
    "owner": "do2user1",
    "fields": [
        {
            "fieldId": -1,
            "position": 0,
            "fieldTechnicalName": "id",
            "dataTypeId": 10,
            "fieldDataType": "INT",
            "startOffset": null,
            "endOffset": null,
            "fieldDataLength": "",
            "dataScale": 0,
            "dataFormat": "",
            "fieldBusinessName": "id",
            "fieldDataTypeSchema": "{\"type\":\"int\",\"id\":10,\\"
            ↪ "name\":\"id\"}",
            "primary": false,
            "searchable": false,
            "masked": false,

```

```

"maskingPattern": "",
"tokenized": false,
"tokenizerAlgorithmId": 0,
"managedListId": 0,
"managedListName": "",
"tokenizerAlgorithmName": null,
"sensitivityId": 0,
"sensitivityValue": null,
"ruleMappings": [
  {
    "ruleSetId": null,
    "ruleId": 8,
    "ruleName": "SimpleRule_API",
    "ruleSetName": null,
    "ruleTag": "",
    "ruleDataType": "STRING",
    "ruleSetDataType": null,
    "ruleSetTag": null,
    "stopExecution": false,
    "executionOrder": 2,
    "variableMapping": [
      {
        "variableName": "primaryVar",
        "fieldId": 11675,
        "fieldTechnicalName": "/Add/STATE",
        "varType": null
      }
    ],
    "expressionPayload": "{\"group\":{\"operator\":\"AND\", \"
↪\"rules\": [{\"data\": \"EQUALS_STRING(primaryVar, \\\"ZPL\\\")\"}], \"firstGroup\"
↪\": true}}"
  }
],
"cdcColumn": false,
"customAttributes": [
  {
    "attributes": [
      {
        "id": "privRating",
        "value": "0",
        "managed": false
      },
      {
        "id": "alias",
        "value": "Uniqueid",
        "managed": false
      }
    ],
    "source": {
      "type": "BR",
      "name": "Zaloni Bedrock"
    }
  }
],
"complexType": false
},
{
  "fieldId": -3,

```

```

"position": 1,
"fieldTechnicalName": "/record/email",
"dataTypeId": 7,
"fieldDataType": "STRING",
"startOffset": null,
"endOffset": null,
"fieldDataLength": "",
"dataScale": 0,
"dataFormat": "",
"fieldBusinessName": "email",
"fields": [],
"fieldDataTypeSchema": "{\"type\":\"string\",\"id\":7,\\"
↪ "name\":\"email\"}",
"primary": false,
"searchable": false,
"masked": false,
"maskingPattern": "",
"tokenized": false,
"tokenizerAlgorithmId": 0,
"managedListId": 0,
"managedListName": "",
"tokenizerAlgorithmName": null,
"sensitivityId": 0,
"sensitivityValue": null,
"ruleMappings": [
  {
    "ruleSetId": null,
    "ruleId": 8,
    "ruleName": "SimpleRule_API",
    "ruleSetName": null,
    "ruleTag": "",
    "ruleDataType": "STRING",
    "ruleSetDataType": null,
    "ruleSetTag": null,
    "stopExecution": false,
    "executionOrder": 2,
    "variableMapping": [
      {
        "variableName": "primaryVar",
        "fieldId": 11675,
        "fieldTechnicalName": "/Add/STATE",
        "varType": null
      }
    ],
    "expressionPayload": "{\"group\":{\"operator\":\"AND\",\\\"rules\\\": [{\\\"data\\\": \\\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\\\"}],\\\"firstGroup\\\":true}}\"
↪ ":true}}"
  }
],
"cdcColumn": false,
"customAttributes": [
  {
    "attributes": [
      {
        "id": "privRating",
        "value": "0",
        "managed": false
      }
    ]
  }
]

```

```

        {
            "id": "alias",
            "value": "Uniqueid",
            "managed": false
        }
    ],
    "source": {
        "type": "BR",
        "name": "Zaloni Bedrock"
    }
}
],
"complexType": false
},
{
    "fieldId": -4,
    "position": 2,
    "fieldTechnicalName": "/record/nickname",
    "dataTypeId": 7,
    "fieldDataType": "STRING",
    "startOffset": null,
    "endOffset": null,
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "nickname",
    "fieldDataTypeSchema": "{\"type\":\"string\",\"id\":7,\
↪"name\":\"nickname\"}",
    "primary": false,
    "searchable": false,
    "masked": false,
    "maskingPattern": "",
    "tokenized": false,
    "tokenizerAlgorithmId": 0,
    "managedListId": 0,
    "managedListName": "",
    "tokenizerAlgorithmName": null,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "ruleMappings": [
        {
            "ruleSetId": null,
            "ruleId": 8,
            "ruleName": "SimpleRule_API",
            "ruleSetName": null,
            "ruleTag": "",
            "ruleDataType": "STRING",
            "ruleSetDataType": null,
            "ruleSetTag": null,
            "stopExecution": false,
            "executionOrder": 2,
            "variableMapping": [
                {
                    "variableName": "primaryVar",
                    "fieldId": 11675,
                    "fieldTechnicalName": "/Add/STATE",
                    "varType": null
                }
            ]
        }
    ]
}

```



```

    ],
    "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\
↪\\"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\
↪":true}}"
  },
  ],
  "cdcColumn": false,
  "customAttributes": [
    {
      "attributes": [
        {
          "id": "privRating",
          "value": "0",
          "managed": false
        },
        {
          "id": "alias",
          "value": "Uniqueid",
          "managed": false
        }
      ],
      "source": {
        "type": "BR",
        "name": "Zaloni Bedrock"
      }
    }
  ],
  "complexType": false
},
{
  "fieldId": -2,
  "position": 3,
  "fieldTechnicalName": "profile",
  "dataTypeId": 18,
  "fieldDataType": "RECORD",
  "startOffset": null,
  "endOffset": null,
  "fieldDataLength": "",
  "dataScale": 0,
  "dataFormat": "",
  "fieldBusinessName": "profile",
  "fieldDataTypeSchema": "{\\"type\\":\\"record\\",\\"id\\":18,\
↪\\"fields\\":[{\\"type\\":\\"string\\",\\"id\\":7,\\"name\\":\\"email\\"},{\\"type\\":\\"string\\
↪",\\"id\\":7,\\"name\\":\\"nickname\\"}],\\"name\\":\\"profile\\"}",
  "primary": false,
  "searchable": false,
  "masked": false,
  "maskingPattern": "",
  "tokenized": false,
  "tokenizerAlgorithmId": 0,
  "managedListId": 0,
  "managedListName": "",
  "tokenizerAlgorithmName": null,
  "sensitivityId": 0,
  "sensitivityValue": null,
  "ruleMappings": [
    {
      "ruleSetId": null,

```

```

        "ruleId": 8,
        "ruleName": "SimpleRule_API",
        "ruleSetName": null,
        "ruleTag": "",
        "ruleDataType": "STRING",
        "ruleSetDataType": null,
        "ruleSetTag": null,
        "stopExecution": false,
        "executionOrder": 2,
        "variableMapping": [
            {
                "variableName": "primaryVar",
                "fieldId": 11675,
                "fieldTechnicalName": "/Add/STATE",
                "varType": null
            }
        ],
        "expressionPayload": "{\"group\":{\"operator\":\"AND\",\\
→"rules\":{\"data\":{\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\"},\"firstGroup\\
→":true}}\"
    },
    "cdcColumn": false,
    "customAttributes": [
        {
            "attributes": [
                {
                    "id": "privRating",
                    "value": "0",
                    "managed": false
                },
                {
                    "id": "alias",
                    "value": "Uniqueid",
                    "managed": false
                }
            ],
            "source": {
                "type": "BR",
                "name": "Zaloni Bedrock"
            }
        }
    ],
    "complexType": false
},
"partitionFields": [],
"businessName": "RC",
"technicalName": "rca",
"description": "",
"subjectArea": "",
"label": "",
"dataStoreId": 2,
"dataFileFormatId": 10,
"delimiter": "",
"complexEntityVersion": null,
"recordLength": null,
"frequencyOfUpdate": "",

```

```

    "sourcePlatform": "2304BP",
    "sourceSchema": "3631H",
    "loadType": "",
    "hasInActiveTable": false,
    "ruleSetId": 0,
    "serdeInfoProperties": {
        "": ""
    },
    "libClasspath": "",
    "serdeLib": "",
    "inputFormatLib": "",
    "outputFormatLib": "",
    "hcatTableId": 0,
    "tableName": "",
    "targetSchema": "test3",
    "entityClusterLocation": [
        {
            "clusterId": "192_168_1_39"
        }
    ],
    "externalDataPath": "/user/do2user1/werth",
    "external": true,
    "tableType": "EXTERNAL",
    "fileFormat": "TextFile",
    "tableProperties": {
        "skip.header.line.count": "0",
        "myKey": "myValue"
    },
    "cdcActionable": true
}

```

Response Code:

201 indicating successful creation. For more details on API response codes, refer to the [API Response Codes](#) section.

Example Response:

- Scenario 1:

If an entity is saved successfully and the *validateRequest* parameter is set as *false* :

```

{
    "responseMessage": "Entity type saved successfully with entity id: 121 and_
↪version: 1",
    "restUri": null,
    "result": {
        "entityTypeId": 11,
        "version": 1,
        "ruleSetId": 0,
        "sharedWith": null,
        "shared": null,
        "ownerProjectId": null,
        "ownerProjectName": null,
        "ruleSetName": null,
        "businessName": "chiva",
        "technicalName": "shba",
        "description": "",
        "subjectArea": "",

```

```

"label": "",
"dataStoreId": 2,
"dataStoreName": null,
"frequencyOfUpdate": "",
"sourcePlatform": "2304453",
"sourceSchema": "36389911",
"loadType": "",
"preQueryStatements": null,
"dataFileFormatId": 10,
"dataFileFormat": "PARQUET",
"metadata": null,
"recordLength": null,
"createdBy": "admin",
"createdTime": "07/24/2017 19:30:03",
"createHiveTable": true,
"customTable": false,
"modifiedBy": "admin",
"modifiedTime": "07/24/2017 19:30:03",
"ingestionTime": null,
"newVersionRequired": false,
"syncedWithHive": false,
"securityContext": {
  "loggedInUser": "admin",
  "doAsUser": "",
  "projectIds": [
    1
  ],
  "artifactPermissions": null,
  "impersonationOn": false,
  "powerUser": false,
  "validImpersonationRequest": true
},
"isDelete": null,
"blockDerivedEntityUpdate": true,
"derived": false,
"fields": [
  {
    "fieldId": 37,
    "position": 1,
    "fieldTechnicalName": "id",
    "dataTypeId": 10,
    "fieldDataType": "INT",
    "fieldDataTypeSchema": "{\"type\":\"int\",\"id\"
↪":10,\"name\":\"id\"}",
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "id",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": "",
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,

```

```

"algoArguments": null,
"maskingPattern": "",
"startOffset": 0,
"endOffset": 0,
"sensitivityId": 0,
"sensitivityValue": null,
"ruleMappings": [
  {
    "ruleSetId": null,
    "ruleId": 8,
    "ruleName": "SimpleRule_API",
    "ruleSetName": null,
    "ruleTag": "",
    "ruleDataType": "STRING",
    "ruleSetDataType": null,
    "ruleSetTag": null,
    "stopExecution": false,
    "executionOrder": 2,
    "variableMapping": [
      {
        "variableName": "primaryVar",
        "fieldId": 11675,
        "fieldTechnicalName": "/Add/STATE",
        "varType": null
      }
    ],
    "expressionPayload": "{\"group\":{\"operator\":\
↪\"AND\", \"rules\": [{\"data\": \"EQUALS_STRING(primaryVar, \\\"ZPL\\\")\"}], \
↪\"firstGroup\": true}}\"
  },
  {
    "cdcColumn": false,
    "customAttributes": [
      {
        "attributes": [
          {
            "id": "privRating",
            "value": "0",
            "managed": false
          },
          {
            "id": "alias",
            "value": "Uniqueid",
            "managed": false
          }
        ],
        "source": {
          "type": "BR",
          "name": "Zaloni Bedrock"
        }
      }
    ],
    "complexType": false
  },
  {
    "fieldId": 35,
    "position": 2,

```

```

"fieldTechnicalName": "/record/email",
"dataTypeId": 7,
"fieldDataType": "STRING",
"fieldDataTypeSchema": "{\"type\":\"string\",\"id\
↪":7,\"name\":\"email\"}",
"fieldDataLength": "",
"dataScale": 0,
"dataFormat": "",
"fieldBusinessName": "email",
"description": null,
"primary": false,
"searchable": false,
"masked": false,
"tokenized": false,
"nonTopLevel": true,
"managedListId": 0,
"managedListName": "",
"tokenizerAlgorithmId": 0,
"tokenizerAlgorithmName": null,
"algoArguments": null,
"maskingPattern": "",
"startOffset": 0,
"endOffset": 0,
"sensitivityId": 0,
"sensitivityValue": null,
"ruleMappings": [
  {
    "ruleSetId": null,
    "ruleId": 8,
    "ruleName": "SimpleRule_API",
    "ruleSetName": null,
    "ruleTag": "",
    "ruleDataType": "STRING",
    "ruleSetDataType": null,
    "ruleSetTag": null,
    "stopExecution": false,
    "executionOrder": 2,
    "variableMapping": [
      {
        "variableName": "primaryVar",
        "fieldId": 11675,
        "fieldTechnicalName": "/Add/STATE",
        "varType": null
      }
    ],
    "expressionPayload": "{\"group\":{\"operator\":\
↪\"AND\",\"rules\": [{\"data\": \"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\"}],\
↪\"firstGroup\":true}}"
  }
],
"cdcColumn": false,
"customAttributes": [
  {
    "attributes": [
      {
        "id": "privRating",
        "value": "0",
        "managed": false
      }
    ]
  }
]

```

```

        },
        {
            "id": "alias",
            "value": "Uniqueid",

            "managed": false
        }
    ],
    "source": {
        "type": "BR",
        "name": "Zaloni Bedrock"
    }
}

],
"complexType": false
},
{
    "fieldId": 34,
    "position": 3,
    "fieldTechnicalName": "/record/nickname",
    "dataTypeId": 7,
    "fieldDataType": "STRING",
    "fieldDataTypeSchema": "{\"type\":\"string\",\"id\":"
    ↪":7,\"name\":\"nickname\"}",
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "nickname",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": true,
    "managedListId": 0,
    "managedListName": "",
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,
    "algoArguments": null,
    "maskingPattern": "",
    "startOffset": 0,
    "endOffset": 0,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "ruleMappings": [
        {
            "ruleSetId": null,
            "ruleId": 8,
            "ruleName": "SimpleRule_API",
            "ruleSetName": null,
            "ruleTag": "",
            "ruleDataType": "STRING",
            "ruleSetDataType": null,
            "ruleSetTag": null,
            "stopExecution": false,
            "executionOrder": 2,
            "variableMapping": [

```

```

        "variableName": "primaryVar",
        "fieldId": 11675,
        "fieldTechnicalName": "/Add/STATE",
        "varType": null
    },
    ],
    "expressionPayload": "{ \"group\": { \"operator\": \"AND\", \"rules\": [ { \"data\": \"EQUALS_STRING(primaryVar, '\\\"ZPL\\\"')\" }, { \"firstGroup\": true } ] } }",
    ],
    "cdcColumn": false,
    "customAttributes": [
        {
            "attributes": [
                {
                    "id": "privRating",
                    "value": "0",
                    "managed": false
                },
                {
                    "id": "alias",
                    "value": "Uniqueid",
                    "managed": false
                }
            ],
            "source": {
                "type": "BR",
                "name": "Zaloni Bedrock"
            }
        }
    ],
    "complexType": false
},
{
    "fieldId": 36,
    "position": 4,
    "fieldTechnicalName": "profile",
    "dataTypeId": 18,
    "fieldDataType": "RECORD",
    "fieldDataTypeSchema": "{ \"type\": \"record\", \"id\": 18, \"fields\": [ { \"type\": \"string\", \"id\": 7, \"name\": \"email\" }, { \"type\": \"string\", \"id\": 7, \"name\": \"nickname\" }, { \"name\": \"profile\" } ] }",
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "profile",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": "",
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,

```



```

"algoArguments": null,
"maskingPattern": "",
"startOffset": 0,
"endOffset": 0,
"sensitivityId": 0,
"sensitivityValue": null,
"ruleMappings": [
  {
    "ruleSetId": null,
    "ruleId": 8,
    "ruleName": "SimpleRule_API",
    "ruleSetName": null,
    "ruleTag": "",
    "ruleDataType": "STRING",
    "ruleSetDataType": null,
    "ruleSetTag": null,
    "stopExecution": false,
    "executionOrder": 2,
    "variableMapping": [
      {
        "variableName": "primaryVar",
        "fieldId": 11675,
        "fieldTechnicalName": "/Add/STATE",
        "varType": null
      }
    ],
    "expressionPayload": "{\"group\":{\"operator\":\
↪\"AND\", \"rules\": [{\"data\": \"EQUALS_STRING(primaryVar, \\\"ZPL\\\")\"}], \
↪\"firstGroup\": true}}\"
  }
],
"cdcColumn": false,
"customAttributes": [
  {
    "attributes": [
      {
        "id": "privRating",
        "value": "0",
        "managed": false
      },
      {
        "id": "alias",
        "value": "Uniqueid
↪",
        "managed": false
      }
    ],
    "source": {
      "type": "BR",
      "name": "Zaloni Bedrock"
    }
  }
],
"complexType": false
}
],
"partitionFields": [],
"ruleSetEntityMapping": null,

```

```

    "complexMetadataEntityMapping": [],
    "complexEntityVersion": null,
    "hcatTableId": 7,
    "tableName": "2304453_36389911_shba_1",
    "targetSchema": "chivas",
    "targetSchemaValue": "chivas",
    "delimiter": "",
    "lineDelimiter": null,
    "external": false,
    "externalDataPath": "/user/bedrock/wer",
    "externalDataPathValue": "/user/bedrock/wer",
    "fileFormat": "TextFile",
    "serdeLib": "",
    "libClasspath": "",
    "inputFormatLib": "",
    "outputFormatLib": "",
    "tableProperties":
      {
        "skip.header.line.count": "0",
        "myKey": "myValue"
      },
    "storageDescriptorProperties": {},
    "serdeInfoProperties": {
      "": ""
    },
    "cdcActionable": true,
    "hasInactiveTable": false,
    "entityClusterLocation": [
      {
        "clusterId": "192_168_1_39"
      }
    ],
    "changeDetected": false,
    "validateRequest": false,
    "latestVersion": true,
    "migrate": false,
    "tableType": "EXTERNAL",
    "skewing": null,
    "bucketing": null,
    "view": null,
    "retainExistingAdditionalAttributes": false,
    "import": false,
    "syncJob": false
  },
  "page": null
}

```

- Scenario 2:

If an entity is saved successfully and the *validateRequest* parameter is set as *true* :

```

{
  "responseMessage": "Entity type saved successfully with entity id: 18 and_
↪version: 1",
  "restUri": null,
  "result": "Entity type saved successfully with entity id: 18 and version: 1",
  "page": null
}

```

Response in Case of Failure:

- Scenario 1:

If the data type of a custom field is undefined; or parameter value is unrecognized:

```
{
  "responseMessage": "Errors while parsing field :<email> :Invalid data type for field; #Errors while parsing field :<nickname> :Invalid data type for field; #",
  "restUri": "/bedrock-app/services/rest/projects/<project_id>/entities",
  "result": null,
  "page": null
}
```

- Scenario 2:

If a similar entity exists in the target instance:

```
{
  "responseMessage": "Unable to save entity. Entity type with same [technical name, source schema and source platform] or [target schema and table name] already exists",
  "restUri": "/bedrock-app/services/rest/projects/<project_id>/entities",
  "result": null,
  "page": null
}
```

1.6.2.2 Update an entity

This API allows you to update an entity type. For updating the entity type, the request payload must have all the basic attributes of the existing entity as described under the *Create/Update an Entity* section. For example,

- To add a new field to an existing entity, add the field details to the request payload of the existing entity.
- To remove a field, remove the particular field details from the request payload.

Before using this service, the *Fetch Details of an Entity* API can be used to get the details of the Entity being updated.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/entities/{entity_type_Id}?
↳projectIds={project_Id}
```

Note: You must pass the *entity_type_Id* and *project_Id* parameters in the API URL. To update an entity under a public project, set the *project_Id* parameter to 1.

Method:

```
PUT
```

Example Request:

- Scenario 1:

To update a *DELIMITED* entity:

```

{
  "entityTypeId": 4,
  "version": 1,
  "newVersionRequired": false,
  "fields": [
    {
      "fieldId": 11,
      "position": 1,
      "fieldTechnicalName": "Id",
      "dataTypeId": 10,
      "fieldDataType": "INT",
      "fieldDataLength": "",
      "dataScale": 0,
      "dataFormat": "",
      "fieldBusinessName": "Id",
      "description": null,
      "primary": false,
      "searchable": false,
      "masked": false,
      "tokenized": false,
      "nonTopLevel": false,
      "managedListId": 0,
      "managedListName": null,
      "tokenizerAlgorithmId": 0,
      "tokenizerAlgorithmName": null,
      "maskingPattern": "",
      "startOffset": 0,
      "endOffset": 0,
      "sensitivityId": 0,
      "sensitivityValue": null,
      "ruleMappings": [
        {
          "ruleSetId": null,
          "ruleId": 8,
          "ruleName": "SimpleRule_API",
          "ruleSetName": null,
          "ruleTag": "",
          "ruleDataType": "STRING",
          "ruleSetDataType": null,
          "ruleSetTag": null,
          "stopExecution": false,
          "executionOrder": 2,
          "variableMapping": [
            {
              "variableName": "primaryVar",
              "fieldId": 11675,
              "fieldTechnicalName": "/Add/STATE",
              "varType": null
            }
          ],
          "expressionPayload": "{\"group\":{\"operator\":\"AND\", \"
↪\"rules\":[{\"data\":\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\"}],\"firstGroup\"
↪\":true}}\"
        }
      ],
      "cdcColumn": false,
      "customAttributes": [
        {

```

```

        "attributes": [
            {
                "id": "privRating",
                "value": "0",
                "managed": false
            },
            {
                "id": "alias",
                "value": "Uniqueid",
                "managed": false
            }
        ],
        "source": {
            "type": "BR",
            "name": "Zaloni Bedrock"
        }
    },
    "complexType": false
},
{
    "fieldId": 10,
    "position": 3,
    "fieldTechnicalName": "Target",
    "dataTypeId": 7,
    "fieldDataType": "STRING",
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "Target",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": null,
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,
    "maskingPattern": "",
    "startOffset": 0,
    "endOffset": 0,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "ruleMappings": [
        {
            "ruleSetId": null,
            "ruleId": 8,
            "ruleName": "SimpleRule_API",
            "ruleSetName": null,
            "ruleTag": "",
            "ruleDataType": "STRING",
            "ruleSetDataType": null,
            "ruleSetTag": null,
            "stopExecution": false,

```

```

        "executionOrder": 2,
        "variableMapping": [
            {
                "variableName": "primaryVar",
                "fieldId": 11675,
                "fieldTechnicalName": "/Add/STATE",
                "varType": null
            }
        ],
        "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"
↪"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\
↪":true}}"
    }

    ],
    "cdcColumn": false,
    "customAttributes": [
        {
            "attributes": [
                {
                    "id": "privRating
↪",
                    "value": "0",
                    "managed": false
                },
                {
                    "id": "alias",
                    "value": "Uniqueid
↪",
                    "managed": false
                }
            ],
            "source": {
                "type": "BR",
                "name": "Zaloni Bedrock"
            }
        }
    ],
    "complexType": false
}

],
"partitionFields": [],
"complexMetadataEntityMapping": [],
"validateRequest": false,
"businessName": "Zone4",
"technicalName": "Zone4",
"description": "",
"subjectArea": "",
"label": "",
"dataStoreId": 2,
"dataFileFormatId": 1,
"delimiter": ",",
"entityCreateOptions": 1,
"complexEntityVersion": null,
"recordLength": null,
"frequencyOfUpdate": "",
"sourcePlatform": "spz4",
"sourceSchema": "ssz4",
"loadType": "",

```

```

    "hasInactiveTable": false,
    "ruleSetId": 0,
    "isComplexEntity": false,
    "serdeInfoProperties": {
        "": ""
    },
    "libClasspath": "",
    "serdeLib": "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
    "inputFormatLib": "org.apache.hadoop.mapred.TextInputFormat",
    "outputFormatLib": "org.apache.hadoop.hive.ql.io.
    ↪HiveIgnoreKeyTextOutputFormat",
    "hcatTableId": 4,
    "tableName": "spz4_ssz4_Zone4_1",
    "targetSchema": "zonesp",
    "external": true,
    "externalDataPath": "/user/bedrock/tweet23",
    "fileFormat": "TextFile",
    "tableProperties":
        {
            "skip.header.line.count": "0",
            "myKey": "myValue"
        },
    "derived": false,
    "position": 1,
    "entityClusterLocation": [
        {
            "clusterId": "192_168_1_39"
        }
    ],
    "tableType": "EXTERNAL",
    "cdcActionable": true
}

```

- Scenario 2:

To update an AVRO entity over the leaf node to associate a data quality rule and update the business name.

```

{
    "version": 1,
    "ruleSetId": 0,
    "businessName": "REMS",
    "technicalName": "remson",
    "description": "",
    "subjectArea": "",
    "label": "",
    "dataStoreId": 2,
    "sourcePlatform": "electronics",
    "sourceSchema": "appliances",
    "dataFileFormatId": 8,
    "dataFileFormat": "AVRO",
    "createdBy": "admin",
    "owner": "do2user1",
    "fields": [
        {
            "fieldId": 11678,
            "position": 1,
            "fieldTechnicalName": "id",
            "dataTypeId": 10,

```

```

        "fieldDataType": "INT",
        "fieldDataTypeSchema": "{\"type\":\"int\",\"id\":10,\\"
↪ "name\":\"id\"}",
        "fieldBusinessName": "ID",
        "nonTopLevel": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "",
                        "value": "",
                        "managed": false
                    }
                ],
                "source": {
                    "type": "BR",
                    "name": "Zaloni Bedrock"
                }
            }
        ],
        "complexType": false
    },
    {
        "fieldId": 11676,
        "position": 2,
        "fieldTechnicalName": "Name",
        "dataTypeId": 7,
        "fieldDataType": "STRING",
        "fieldDataTypeSchema": "{\"type\":\"string\",\"id\":7,\\"
↪ "name\":\"Name\"}",
        "fieldBusinessName": "Name",
        "description": "",
        "primary": false,
        "nonTopLevel": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "",
                        "value": "",
                        "managed": false
                    }
                ],
                "source": {
                    "type": "BR",
                    "name": "Zaloni Bedrock"
                }
            }
        ],
        "complexType": false
    },
    {
        "fieldId": 11677,
        "position": 3,
        "fieldTechnicalName": "/Add/Street",
        "dataTypeId": 7,
        "fieldDataType": "STRING",
        "fieldDataTypeSchema": "{\"type\":\"string\",\"id\":7,\\"
↪ "name\":\"Street\"}",

```



```

        "fieldBusinessName": "Street",
        "description": "",
        "primary": false,
        "nonTopLevel": true,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "",
                        "value": "",
                        "managed": false
                    }
                ],
                "source": {
                    "type": "BR",
                    "name": "Zaloni Bedrock"
                }
            }
        ],
        "complexType": false
    },
    {
        "fieldId": 11679,
        "position": 4,
        "fieldTechnicalName": "/Add/PIN",
        "dataTypeId": 27,
        "fieldDataType": "BIGINTEGER",
        "fieldDataTypeSchema": "{\"type\":\"biginteger\",\"id\":"
↪":27,\"name\":\"PIN\"}",
        "fieldBusinessName": "PIN",
        "description": "",
        "primary": false,
        "nonTopLevel": true,
        "cdcColumn": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "",
                        "value": "",
                        "managed": false
                    }
                ],
                "source": {
                    "type": "BR",
                    "name": "Zaloni Bedrock"
                }
            }
        ],
        "complexType": false
    },
    {
        "fieldId": 11674,
        "position": 5,
        "fieldTechnicalName": "/Add/DIST",
        "dataTypeId": 7,
        "fieldDataType": "STRING",
        "fieldDataTypeSchema": "{\"type\":\"string\",\"id\":7,\""
↪"name\":\"DIST\"}",

```

```

"fieldDataLength": "",
"dataScale": 0,
"dataFormat": "",
"fieldBusinessName": "DIST",
"description": "",
"primary": false,
"searchable": false,
"masked": false,
"tokenized": false,
"nonTopLevel": true,
"managedListId": 0,
"managedListName": "",
"tokenizerAlgorithmId": 0,
"tokenizerAlgorithmName": null,
"algoArguments": null,
"maskingPattern": "",
"startOffset": 0,
"endOffset": 0,
"sensitivityId": 0,
"sensitivityValue": null,
"ruleMappings": [
  {
    "ruleSetId": null,
    "ruleId": 8,
    "ruleName": "SimpleRule_API",
    "ruleSetName": null,
    "ruleTag": "",
    "ruleDataType": "STRING",
    "ruleSetDataType": null,
    "ruleSetTag": null,
    "stopExecution": false,
    "executionOrder": 2,
    "variableMapping": [
      {
        "variableName": "primaryVar",
        "fieldId": 11675,
        "fieldTechnicalName": "/Add/STATE",
        "varType": null
      }
    ],
    "expressionPayload": "{\"group\":{\"operator\":\"AND\", \"
↪\"rules\": [{\"data\": \"EQUALS_STRING(primaryVar, \\\"ZPL\\\")\"}, {\"firstGroup\"
↪\": true}]}"
  }
],
"cdcColumn": false,
"customAttributes": [
  {
    "attributes": [
      {
        "id": "",
        "value": "",
        "managed": false
      }
    ],
    "source": {
      "type": "BR",
      "name": "Zaloni Bedrock"
    }
  }
]

```

```

    }
  },
  "complexType": false
},
{
  "fieldId": 11675,
  "position": 6,
  "fieldTechnicalName": "/Add/STATE",
  "dataTypeId": 7,
  "fieldDataType": "STRING",
  "fieldDataTypeSchema": "{\"type\":\"string\",\"id\":7,\"
↪ "name\":\"STATE\"}",
  "fieldDataLength": "",
  "dataScale": 0,
  "dataFormat": "",
  "fieldBusinessName": "STATE LOCATION",
  "description": "",
  "primary": false,
  "searchable": false,
  "masked": false,
  "tokenized": false,
  "nonTopLevel": true,
  "managedListId": 0,
  "managedListName": "",
  "tokenizerAlgorithmId": 0,
  "tokenizerAlgorithmName": null,
  "algoArguments": null,
  "maskingPattern": "",
  "startOffset": 0,
  "endOffset": 0,
  "sensitivityId": 0,
  "sensitivityValue": null,
  "ruleMappings": [
    {
      "ruleSetId": 3,
      "ruleId": null,
      "ruleName": null,
      "ruleSetName": "Rule_Set_1",
      "ruleTag": null,
      "ruleDataType": null,
      "ruleSetDataType": "STRING",
      "ruleSetTag": null,
      "stopExecution": false,
      "executionOrder": 1,
      "variableMapping": [
        {
          "variableName":
↪ "primaryVariables",
          "fieldId": 116756,
          "fieldTechnicalName": "/"
↪ Add/STATE",
          "varType": null
        }
      ],
      "expressionPayload": ""
    }
  ],
  {

```

```

        "ruleSetId": null,
        "ruleId": 8,
        "ruleName": "SimpleRule_API",
        "ruleSetName": null,
        "ruleTag": "",
        "ruleDataType": "STRING",
        "ruleSetDataType": null,
        "ruleSetTag": null,
        "stopExecution": false,
        "executionOrder": 2,
        "variableMapping": [
            {
                "variableName":
↪ "primaryVar",
                "fieldId": 11675,
                "fieldTechnicalName": "/"
↪ Add/STATE",
                "varType": null
            }
        ],
        "expressionPayload": "{\"group\\\":{\\
↪ \"operator\\\":\\\"AND\\\",\\\"rules\\\":[{\\\"data\\\":\\\"EQUALS_STRING(primaryVar,\\\"ZPL\\\"\\
↪ )\\\"}],\\\"firstGroup\\\":true}}\"
    }
],
    "cdcColumn": false,
    "customAttributes": [
        {
            "attributes": [
                {
                    "id": "",
                    "value": "",
                    "managed": false
                }
            ],
            "source": {
                "type": "BR",
                "name": "Zaloni Bedrock"
            }
        }
    ],
    "complexType": false
},
{
    "fieldId": 11673,
    "position": 7,
    "fieldTechnicalName": "Updated_Customer_Address",
    "dataTypeId": 18,
    "fieldDataType": "RECORD",
    "fieldDataTypeSchema": "{\"type\\\":\\\"record\\\",\\\"id\\\":18,\\
↪ \"fields\\\":[{\\\"type\\\":\\\"string\\\",\\\"id\\\":7,\\\"name\\\":\\\"Street\\\"},{\\\"type\\\":\\
↪ \"biginteger\\\",\\\"id\\\":27,\\\"name\\\":\\\"PIN\\\"},{\\\"type\\\":\\\"string\\\",\\\"id\\\":7,\\\"name\\
↪ \\\":\\\"DIST\\\"},{\\\"type\\\":\\\"string\\\",\\\"id\\\":7,\\\"name\\\":\\\"STATE\\\"}],\\\"name\\\":\\\"Add\\\"}
↪ \",
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "Add",

```

```

    "description": "",
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": "",
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,
    "algoArguments": null,
    "maskingPattern": "",
    "startOffset": 0,
    "endOffset": 0,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "ruleMappings": [
      {
        "ruleSetId": null,
        "ruleId": 8,
        "ruleName": "SimpleRule_API",
        "ruleSetName": null,
        "ruleTag": "",
        "ruleDataType": "STRING",
        "ruleSetDataType": null,
        "ruleSetTag": null,
        "stopExecution": false,
        "executionOrder": 2,
        "variableMapping": [
          {
            "variableName": "primaryVar",
            "fieldId": 11675,
            "fieldTechnicalName": "/Add/STATE",
            "varType": null
          }
        ],
        "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"
↪"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\
↪":true}}"
      }
    ],
    "cdcColumn": false,
    "customAttributes": [
      {
        "attributes": [
          {
            "id": "",
            "value": "",
            "managed": false
          }
        ],
        "source": {
          "type": "BR",
          "name": "Zaloni Bedrock"
        }
      }
    ],
    "complexType": false

```

```

    },
    "tableName": "electronics_appliances_remson_1",
    "targetSchema": "chivas",
    "delimiter": "",
    "external": false,
    "externalDataPath": null,
    "fileFormat": "TextFile",
    "serdeLib": "",
    "libClasspath": "",
    "inputFormatLib": "",
    "outputFormatLib": "",
    "tableProperties": {
        "skip.header.line.count": "0",
        "myKey": "myValue"
    },
    "storageDescriptorProperties": {},
    "serdeInfoProperties": {
        "": ""
    },
    "cdcActionable": false,
    "hasInActiveTable": false,
    "entityClusterLocation": [
        {
            "clusterId": "192_168_1_39"
        }
    ],
    "tableType": "INTERNAL",
    "skewing": {
        "skewingId": 0,
        "storedAsDirectories": false,
        "skewingFields": []
    },
    "bucketing": {
        "bucketingId": 0,
        "numberOfBuckets": 0,
        "bucketingFields": [],
        "sortingFields": []
    }
}

```

- Scenario 3:

To update a *PARQUET* entity:

```

{
    "dataFileFormatId": 10,
    "delimiter": "",
    "sourcePlatform": "sp",
    "sourceSchema": "ss",
    "createdTime": "07/27/2015 15:05:11",
    "createdBy": "admin",
    "entityTypeId": 185,
    "businessName": "APIParq",
    "technicalName": "APIParq",
    "targetSchema": "sauradata",
    "modifiedTime": "07/27/2015 15:05:11",
}

```

```

"modifiedBy": "admin",
"ruleSetName": null,
"partitionFields": [],
"ruleSetId": 0,
"dataFileFormat": "PARQUET",
"inputFormatLib": "",
"outputFormatLib": "",
"tableProperties":
    {
        "skip.header.line.count": "0",
        "myKey": "myValue"
    },
"serdeLib": "",
"recordLength": 0,
"lineDelimiter": null,
"externalDataPath": null,
"libClasspath": "",
"dataStoreId": 2,
"subjectArea": "",
"dataStoreName": "HDFS",
"hcatTableId": 183,
"cdcActionable": true,
"hasInactiveTable": false,
"ruleSetEntityMapping": [],
"derived": false,
"fileFormat": "TextFile",
"isDelete": null,
"preQueryStatements": null,
"frequencyOfUpdate": "",
"loadType": "",
"complexEntityVersion": null,
"label": "",
"tableName": "sp_ss_APIParq_1",
"external": true,
"description": "",
"version": 1,
"fields": [
    {
        "dataFormat": "",
        "primary": false,
        "fieldId": 791,
        "fieldBusinessName": "int BN changed",
        "fieldTechnicalName": "/topRec/int",
        "fieldDataType": "INT",
        "dataTypeId": 14,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "fieldDataTypeSchema": null,
        "nonTopLevel": true,
        "fieldDataLength": "",
        "managedListId": 0,
        "dataScale": 0,
        "maskingPattern": "",
        "tokenizerAlgorithmName": null,
        "managedListName": null,
        "masked": false,
        "tokenized": false,
        "tokenizerAlgorithmId": 0,
    }

```

```

"ruleMappings": [
  {
    "ruleSetId": null,
    "ruleId": 8,
    "ruleName": "SimpleRule_API",
    "ruleSetName": null,
    "ruleTag": "",
    "ruleDataType": "STRING",
    "ruleSetDataType": null,
    "ruleSetTag": null,
    "stopExecution": false,
    "executionOrder": 2,
    "variableMapping": [
      {
        "variableName": "primaryVar",
        "fieldId": 11675,
        "fieldTechnicalName": "/Add/STATE",
        "varType": null
      }
    ],
    "expressionPayload": "{\"group\":{\"operator\":\"AND\",\\
↪\"rules\":{\"data\":{\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\"}},\"firstGroup\\
↪\":true}}\"
  }
],
"cdcColumn": false,
"customAttributes": [
  {
    "attributes": [
      {
        "id": "privRating",
        "value": "0",
        "managed": false
      },
      {
        "id": "alias",
        "value": "Uniqueid",
        "managed": false
      }
    ],
    "source": {
      "type": "BR",
      "name": "Zaloni Bedrock"
    }
  },
  {
    "complexType": false,
    "endOffset": 0,
    "startOffset": 0,
    "position": 0,
    "searchable": false
  },
  {
    "dataFormat": "",
    "primary": false,
    "fieldId": 792,

```



```

        "fieldBusinessName": "recStr",
        "fieldTechnicalName": "/topRec/recStr",
        "fieldDataType": "STRING",
        "dataTypeId": 4,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "fieldDataTypeSchema": null,
        "nonTopLevel": true,
        "fieldDataLength": "",
        "managedListId": 0,
        "dataScale": 0,
        "maskingPattern": "",
        "tokenizerAlgorithmName": null,
        "managedListName": null,
        "masked": false,
        "tokenized": false,
        "tokenizerAlgorithmId": 0,
        "ruleMappings": [
            {
                "ruleName": "avro_area",
                "ruleSetName": null,
                "ruleSetId": null,
                "ruleId": 7,
                "executionOrder": 1,
                "stopExecution": false,
            }
        ],
        "variableMapping": [
            {
                "variableName": "primaryVar",
                "fieldId": 11675,
                "fieldTechnicalName": "/Add/STATE",
                "varType": null
            }
        ],
        "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"rules\\":[{\\
↪\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\\\"}\\),\\"firstGroup\\":true}]}",
        ],
        "cdcColumn": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "privRating",
                        "value": "0",
                        "managed": false
                    },
                    {
                        "id": "alias",
                        "value": "Uniqueid",
                        "managed": false
                    }
                ]
            },
            {
                "source": {
                    "type": "BR",
                    "name": "Zaloni Bedrock"
                }
            }
        ]
    },
    ↪",
    ↪",
    }

```

```

    },
    "complexType": false,
    "endOffset": 0,
    "startOffset": 0,
    "position": 1,
    "searchable": false
  },
  {
    "dataFormat": "",
    "primary": false,
    "fieldId": 790,
    "fieldBusinessName": "TopRec",
    "fieldTechnicalName": "topRec",
    "fieldDataType": "RECORD",
    "dataTypeId": 22,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "fieldDataTypeSchema": "{\"type\":\"record\",\"id\":\"22\",
    ↪"fields\":[{\"type\":\"int\",\"id\":\"14\",\"name\":\"int\"},{\"type\":\"string\",
    ↪"id\":\"4\",\"name\":\"recStr\"}],\"name\":\"topRec\"}",
    "nonTopLevel": false,
    "fieldDataLength": "",
    "managedListId": 0,
    "dataScale": 0,
    "maskingPattern": "",
    "tokenizerAlgorithmName": null,
    "managedListName": null,
    "masked": false,
    "tokenized": false,
    "tokenizerAlgorithmId": 0,
    "ruleMappings": [
      {
        "ruleName": "avro_area",
        "ruleSetName": null,
        "ruleSetId": null,
        "ruleId": 7,
        "executionOrder": 1,
        "stopExecution": false
      }
    ],
    "variableMapping": [
      {
        "variableName": "primaryVar",
        "fieldId": 11675,
        "fieldTechnicalName": "/Add/STATE",
        "varType": null
      }
    ],
    "expressionPayload": "{\"group\":{\"operator\":\"AND\",\"rules\":[{
    ↪"data\":\"EQUALS_STRING(primaryVar,\"\"ZPL\"\" )\"}],\"firstGroup\":true}}",
    },
    "cdcColumn": false,
    "customAttributes": [
      {
        "attributes": [
          {
            "id": "privRating
    ↪",

```

```

        "value": "0",
        "managed": false
    },
    {
        "id": "alias",
        "value": "Uniqueid",
        "managed": false
    }
],
"source": {
    "type": "BR",
    "name": "Zaloni Bedrock"
}
},
"complexType": true,
"endOffset": 0,
"startOffset": 0,
"position": 2,
"searchable": false
},
{
    "dataFormat": "",
    "primary": false,
    "fieldId": 789,
    "fieldBusinessName": "topStr",
    "fieldTechnicalName": "topStr",
    "fieldDataType": "STRING",
    "dataTypeId": 4,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "fieldDataTypeSchema": null,
    "nonTopLevel": false,
    "fieldDataLength": "",
    "managedListId": 0,
    "dataScale": 0,
    "maskingPattern": "",
    "tokenizerAlgorithmName": null,
    "managedListName": null,
    "masked": false,
    "tokenized": false,
    "tokenizerAlgorithmId": 0,
    "ruleMappings": [
        {
            "ruleName": "avro_area",
            "ruleSetName": null,
            "ruleSetId": null,
            "ruleId": 7,
            "executionOrder": 1,
            "stopExecution": false
        }
    ],
    "variableMapping": [
        {
            "variableName": "primaryVar",
            "fieldId": 11675,
            "fieldTechnicalName": "/Add/STATE",
            "varType": null
        }
    ]
}

```

```

    ],
    "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"rules\\":[{\\"
↪"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\":true}}"}
    },
    "cdcColumn": false,
    "customAttributes": [
        {
            "attributes": [
                {
                    "id": "privRating
↪",
                    "value": "0",
                    "managed": false
                },
                {
                    "id": "alias",
                    "value": "Uniqueid
↪",
                    "managed": false
                }
            ],
            "source": {
                "type": "BR",
                "name": "Zaloni Bedrock"
            }
        }
    ],
    "complexType": false,
    "endOffset": 0,
    "startOffset": 0,
    "position": 3,
    "searchable": false
},
"entityClusterLocation": [
    {
        "clusterId": "192_168_1_39"
    }
],
"tableType": "EXTERNAL",
"owner": null
}

```

- Scenario 4:

To add *phone* leaf field to the payload in the **Example Request 4** of *Create an entity* :

```

{
    "entityTypeId": 5,
    "version": 1,
    "newVersionRequired": false,
    "createdBy": "admin",
    "createdTime": "07/30/2015 10:47:01",
    "modifiedBy": "admin",
    "fields": [
        {
            "position": 3,

```

```

        "endOffset": 0,
        "startOffset": 0,
        "dataFormat": "",
        "primary": false,
        "fieldId": 15,
        "fieldBusinessName": "id",
        "fieldTechnicalName": "id",
        "fieldDataType": "INT",
        "dataTypeId": 14,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "nonTopLevel": false,
        "fieldDataTypeSchema": "{\"type\":\"int\",\"id\":\"14,\"
↪ "name\":\"id\"}",
        "fieldDataLength": "",
        "maskingPattern": "",
        "tokenizerAlgorithmName": null,
        "managedListName": null,
        "searchable": false,
        "managedListId": 0,
        "dataScale": 0,
        "masked": false,
        "tokenized": false,
        "tokenizerAlgorithmId": 0,
        "ruleMappings": [
            {
                "ruleName": "avro_area",
                "ruleSetName": null,
                "ruleSetId": null,
                "ruleId": 7,
                "executionOrder": 1,
                "stopExecution": false
            }
        ],
        "variableMapping": [
            {
                "variableName": "primaryVar",
                "fieldId": 11675,
                "fieldTechnicalName": "/Add/STATE",
                "varType": null
            }
        ],
        "expressionPayload": "{\"group\":{\"operator\":\"AND\",\"rules\":[{
↪ "data\":\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\"}],\"firstGroup\":true}}\"
        },
        "cdcColumn": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "privRating",
                        "value": "0",
                        "managed": false
                    },
                    {
                        "id": "alias",
                        "value": "Uniqueid"
                    }
                ]
            }
        ],
        ↪ "
    },
    ↪ "

```

```

"managed": false
    },
    ],
    "source": {
        "type": "BR",
        "name": "Zaloni Bedrock"
    }
},
    ],
    "complexType": false
},
{
    "fieldId": 16,
    "position": 4,
    "fieldTechnicalName": "/profile/email",
    "dataTypeId": 4,
    "fieldDataType": "STRING",
    "startOffset": 0,
    "endOffset": 0,
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "email",
    "fieldDataTypeSchema": "{\"type\":\"string\",\"id\":4,\
↪"name\":\"email\"}",
    "primary": false,
    "searchable": false,
    "masked": false,
    "maskingPattern": "",
    "tokenized": false,
    "tokenizerAlgorithmId": 0,
    "managedListId": 0,
    "managedListName": null,
    "tokenizerAlgorithmName": null,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "ruleMappings": [
        {
            "ruleName": "avro_area",
            "ruleSetName": null,
            "ruleSetId": null,
            "ruleId": 7,
            "executionOrder": 1,
            "stopExecution": false
        }
    ],
    "variableMapping": [
        {
            "variableName": "primaryVar",
            "fieldId": 11675,
            "fieldTechnicalName": "/Add/STATE",
            "varType": null
        }
    ],
    "expressionPayload": "{\"group\":{\"operator\":\"AND\",\"rules\":[{\
↪"data\":\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\"}],\"firstGroup\":true}}",
    ],
    "cdcColumn": false,
    "customAttributes": [

```

```

        {
            "attributes": [
                {
                    "id": "privRating",
                    "value": "0",
                    "managed": false
                },
                {
                    "id": "alias",
                    "value": "Uniqueid",
                    "managed": false
                }
            ],
            "source": {
                "type": "BR",
                "name": "Zaloni Bedrock"
            }
        },
        {
            "complexType": false,
            "nonTopLevel": true
        },
        {
            "fieldId": -1,
            "position": 5,
            "fieldTechnicalName": "/profile/phone",
            "dataTypeId": 4,
            "fieldDataType": "STRING",
            "startOffset": null,
            "endOffset": null,
            "fieldDataLength": "",
            "dataScale": 0,
            "dataFormat": "",
            "fieldBusinessName": "phone",
            "fieldDataTypeSchema": "{\"type\":\"string\",\"id\":4,\"
        ↪ "name\":\"phone\"}",
            "isComplexField": false,
            "primary": false,
            "searchable": false,
            "masked": false,
            "maskingPattern": "",
            "tokenized": false,
            "tokenizerAlgorithmId": 0,
            "managedListId": 0,
            "managedListName": "",
            "tokenizerAlgorithmName": null,
            "sensitivityId": 0,
            "sensitivityValue": null,
            "ruleMappings": [
                {
                    "ruleName": "avro_area",
                    "ruleSetName": null,
                    "ruleSetId": null,
                    "ruleId": 7,
                    "executionOrder": 1,
                    "stopExecution": false
                }
            ]
        }
    ],
    "managed": false
}

```

```

        "variableMapping": [
            {
                "variableName": "primaryVar",
                "fieldId": 11675,
                "fieldTechnicalName": "/Add/STATE",
                "varType": null
            }
        ],
        "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"rules\\":[{\\"
↪ "data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\":true}}"
            }
        ],
        "cdcColumn": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "privRating
↪ ",
                        "value": "0",
                        "managed": false
                    },
                    {
                        "id": "alias",
                        "value": "Uniqueid
↪ ",
                        "managed": false
                    }
                ],
                "source": {
                    "type": "BR",
                    "name": "Zaloni Bedrock"
                }
            }
        ],
        "complexType": false
    },
    {
        "position": 6,
        "endOffset": 0,
        "startOffset": 0,
        "dataFormat": "",
        "primary": false,
        "fieldId": 17,
        "fieldBusinessName": "profile",
        "fieldTechnicalName": "profile",
        "fieldDataType": "RECORD",
        "dataTypeId": 22,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "nonTopLevel": false,
        "fieldDataTypeSchema": "{\\"type\\":\\"record\\",\\"id\\":22,\\"
↪ "fields\\":[{\\"type\\":\\"string\\",\\"id\\":4,\\"name\\":\\"email\\"},{\\"type\\":\\"string\\",
↪ "\",\\"id\\":4,\\"name\\":\\"phone\\"}],\\"name\\":\\"profile\\"}",
        "fieldDataLength": "",
        "maskingPattern": "",
        "tokenizerAlgorithmName": null,
        "managedListName": null,
    }

```



```

        "searchable": false,
        "managedListId": 0,
        "dataScale": 0,
        "masked": false,
        "tokenized": false,
        "tokenizerAlgorithmId": 0,
        "ruleMappings": [
            {
                "ruleName": "avro_area",
                "ruleSetName": null,
                "ruleSetId": null,
                "ruleId": 7,
                "executionOrder": 1,
                "stopExecution": false
            }
        ],
        "variableMapping": [
            {
                "variableName": "primaryVar",
                "fieldId": 11675,
                "fieldTechnicalName": "/Add/STATE",
                "varType": null
            }
        ],
        "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"},\\"firstGroup\\":true]}}"
        ↪,
        "cdcColumn": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        ↪,
                        "id": "privRating",
                        "value": "0",
                        "managed": false
                    },
                    {
                        "id": "alias",
                        "value": "Uniqueid",
                        ↪,
                        "managed": false
                    }
                ],
                "source": {
                    "type": "BR",
                    "name": "Zaloni Bedrock"
                }
            }
        ],
        "complexType": true,
    },
    "partitionFields": [],
    "validateRequest": false,
    "businessName": "Clickable",
    "technicalName": "clicking",
    "description": "",
    "subjectArea": "",

```

```

    "label": "",
    "dataStoreId": 2,
    "dataFileFormatId": 10,
    "delimiter": "",
    "entityCreateOptions": 1,
    "complexEntityVersion": null,
    "recordLength": null,
    "frequencyOfUpdate": "",
    "sourcePlatform": "cliq",
    "sourceSchema": "cliqdb",
    "loadType": "",
    "hasInactiveTable": false,
    "ruleSetId": 0,
    "isComplexEntity": true,
    "serdeInfoProperties": {
        "": ""
    },
    "libClasspath": "",
    "serdeLib": "",
    "inputFormatLib": "",
    "outputFormatLib": "",
    "targetSchema": "test3",
    "external": false,
    "externalDataPath": null,
    "fileFormat": "TextFile",
    "tableProperties": {
        "skip.header.line.count": "0",
        "myKey": "myValue"
    },
    "derived": false,
    "position": 4,
    "entityClusterLocation": [
        {
            "clusterId": "192_168_1_39"
        }
    ],
    "tableType": "EXTERNAL",
    "cdcActionable": true
}

```

Example Response:

If the entity type is updated successfully:

```

{
    "responseMessage": "Entity type updated successfully with entity id: 112 and ↵
    ↵version: 1",
    "restUri": null,
    "result": {
        "entityTypeId": 112,
        "version": 1,
        "ruleSetId": 0,
        "sharedWith": 0,
        "shared": false,
        "ownerProjectId": 1,
        "ownerProjectName": "Public Project",
        "ruleSetName": null,
    }
}

```

```

    "businessName": "emp62",
    "technicalName": "emp6",
    "description": null,
    "subjectArea": null,
    "label": null,
    "dataStoreId": 2,
    "dataStoreName": "HDFS",
    "frequencyOfUpdate": null,
    "sourcePlatform": "HIVE",
    "sourceSchema": "inventory",
    "loadType": null,
    "preQueryStatements": null,
    "dataFileFormatId": 1,
    "dataFileFormat": "DELIMITED",
    "metadata": "{ \"hasComplexFields\": false, \"schemaUpload\": false, \"
↪ \"uiFields\": [ { \"name\": \"delimiter\", \"required\": false }, { \"name\": \"
↪ \"lineDelimiter\", \"required\": false } ] }",
    "recordLength": null,
    "createdBy": "admin",
    "createdTime": "07/25/2017 18:38:53",
    "createHiveTable": true,
    "customTable": true,
    "modifiedBy": "admin",
    "modifiedTime": "07/25/2017 19:20:45",
    "ingestionTime": null,
    "newVersionRequired": false,
    "syncedWithHive": false,
    "owner": "do2user1",
    "securityContext": {
        "loggedInUser": "admin",
        "doAsUser": "do2user1",
        "projectIds": [
            1
        ],
        "artifactPermissions": null,
        "impersonationOn": true,
        "powerUser": true,
        "validImpersonationRequest": true
    },
    "isDelete": null,
    "blockDerivedEntityUpdate": true,
    "derived": false,
    "fields": [
        {
            "fieldId": 42,
            "position": 1,
            "fieldTechnicalName": "id",
            "dataTypeId": 10,
            "fieldDataType": "INT",
            "fieldDataTypeSchema": "{ \"type\": \"INT\" }",
            "fieldDataLength": "",
            "dataScale": 0,
            "dataFormat": null,
            "fieldBusinessName": "id",
            "description": null,
            "primary": false,
            "searchable": false,
            "masked": false,

```

```

        "tokenized": false,
        "nonTopLevel": false,
        "managedListId": 0,
        "managedListName": null,
        "tokenizerAlgorithmId": 0,
        "tokenizerAlgorithmName": null,
        "algoArguments": null,
        "maskingPattern": null,
        "startOffset": 0,
        "endOffset": 0,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleName": "avro_area",
                "ruleSetName": null,
                "ruleSetId": null,
                "ruleId": 7,
                "executionOrder": 1,
                "stopExecution": false
            }
        ],
        "variableMapping": [
            {
                "variableName": "primaryVar",
                "fieldId": 11675,
                "fieldTechnicalName": "/Add/STATE",
                "varType": null
            }
        ],
        "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"rules\\":{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}},\\"firstGroup\\":true}}",
        ],
        "cdcColumn": false,
        "customAttributes": [],
        "complexType": false
    },
    {
        "fieldId": 40,
        "position": 2,
        "fieldTechnicalName": "name",
        "dataTypeId": 7,
        "fieldDataType": "STRING",
        "fieldDataTypeSchema": "{\\"type\\":\\"STRING\\"}",
        "fieldDataLength": "",
        "dataScale": 0,
        "dataFormat": null,
        "fieldBusinessName": "name",
        "description": null,
        "primary": false,
        "searchable": false,
        "masked": false,
        "tokenized": false,
        "nonTopLevel": false,
        "managedListId": 0,
        "managedListName": null,
        "tokenizerAlgorithmId": 0,
        "tokenizerAlgorithmName": null,
        "algoArguments": null,

```

```

        "maskingPattern": null,
        "startOffset": 0,
        "endOffset": 0,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleName": "avro_area",
                "ruleSetName": null,
                "ruleSetId": null,
                "ruleId": 7,
                "executionOrder": 1,
                "stopExecution": false
            }
        ],
        "variableMapping": [
            {
                "variableName": "primaryVar",
                "fieldId": 11675,
                "fieldTechnicalName": "/Add/STATE",
                "varType": null
            }
        ],
        "expressionPayload": "{\n\"group\":{\n\"operator\":\n\"AND\", \n\"rules\"↪\": [{\n\"data\":\n\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\"}], \n\"firstGroup\":true}}",
        "cdcColumn": false,
        "customAttributes": [],
        "complexType": false
    },
    {
        "fieldId": 41,
        "position": 3,
        "fieldTechnicalName": "age",
        "dataTypeId": 22,
        "fieldDataType": "DECIMAL",
        "fieldDataTypeSchema": "{\n\"type\":\n\"DECIMAL\"",
        "fieldDataLength": "30",
        "dataScale": 20,
        "dataFormat": null,
        "fieldBusinessName": "age",
        "description": null,
        "primary": false,
        "searchable": false,
        "masked": false,
        "tokenized": false,
        "nonTopLevel": false,
        "managedListId": 0,
        "managedListName": null,
        "tokenizerAlgorithmId": 0,
        "tokenizerAlgorithmName": null,
        "algoArguments": null,
        "maskingPattern": null,
        "startOffset": 0,
        "endOffset": 0,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "ruleMappings": [
            {

```

```

        "ruleName": "avro_area",
        "ruleSetName": null,
        "ruleSetId": null,
        "ruleId": 7,
        "executionOrder": 1,
        "stopExecution": false

    "variableMapping": [
        {
            "variableName": "primaryVar",
            "fieldId": 11675,
            "fieldTechnicalName": "/Add/STATE",
            "varType": null
        }
    ],
    "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\":true}}",
    },
    "cdcColumn": false,
    "customAttributes": [],
    "complexType": false
},
"partitionFields": [],
"ruleSetEntityMapping": [],
"complexMetadataEntityMapping": [],
"complexEntityVersion": null,
"hcatTableId": 11,
"tableName": "emp6",
"targetSchema": "inventory",
"targetSchemaValue": "inventory",
"delimiter": ",",
"lineDelimiter": null,
"external": false,
"externalDataPath": null,
"externalDataPathValue": null,
"fileFormat": null,
"serdeLib": "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
"libClasspath": null,
"inputFormatLib": "org.apache.hadoop.mapred.TextInputFormat",
"outputFormatLib": "org.apache.hadoop.hive.ql.io.
↪HiveIgnoreKeyTextOutputFormat",
"tableProperties": {
    "totalSize": "0",
    "numRows": "0",
    "rawDataSize": "0",
    "COLUMN_STATS_ACCURATE": "{\\"BASIC_STATS\\":\\"true\\"}",
    "numFiles": "0",
    "transient_lastDdlTime": "1500988125"
},
"storageDescriptorProperties": {},
"serdeInfoProperties": {
    "serialization.format": "1"
},
"cdcActionable": false,
"hasInActiveTable": false,
"entityClusterLocation": [
    {

```

```
        "clusterId": "192_168_1_39"
      },
      ],
      "changeDetected": false,
      "validateRequest": false,
      "latestVersion": true,
      "migrate": false,
      "tableType": "INTERNAL",
      "skewing": null,
      "bucketing": null,
      "view": null,
      "retainExistingAdditionalAttributes": false,
      "import": false,
      "syncJob": false
    },
    "page": null
  }
}
```

Response in Case of Failure:

- Scenario 1:

If the entity type Id or version do not exist:

```
{
  "responseMessage": "Artifact with Id 448 and type ENTITY cannot be
↪accessed from project with Id 1",
  "restUri": "/bedrock-app/services/rest/entities/44448",
  "result": null,
  "page": null
}
```

- Scenario 2:

If the field Id does not exist:

```
{
  "responseMessage": "Field id :<field id> is not valid",
  "restUri": "/bedrock-app/services/rest/projects/<project_id>/entities",
  "result": null,
  "page": null
}
```

- Scenario 3:

If the version of the complex entity type is not defined:

```
{
  "responseMessage": "An Error Occurred While Serving Request. Check Logs
↪For More Details",
  "restUri": "/bedrock-app/services/rest/entities/116",
  "result": null,
  "page": null
}
```

1.6.2.3 Create entity from a Hive table

This API allows you to create an entity in ZDP by using an already existing table in Hive.

- The service uses the Hive table definitions to create the entity in ZDP.
- By using this service, a table in Hive is exposed in ZDP as an entity and can be part of various operations.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/projects/{project_Id}/  
↪inventory/entities
```

Note: You must pass the *project_Id* in the URL.

Method:

POST

Request Payload:

```
{  
  "targetSchema": "<This is the Hive database name (or the 'Target Schema'),  
↪under which the Hive table exists.>",  
  "tableName": "<This is the Hive Table name whose definitions will be used to  
↪create the entity in ZDP.>",  
  "owner": "<This is the owner of the Hive table. The same user also owns the  
↪entity that is created in ZDP by this service.>",  
  "clusterId": "<This is the cluster Id that points to the corresponding Hive  
↪Metastore (where Hive table can be accessed).>"  
}
```

Example Request:

```
{  
  *"targetSchema": "chivas",  
  *"tableName": "100records",  
  "owner": "do2user1",  
  *"clusterId": "1"  
}
```

Example Response:

If the entity is successfully registered in ZDP:

```
{  
  "responseMessage": null,  
  "restUri": null,  
  "result": {  
    "entityTypeId": 255,  
    "version": 1,  
    "ruleSetId": 0,  
    "sharedWith": null,  
    "shared": null,  
    "ownerProjectId": null,  
    "ownerProjectName": null,  
    "ruleSetName": null,  
    "businessName": "100records",  
    "technicalName": "100records",  
    "description": null,  
    "subjectArea": null,  
    "label": null,  
  }  
}
```



```

"dataStoreId": 2,
"dataStoreName": "HDFS",
"frequencyOfUpdate": null,
"sourcePlatform": "HIVE",
"sourceSchema": "chivas",
"loadType": null,
"preQueryStatements": null,
"dataFileFormatId": 1,
"dataFileFormat": "DELIMITED",
"metadata": null,
"recordLength": null,
"createdBy": "admin",
"createTime": "06/09/2017 16:14:55",
"createHiveTable": false,
"customTable": true,
"modifiedBy": "admin",
"modifiedTime": "06/09/2017 16:14:55",
"ingestionTime": null,
"newVersionRequired": false,
"syncedWithHive": true,
"owner": "do2user1",
"securityContext": {
  "loggedInUser": "admin",
  "doAsUser": "do2user1",
  "projectIds": [
    1
  ],
  "artifactPermissions": null,
  "impersonationOn": true,
  "powerUser": true,
  "validImpersonationRequest": true
},
"isDelete": null,
"blockDerivedEntityUpdate": false,
"derived": false,
"fields": [
  {
    "fieldId": 864,
    "position": 1,
    "fieldTechnicalName": "id",
    "dataTypeId": 10,
    "fieldDataType": "INT",
    "fieldDataTypeSchema": "{\"type\":\"INT\"}",
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": null,
    "fieldBusinessName": "id",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": null,
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,
    "algoArguments": null,

```

```

        "maskingPattern": null,
        "startOffset": 0,
        "endOffset": 0,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleSetId": null,
                "ruleId": 8,
                "ruleName": "SimpleRule_API",
                "ruleSetName": null,
                "ruleTag": "",
                "ruleDataType": "STRING",
                "ruleSetDataType": null,
                "ruleSetTag": null,
                "stopExecution": false,
                "executionOrder": 2,
                "variableMapping": [
                    {
                        "variableName": "primaryVar",
                        "fieldId": 11675,
                        "fieldTechnicalName": "/Add/STATE",
                        "varType": null
                    }
                ],
                "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\":true}}",
            }
        ],
        "cdcColumn": false,
        "customAttributes": null,
        "complexType": true
    },
    {
        "fieldId": 865,
        "position": 2,
        "fieldTechnicalName": "first_name",
        "dataTypeId": 7,
        "fieldDataType": "STRING",
        "fieldDataTypeSchema": "{\\"type\\":\\"STRING\\"}",
        "fieldDataLength": "",
        "dataScale": 0,
        "dataFormat": null,
        "fieldBusinessName": "first_name",
        "description": null,
        "primary": false,
        "searchable": false,
        "masked": false,
        "tokenized": false,
        "nonTopLevel": false,
        "managedListId": 0,
        "managedListName": null,
        "tokenizerAlgorithmId": 0,
        "tokenizerAlgorithmName": null,
        "algoArguments": null,
        "maskingPattern": null,
        "startOffset": 0,
    }

```

```

        "endOffset": 0,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleSetId": null,
                "ruleId": 8,
                "ruleName": "SimpleRule_API",
                "ruleSetName": null,
                "ruleTag": "",
                "ruleDataType": "STRING",
                "ruleSetDataType": null,
                "ruleSetTag": null,
                "stopExecution": false,
                "executionOrder": 2,
                "variableMapping": [
                    {
                        "variableName": "primaryVar",
                        "fieldId": 11675,
                        "fieldTechnicalName": "/Add/STATE",
                        "varType": null
                    }
                ],
                "expressionPayload": "{\"group\":{\"operator\":\"AND\\
↪\", \"rules\": [{\"data\": \"EQUALS_STRING(primaryVar, \\\"ZPL\\\")\"}], \"firstGroup\\
↪\": true}}\"
            }
        ],
        "cdcColumn": false,
        "customAttributes": null,
        "complexType": true
    },
    {
        "fieldId": 866,
        "position": 3,
        "fieldTechnicalName": "last_name",
        "dataTypeId": 7,
        "fieldDataType": "STRING",
        "fieldDataTypeSchema": "{\"type\":\"STRING\"}",
        "fieldDataLength": "",
        "dataScale": 0,
        "dataFormat": null,
        "fieldBusinessName": "last_name",
        "description": null,
        "primary": false,
        "searchable": false,
        "masked": false,
        "tokenized": false,
        "nonTopLevel": false,
        "managedListId": 0,
        "managedListName": null,
        "tokenizerAlgorithmId": 0,
        "tokenizerAlgorithmName": null,
        "algoArguments": null,
        "maskingPattern": null,
        "startOffset": 0,
        "endOffset": 0,
        "sensitivityId": 0,

```

```

        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleSetId": null,
                "ruleId": 8,
                "ruleName": "SimpleRule_API",
                "ruleSetName": null,
                "ruleTag": "",
                "ruleDataType": "STRING",
                "ruleSetDataType": null,
                "ruleSetTag": null,
                "stopExecution": false,
                "executionOrder": 2,
                "variableMapping": [
                    {
                        "variableName": "primaryVar",
                        "fieldId": 11675,
                        "fieldTechnicalName": "/Add/STATE",
                        "varType": null
                    }
                ],
                "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\\\"}\\")\\}],\\"firstGroup\\":true}}"}
            },
            {
                "cdcColumn": false,
                "customAttributes": null,
                "complexType": true
            }
        ],
        "partitionFields": [],
        "ruleSetEntityMapping": null,
        "complexMetadataEntityMapping": [],
        "complexEntityVersion": null,
        "hcatTableId": 93,
        "tableName": "100records",
        "targetSchema": "chivas",
        "targetSchemaValue": "chivas",
        "delimiter": ",",
        "lineDelimiter": null,
        "external": false,
        "externalDataPath": null,
        "externalDataPathValue": null,
        "fileFormat": null,
        "serdeLib": "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
        "libClasspath": null,
        "inputFormatLib": "org.apache.hadoop.mapred.TextInputFormat",
        "outputFormatLib": "org.apache.hadoop.hive.ql.io.
        ↪HiveIgnoreKeyTextOutputFormat",
        "tableProperties": {
            "transient_lastDdlTime": "1496997948",
            "totalSize": "1901",
            "numFiles": "1"
        },
        "storageDescriptorProperties": {},
        "serdeInfoProperties": {
            "serialization.format": ",",

```

```

        "field.delim": ",",
    },
    "cdcActionable": false,
    "hasInActiveTable": false,
    "validateRequest": false,
    "latestVersion": true,
    "migrate": false,
    "tableType": "TABLE",
    "skewing": {
        "skewingId": 0,
        "storedAsDirectories": false,
        "skewingFields": []
    },
    "bucketing": null,
    "view": null,
    "import": false,
    "syncJob": false
},
"page": null
}

```

Response in case of Failure: - Scenario 1:

If the user (or 'owner') does not have enough permissions:

```

{
  "responseMessage": "User doesn't have permissions to register entity with_
↪database=chivi and table=1002567",
  "restUri": "/bedrock-app/services/rest/metadata/projects/1/inventory/
↪entities",
  "result": null,
  "page": null
}

```

- Scenario 2:

If the combination of the 'targetSchema' and 'tableName' is invalid:

```

{
  "responseMessage": "No artifact with target schema=zigmo and table_
↪name=raga, present.",
  "restUri": "/bedrock-app/services/rest/metadata/projects/1/inventory/
↪entities",
  "result": null,
  "page": null
}

```

1.6.3 Sync Entity with the Hive table

This API allows you to auto-update (or sync) the entity in ZDP with the corresponding table in *Hive*.

- The service can prove to be useful when the Hive table (or table definitions) corresponding to the entity (in ZDP) has been updated and thus requiring an sync across the both. For example, when columns are added or the Table Properties are modified in the Hive table.
- In case there is no change or update in the Hive table, the service ignores the same.
- Any user having the *Manage Entity Types* permission in ZDP can use this service.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/synchronization/  
→entities/{entity_Id}/versions/{entity_version}?projectId={project_Id}&clusterId=  
→{cluster_Id}
```

Note: You must pass the *entity_Id*, *entity_version*, *project_Id* and the *cluster_Id* in the request URL.

Method:

PUT

Example Response:

If the service request is successful:

- The flag - "*updateNotRequired*" : *false* (in the response) indicates the Hive table had certain updates and the corresponding Entity was synchronized.
- "*updateNotRequired*" : *true* (in the response) indicates that the corresponding Hive table has no update.

```
{  
  "responseMessage": null,  
  "restUri": null,  
  "result": {  
    "entityTypeId": 59,  
    "version": 1,  
    "ruleSetId": 0,  
    "sharedWith": null,  
    "shared": null,  
    "ownerProjectId": null,  
    "ownerProjectName": null,  
    "ruleSetName": null,  
    "businessName": "100rec",  
    "technicalName": "100rec",  
    "description": null,  
    "subjectArea": null,  
    "label": null,  
    "dataStoreId": 2,  
    "dataStoreName": "HDFS",  
    "frequencyOfUpdate": null,  
    "sourcePlatform": "HIVE",  
    "sourceSchema": "chivi",  
    "loadType": null,  
    "preQueryStatements": null,  
    "dataFileFormatId": 1,  
    "dataFileFormat": "DELIMITED",  
    "metadata": "{ \"hasComplexFields\": false, \"schemaUpload\": false, \  
→\"uiFields\": [ { \"name\": \"delimiter\", \"required\": false }, { \"name\": \  
→\"lineDelimiter\", \"required\": false } ] }",  
    "recordLength": null,  
    "createdBy": "admin",  
    "createdTime": "06/14/2017 13:46:21",  
    "createHiveTable": true,  
    "customTable": true,  
    "modifiedBy": "admin",  
    "modifiedTime": "06/14/2017 13:47:32",  
    "ingestionTime": null,  
  },  
}
```

```

"newVersionRequired": false,
"syncedWithHive": true,
"owner": "do2user1",
"securityContext": {
  "loggedInUser": "admin",
  "doAsUser": "do2user1",
  "projectIds": [
    1
  ],
  "artifactPermissions": null,
  "impersonationOn": true,
  "powerUser": true,
  "validImpersonationRequest": true
},
"isDelete": null,
"blockDerivedEntityUpdate": false,
"derived": false,
"fields": [
  {
    "fieldId": 392,
    "position": 1,
    "fieldTechnicalName": "id",
    "dataTypeId": 10,
    "fieldDataType": "INT",
    "fieldDataTypeSchema": "{\"type\":\"INT\"}",
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": null,
    "fieldBusinessName": "id",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": null,
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,
    "algoArguments": null,
    "maskingPattern": null,
    "startOffset": 0,
    "endOffset": 0,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "ruleMappings": [
      {
        "ruleSetId": null,
        "ruleId": 8,
        "ruleName": "SimpleRule_API",
        "ruleSetName": null,
        "ruleTag": "",
        "ruleDataType": "STRING",
        "ruleSetDataType": null,
        "ruleSetTag": null,
        "stopExecution": false,
        "executionOrder": 2,
        "variableMapping": [

```

```

        {
            "variableName": "primaryVar",
            "fieldId": 11675,
            "fieldTechnicalName": "/Add/STATE",
            "varType": null
        },
        {
            "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\\\"}\\\"}],\\"firstGroup\\":true}}",
            "cdcColumn": false,
            "customAttributes": null,
            "complexType": true
        },
        {
            "fieldId": 391,
            "position": 2,
            "fieldTechnicalName": "first_name",
            "dataTypeId": 7,
            "fieldDataType": "STRING",
            "fieldDataTypeSchema": "{\\"type\\":\\"STRING\\"}",
            "fieldDataLength": "",
            "dataScale": 0,
            "dataFormat": null,
            "fieldBusinessName": "first_name",
            "description": null,
            "primary": false,
            "searchable": false,
            "masked": false,
            "tokenized": false,
            "nonTopLevel": false,
            "managedListId": 0,
            "managedListName": null,
            "tokenizerAlgorithmId": 0,
            "tokenizerAlgorithmName": null,
            "algoArguments": null,
            "maskingPattern": null,
            "startOffset": 0,
            "endOffset": 0,
            "sensitivityId": 0,
            "sensitivityValue": null,
            "ruleMappings": [
                {
                    "ruleSetId": null,
                    "ruleId": 8,
                    "ruleName": "SimpleRule_API",
                    "ruleSetName": null,
                    "ruleTag": "",
                    "ruleDataType": "STRING",
                    "ruleSetDataType": null,
                    "ruleSetTag": null,
                    "stopExecution": false,
                    "executionOrder": 2,
                    "variableMapping": [
                        {
                            "variableName": "primaryVar",

```



```

        "fieldId": 11675,
        "fieldTechnicalName": "/Add/STATE",
        "varType": null
    },
    ],
    "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\\\"}\\",\\"firstGroup\\":true}]}"
},
    ],
    "cdcColumn": false,
    "customAttributes": null,
    "complexType": true
},
{
    "fieldId": 390,
    "position": 3,
    "fieldTechnicalName": "last_name",
    "dataTypeId": 7,
    "fieldDataType": "STRING",
    "fieldDataTypeSchema": "{\\"type\\":\\"STRING\\"}",
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": null,
    "fieldBusinessName": "last_name",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": null,
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,
    "algoArguments": null,
    "maskingPattern": null,
    "startOffset": 0,
    "endOffset": 0,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "ruleMappings": [
        {
            "ruleSetId": null,
            "ruleId": 8,
            "ruleName": "SimpleRule_API",
            "ruleSetName": null,
            "ruleTag": "",
            "ruleDataType": "STRING",
            "ruleSetDataType": null,
            "ruleSetTag": null,
            "stopExecution": false,
            "executionOrder": 2,
            "variableMapping": [
                {
                    "variableName": "primaryVar",
                    "fieldId": 11675,
                    "fieldTechnicalName": "/Add/STATE",

```

```

        "varType": null
      },
      ],
      "expressionPayload": "{\"group\":{\"operator\":\"AND\\
↪\", \"rules\": [{\"data\": \"EQUALS_STRING(primaryVar, \\\"ZPL\\\")\"}], \"firstGroup\\
↪\": true}}\"
    }
  ],
  "cdcColumn": false,
  "customAttributes": null,
  "complexType": true
}
],
"partitionFields": [],
"ruleSetEntityMapping": [],
"complexMetadataEntityMapping": [],
"complexEntityVersion": null,
"hcatTableId": 48,
"tableName": "100records",
"targetSchema": "chivas",
"targetSchemaValue": "chivas",
"delimiter": ",",
"lineDelimiter": null,
"external": false,
"externalDataPath": null,
"externalDataPathValue": null,
"fileFormat": null,
"serdeLib": "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
"libClasspath": null,
"inputFormatLib": "org.apache.hadoop.mapred.TextInputFormat",
"outputFormatLib": "org.apache.hadoop.hive ql.io.
↪HiveIgnoreKeyTextOutputFormat",
"tableProperties": {
  "transient_lastDdlTime": "1496997948",
  "totalSize": "1901",
  "serialization.format": ",",
  "field.delim": ",",
  "numFiles": "1"
},
"storageDescriptorProperties": {},
"serdeInfoProperties": {
  "serialization.format": ",",
  "field.delim": ","
},
"cdcActionable": false,
"hasInActiveTable": false,
"validateRequest": false,
"latestVersion": true,
"migrate": false,
"tableType": "TABLE",
"skewing": {
  "skewingId": 0,
  "storedAsDirectories": false,
  "skewingFields": []
},
"bucketing": {
  "bucketingId": 0,
  "numberOfBuckets": 0,

```

```

        "bucketingFields": null,
        "sortingFields": null
    },
    "view": {
        "query": null,
        "associatedEntities": null
    },
    "import": false,
    "syncJob": false
},
"page": null
}

```

Response in case of Failure

- Scenario 1:

If the *clusterId* parameter passed in the *URL* does not exist (or not available):

```

{
    "responseMessage": "There is no cluster with clusterId=4",
    "restUri": "/bedrock-app/services/rest/metadata/synchronization/entities/
↪131/versions/1",
    "result": null,
    "page": null
}

```

- Scenario 2:

If the *projectId* (passed as *URL* parameter) does not exist:

```

{
    "responseMessage": "Project with Id 9 doesn't exist",
    "restUri": "/bedrock-app/services/rest/metadata/synchronization/entities/
↪136/versions/1",
    "result": null,
    "page": null
}

```

1.6.4 Fetch Schema Parser Attributes

This API allows you to fetch the schema of the parser attributes. The parser attributes can be *JSON*, *DELIMITED*, and *AVRO*.

URL:

```

http://<bedrock-host>:port/bedrock-app/services/rest/entities/parsers?parserType=
↪{parser_type}

```

Note: You must pass the *parser_type* parameter in the API URL. The sample values are *DELIMITED*, *JSON*, and *AVRO*.

Method:

```
GET
```

Example Response:

- Scenario 1:

If the *parser_type* is *DELIMITED*:

```
{
  "responseMessage": null,
  "restUri": null,
  "result": [
    {
      "attributeId": 1,
      "attributeKey": "delimiter",
      "label": "Delimiter",
      "type": "TEXT",
      "defaultValue": ",",
      "dataFileFormat": "DELIMITED"
    },
    {
      "attributeId": 2,
      "attributeKey": "quoteCharacter",
      "label": "Quote Character",
      "type": "TEXT",
      "defaultValue": "\"",
      "dataFileFormat": "DELIMITED"
    },
    {
      "attributeId": 3,
      "attributeKey": "firstRowContainsHeader",
      "label": "First Row Contains Header",
      "type": "BOOLEAN",
      "defaultValue": "true",
      "dataFileFormat": "DELIMITED"
    }
  ],
  "page": null
}
```

- Scenario 2:

If the *parser_type* is *AVRO*:

```
{
  "responseMessage": null,
  "restUri": null,
  "result": [
    {
      "attributeId": 4,
      "attributeKey": "avroFileType",
      "label": "File Type",
      "type": "RADIO",
      "defaultValue": "Data, Schema",
      "dataFileFormat": "AVRO"
    }
  ],
  "page": null
}
```

- Scenario 3:

If the *parser_type* is *JSON*:

```
{
  "responseMessage": null,
  "restUri": null,
  "result": [],
  "page": null
}
```

Response in Case of Failure:

If the *parser_type* is not present.

```
{
  "responseMessage": "Required String parameter 'parseType' is not present",
  "restUri": "/bedrock-app/services/rest/entities/parsers",
  "result": null,
  "page": null
}
```

1.6.5 Upload Data File for Schema Discovery

This API allows you to upload data file for schema discovery. This API is useful if you do not want to manually add multiple fields and define their type. By using this API, you can upload a single file with multiple fields.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/entities/schemas?
  ↪parserAttributes={parser_attribute}&parserType={parser_type}
```

Note: You must pass the *parser_type* and *parser_attribute* parameters in the API URL. The sample values for *parser_type* are *DELIMITED*, *JSON*, and *AVRO*.

- If the *parser_type* is *JSON*, the *parser_attribute* is:

```
{}
```

- If the *parser_type* is *AVRO* and the *file_type* is *data*, the *parser_attribute* is:

```
{"File Type": "Data"}
```

- If the *parser_type* is *AVRO* and the *file_type* is *schema*, the *parser_attribute* is:

```
{"File Type": "Schema"}
```

- If the *parser_type* is *DELIMITED*, the *parser_attribute* is:

```
{
  "Delimiter": ",",
  "Quote Character": "!",
  "First Row Contains Header": "true"
}
```

Method:

POST

Parameter Description:

The parameter type must be of form-data with the key-values as below:

Key	Value
dataFile	Upload the data file from your system. It can be .avro,.avsc,.csv, etc files.

Example Response:

- Scenario 1:

If the *parser_type* is *JSON*:

```
{
  "responseMessage": null,
  "restUri": null,
  "result": {
    "entityTypeId": 0,
    "version": 0,
    "ruleSetId": 0,
    "sharedWith": null,
    "shared": null,
    "ownerProjectId": null,
    "ownerProjectName": null,
    "ruleSetName": null,
    "businessName": "Example",
    "technicalName": "Example",
    "description": "Entity generated from file [Example.json]",
    "subjectArea": null,
    "label": null,
    "dataStoreId": 2,
    "dataStoreName": "HDFS",
    "frequencyOfUpdate": null,
    "sourcePlatform": "undefined",
    "sourceSchema": "undefined",
    "loadType": null,
    "preQueryStatements": null,
    "dataFileFormatId": 0,
    "dataFileFormat": "JSON",
    "metadata": null,
    "recordLength": null,
    "createdBy": null,
    "createdTime": null,
    "customTable": false,
    "modifiedBy": null,
    "modifiedTime": null,
    "ingestionTime": null,
    "newVersionRequired": false,
    "securityContext": null,
    "isDelete": null,
    "blockDerivedEntityUpdate": false,
    "derived": false,
    "fields": [
      {
        "fieldId": 0,
        "position": 1,
```

```

        "fieldTechnicalName": "_corrupt_record",
        "dataTypeId": 0,
        "fieldDataType": "STRING",
        "fieldDataTypeSchema": "{\"name\":\"_corrupt_
→record\\",\"type\":\"string\"}",
        "fieldDataLength": null,
        "dataScale": 0,
        "dataFormat": null,
        "fieldBusinessName": "_corrupt_record",
        "description": null,
        "primary": false,
        "searchable": false,
        "masked": false,
        "tokenized": false,
        "nonTopLevel": false,
        "managedListId": 0,
        "managedListName": null,
        "tokenizerAlgorithmId": 0,
        "tokenizerAlgorithmName": null,
        "algoArguments": null,
        "maskingPattern": null,
        "startOffset": 0,
        "endOffset": 0,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleSetId": null,
                "ruleId": 8,
                "ruleName": "SimpleRule_API",
                "ruleSetName": null,
                "ruleTag": "",
                "ruleDataType": "STRING",
                "ruleSetDataType": null,
                "ruleSetTag": null,
                "stopExecution": false,
                "executionOrder": 2,
                "variableMapping": [
                    {
                        "variableName": "primaryVar",
                        "fieldId": 11675,
                        "fieldTechnicalName": "/Add/STATE",
                        "varType": null
                    }
                ],
                "expressionPayload": "{\"group\":{\"operator\":\"AND\\",\
→\"rules\":[{\"data\":\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\"}],\"firstGroup\
→\":true}}"
            }
        ],
        "cdcColumn": false,
        "customAttributes": null,
        "complexType": false
    },
    "partitionFields": [],
    "ruleSetEntityMapping": null,
    "complexMetadataEntityMapping": [],

```

```

        "complexEntityVersion": null,
        "hcatTableId": 0,
        "tableName": null,
        "targetSchema": "auto",
        "targetSchemaValue": "auto",
        "delimiter": null,
        "lineDelimiter": null,
        "external": false,
        "externalDataPath": null,
        "externalDataPathValue": null,
        "fileFormat": "JSON",
        "serdeLib": null,
        "libClasspath": null,
        "inputFormatLib": null,
        "outputFormatLib": null,
        "tableProperties":
            {
                "skip.header.line.count": "0",
                "myKey": "myValue"
            },
        "storageDescriptorProperties": {},
        "serdeInfoProperties": {},
        "cdcActionable": false,
        "hasInActiveTable": false,
        "validateRequest": false,
        "latestVersion": false,
        "migrate": false,
        "syncJob": false
    },
    "page": null
}

```

- Scenario 2:

If the *parser_type* is *DELIMITED*:

```

{
    "responseMessage": null,
    "restUri": null,
    "result": {
        "entityTypeId": 0,
        "version": 0,
        "ruleSetId": 0,
        "sharedWith": null,
        "shared": null,
        "ownerProjectId": null,
        "ownerProjectName": null,
        "ruleSetName": null,
        "businessName": "Entity_template",
        "technicalName": "Entity_template",
        "description": "Entity generated from file [Entity_template.csv]",
        "subjectArea": null,
        "label": null,
        "dataStoreId": 2,
        "dataStoreName": "HDFS",
        "frequencyOfUpdate": null,
        "sourcePlatform": "undefined",
        "sourceSchema": "undefined",
    }
}

```



```

"loadType": null,
"preQueryStatements": null,
"dataFileFormatId": 0,
"dataFileFormat": "DELIMITED",
"metadata": null,
"recordLength": null,
"createdBy": null,
"createdTime": null,
"customTable": false,
"modifiedBy": null,
"modifiedTime": null,
"ingestionTime": null,
"newVersionRequired": false,
"securityContext": null,
"isDelete": null,
"blockDerivedEntityUpdate": false,
"derived": false,
"fields": [
  {
    "fieldId": 0,
    "position": 2,
    "fieldTechnicalName": "Data_Type",
    "dataTypeId": 0,
    "fieldDataType": "STRING",
    "fieldDataTypeSchema": "{ \"name\": \"Data_Type\", \"type\": \"STRING\"
→ }",
    "fieldDataLength": null,
    "dataScale": 0,
    "dataFormat": null,
    "fieldBusinessName": "Data_Type",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": null,
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,
    "algoArguments": null,
    "maskingPattern": null,
    "startOffset": 0,
    "endOffset": 0,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "ruleMappings": [
      {
        "ruleSetId": null,
        "ruleId": 8,
        "ruleName": "SimpleRule_API",
        "ruleSetName": null,
        "ruleTag": "",
        "ruleDataType": "STRING",
        "ruleSetDataType": null,
        "ruleSetTag": null,
        "stopExecution": false,
        "executionOrder": 2,

```

```

        "variableMapping": [
            {
                "variableName": "primaryVar",
                "fieldId": 11675,
                "fieldTechnicalName": "/Add/STATE",
                "varType": null
            }
        ],
        "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"
↪"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\
↪":true}}"
    },
    {
        "cdcColumn": false,
        "complexType": false
    },
    {
        "fieldId": 0,
        "position": 12,
        "fieldTechnicalName": "Description",
        "dataTypeId": 0,
        "fieldDataType": "STRING",
        "fieldDataTypeSchema": "{\\"name\\":\\"Description\\",\\"type\\":\\"
↪"STRING\\"}",
        "fieldDataLength": null,
        "dataScale": 0,
        "dataFormat": null,
        "fieldBusinessName": "Description",
        "description": null,
        "primary": false,
        "searchable": false,
        "masked": false,
        "tokenized": false,
        "nonTopLevel": false,
        "managedListId": 0,
        "managedListName": null,
        "tokenizerAlgorithmId": 0,
        "tokenizerAlgorithmName": null,
        "algoArguments": null,
        "maskingPattern": null,
        "startOffset": 0,
        "endOffset": 0,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleSetId": null,
                "ruleId": 8,
                "ruleName": "SimpleRule_API",
                "ruleSetName": null,
                "ruleTag": "",
                "ruleDataType": "STRING",
                "ruleSetDataType": null,
                "ruleSetTag": null,
                "stopExecution": false,
                "executionOrder": 2,
                "variableMapping": [
                    {

```

```

        "variableName": "primaryVar",
        "fieldId": 11675,
        "fieldTechnicalName": "/Add/STATE",
        "varType": null
      },
      ],
      "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\
↪\\"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\
↪":true}}"
    },
    ],
    "cdcColumn": false,
    "complexType": false
  }
],
"partitionFields": [],
"ruleSetEntityMapping": null,
"complexMetadataEntityMapping": [],
"complexEntityVersion": null,
"hcatTableId": 0,
"tableName": null,
"targetSchema": "auto",
"targetSchemaValue": "auto",
"delimiter": null,
"lineDelimiter": null,
"external": false,
"externalDataPath": null,
"externalDataPathValue": null,
"fileFormat": "DELIMITED",
"serdeLib": null,
"libClasspath": null,
"inputFormatLib": null,
"outputFormatLib": null,
"tableProperties":
  {
    "skip.header.line.count": "0",
    "myKey": "myValue"
  },
"storageDescriptorProperties": {},
"serdeInfoProperties": {},
"cdcActionable": false,
"hasInActiveTable": false,
"validateRequest": false,
"latestVersion": false,
"migrate": false,
"syncJob": false
},
"page": null
}

```

- Scenario 3:

If the *parser_type* is *AVRO*:

```

{
  "responseMessage": null,
  "restUri": null,
  "result": {

```

```

"entityTypeId": 0,
"version": 0,
"ruleSetId": 0,
"sharedWith": null,
"shared": null,
"ownerProjectId": null,
"ownerProjectName": null,
"ruleSetName": null,
"businessName": "tulya_avro",
"technicalName": "tulya_avro",
"description": "Entity generated from file [tulya_avro.avsc]",
"subjectArea": null,
"label": null,
"dataStoreId": 2,
"dataStoreName": "HDFS",
"frequencyOfUpdate": null,
"sourcePlatform": "undefined",
"sourceSchema": "undefined",
"loadType": null,
"preQueryStatements": null,
"dataFileFormatId": 0,
"dataFileFormat": "AVRO",
"metadata": null,
"recordLength": null,
"createdBy": null,
"createdTime": null,
"customTable": false,
"modifiedBy": null,
"modifiedTime": null,
"ingestionTime": null,
"newVersionRequired": false,
"securityContext": null,
"isDelete": null,
"blockDerivedEntityUpdate": false,
"derived": false,
"fields": [
  {
    "fieldId": 0,
    "position": 1,
    "fieldTechnicalName": "id",
    "dataTypeId": 0,
    "fieldDataTypes": "INTEGER",
    "fieldDataTypesSchema": "{ \"name\": \"id\", \"type\"
↪\": \"integer\" }",
    "fieldDataLength": null,
    "dataScale": 0,
    "dataFormat": null,
    "fieldBusinessName": "id",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": null,
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,

```

```

        "algoArguments": null,
        "maskingPattern": null,
        "startOffset": 0,
        "endOffset": 0,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleSetId": null,
                "ruleId": 8,
                "ruleName": "SimpleRule_API",
                "ruleSetName": null,
                "ruleTag": "",
                "ruleDataType": "STRING",
                "ruleSetDataType": null,
                "ruleSetTag": null,
                "stopExecution": false,
                "executionOrder": 2,
                "variableMapping": [
                    {
                        "variableName": "primaryVar",
                        "fieldId": 11675,
                        "fieldTechnicalName": "/Add/STATE",
                        "varType": null
                    }
                ],
                "expressionPayload": "{\"group\":{\"operator\":\"AND\",\\
↪\"rules\": [{\"data\": \"EQUALS_STRING(primaryVar, \\\"ZPL\\\")\"}], \"firstGroup\\
↪\": true}}\"
            }
        ],
        "cdcColumn": false,
        "customAttributes": null,
        "complexType": false
    }
]
"partitionFields": [],
"ruleSetEntityMapping": null,
"complexMetadataEntityMapping": [],
"complexEntityVersion": null,
"hcatTableId": 0,
"tableName": null,
"targetSchema": "auto",
"targetSchemaValue": "auto",
"delimiter": null,
"lineDelimiter": null,
"external": false,
"externalDataPath": null,
"externalDataPathValue": null,
"fileFormat": "AVRO",
"serdeLib": null,
"libClasspath": null,
"inputFormatLib": null,
"outputFormatLib": null,
"tableProperties": {
    "skip.header.line.count": "0",
    "myKey": "myValue"
}

```

```

        },
        "storageDescriptorProperties": {},
        "serdeInfoProperties": {},
        "cdcActionable": false,
        "hasInActiveTable": false,
        "validateRequest": false,
        "latestVersion": false,
        "migrate": false,
        "syncJob": false
    },
    "page": null
}

```

You can use the success response JSON from this API in the payload of the [Create an entity](#) API, to create an entity.

Response in Case of Failure:

- Scenario 1:

If a valid *AVRO/JSON/DELIMITED* file is not uploaded:

```

{
  "responseMessage": "Error while parsing schema from data file",
  "restUri": "/bedrock-app/services/rest/entities/schemas",
  "result": null,
  "page": null
}

```

- Scenario 2:

If the uploaded *AVRO* file details are not in sync with the selected file type (*Data* or *Schema*). For example, if you upload a *Data* file and select the *file_type* as *Schema*:

```

{
  "responseMessage": "Error while parsing schema from data file",
  "restUri": "/bedrock-app/services/rest/entities/schemas",
  "result": null,
  "page": null
}

```

- Scenario 3:

If a valid *DELIMITED* file has identical quote character and delimiter:

```

{
  "responseMessage": "Error while parsing schema from data file Delimiter and quote charecter must not be the same.",
  "restUri": "/bedrock-app/services/rest/entities/schemas",
  "result": null,
  "page": null
}

```

1.6.6 Fetch Details of an Entity

This API allows you to fetch the details of an entity type.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/entities/{entity_Id}/versions/
↳ {version_Id}?projectIds={project_Id}
```

Note: You must pass the *entity_Id*, *project_Id*, and *version_Id* parameters in the API URL. To fetch the details about an entity under the public project, set the *project_Id* parameter to *1*.

Method:

```
GET
```

Example Response:

- Scenario 1:

For a delimited *HDFS* entity:

```
{
  "responseMessage": "Success",
  "restUri": null,
  "result": {
    "entityTypeId": 4,
    "version": 1,
    "ruleSetId": 0,
    "ruleSetName": null,
    "businessName": "Zone4",
    "technicalName": "Zone4",
    "description": "",
    "subjectArea": "",
    "label": "",
    "dataStoreId": 2,
    "dataStoreName": "HDFS",
    "frequencyOfUpdate": "",
    "sourcePlatform": "spz4",
    "sourceSchema": "ssz4",
    "loadType": "",
    "preQueryStatements": null,
    "dataFileFormatId": 1,
    "dataFileFormat": "DELIMITED",
    "metadata": "{ \"hasComplexFields\": false, \"schemaUpload\": \"
↳ false, \"uiFields\": [ { \"name\": \"delimiter\", \"required\": false }, { \"
↳ name\": \"lineDelimiter\", \"required\": false } ] }",
    "recordLength": null,
    "createdBy": "admin",
    "createdTime": "06/06/2016 14:33:39",
    "customTable": false,
    "modifiedBy": "admin",
    "modifiedTime": "06/06/2016 17:02:35",
    "ingestionTime": null,
    "newVersionRequired": false,
    "securityContext": null,
    "isDelete": null,
    "derived": false,
    "fields": [
      {
        "fieldId": 11,
        "position": 1,
        "fieldTechnicalName": "Id",
```

```

        "dataTypeId": 10,
        "fieldDataType": "INT",
        "fieldDataTypeSchema": null,
        "fieldDataLength": "",
        "dataScale": 0,
        "dataFormat": "",
        "fieldBusinessName": "Id",
        "description": null,
        "primary": false,
        "searchable": false,
        "masked": false,
        "tokenized": false,
        "nonTopLevel": false,
        "managedListId": 0,
        "managedListName": null,
        "tokenizerAlgorithmId": 0,
        "tokenizerAlgorithmName": null,
        "maskingPattern": "",
        "startOffset": 0,
        "endOffset": 0,
        "sensitivityId": 0,
        "sensitivityValue": null,
        "ruleMappings": [
            {
                "ruleSetId": null,
                "ruleId": 8,
                "ruleName": "SimpleRule_API",
                "ruleSetName": null,
                "ruleTag": "",
                "ruleDataType": "STRING",
                "ruleSetDataType": null,
                "ruleSetTag": null,
                "stopExecution": false,
                "executionOrder": 2,
                "variableMapping": [
                    {
                        "variableName": "primaryVar",
                        "fieldId": 11675,
                        "fieldTechnicalName": "/Add/STATE",
                        "varType": null
                    }
                ],
                "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\\"
↪ "rules\\": [{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\
↪ ":true}}"
            }
        ],
        "cdcColumn": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "privRating
↪ ",
                        "value": "0",
                        "managed": false
                    }
                ]
            }
        ]
    }

```



```

        "id": "alias",
        "value": "Uniqueid"

        "managed": false
    }
    ],
    "source": {
        "type": "BR",
        "name": "Zaloni Bedrock"
    }
    },
    "complexType": false
},
{
    "fieldId": 12,
    "position": 2,
    "fieldTechnicalName": "EFID",
    "dataTypeId": 7,
    "fieldDataType": "STRING",
    "fieldDataTypeSchema": null,
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "EFID",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": null,
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,
    "maskingPattern": "",
    "startOffset": 0,
    "endOffset": 0,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "ruleMappings": [
        {
            "ruleSetId": null,
            "ruleId": 8,
            "ruleName": "SimpleRule_API",
            "ruleSetName": null,
            "ruleTag": "",
            "ruleDataType": "STRING",
            "ruleSetDataType": null,
            "ruleSetTag": null,
            "stopExecution": false,
            "executionOrder": 2,
            "variableMapping": [
                {
                    "variableName": "primaryVar",
                    "fieldId": 11675,
                    "fieldTechnicalName": "/Add/STATE",
                    "varType": null
                }
            ]
        }
    ]
}

```

```

    },
    "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\
↪\\"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\")\\"}],\\"firstGroup\\
↪":true}}"
  },
  ],
  "cdcColumn": false,
  "customAttributes": [
    {
      "attributes": [
        {
          "id": "privRating
↪",
          "value": "0",
          "managed": false
        },
        {
          "id": "alias",
          "value": "Uniqueid
↪",
          "managed": false
        }
      ],
      "source": {
        "type": "BR",
        "name": "Zaloni Bedrock"
      }
    }
  ],
  "complexType": false
},
{
  "fieldId": 10,
  "position": 3,
  "fieldTechnicalName": "Target",
  "dataTypeId": 7,
  "fieldDataType": "STRING",
  "fieldDataTypeSchema": null,
  "fieldDataLength": "",
  "dataScale": 0,
  "dataFormat": "",
  "fieldBusinessName": "Target",
  "description": null,
  "primary": false,
  "searchable": false,
  "masked": false,
  "tokenized": false,
  "nonTopLevel": false,
  "managedListId": 0,
  "managedListName": null,
  "tokenizerAlgorithmId": 0,
  "tokenizerAlgorithmName": null,
  "maskingPattern": "",
  "startOffset": 0,
  "endOffset": 0,
  "sensitivityId": 0,
  "sensitivityValue": null,

```

```

        "ruleMappings": [
            {
                "ruleSetId": null,
                "ruleId": 8,
                "ruleName": "SimpleRule_API",
                "ruleSetName": null,
                "ruleTag": "",
                "ruleDataType": "STRING",
                "ruleSetDataType": null,
                "ruleSetTag": null,
                "stopExecution": false,
                "executionOrder": 2,
                "variableMapping": [
                    {
                        "variableName": "primaryVar",
                        "fieldId": 11675,
                        "fieldTechnicalName": "/Add/STATE",
                        "varType": null
                    }
                ],
                "expressionPayload": "{\"group\":{\"operator\":\"AND\",\\
↪\"rules\":{\"data\":{\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")}},\\\"firstGroup\\
↪\":true}}\"
            }
        ],
        "cdcColumn": false,
        "customAttributes": [
            {
                "attributes": [
                    {
                        "id": "privRating",
                        "value": "0",
                        "managed": false
                    },
                    {
                        "id": "alias",
                        "value": "Uniqueid",
                        "managed": false
                    }
                ],
                "source": {
                    "type": "BR",
                    "name": "Zaloni Bedrock"
                }
            }
        ],
        "complexType": false
    },
    "partitionFields": [],
    "ruleSetEntityMapping": [],
    "complexMetadataEntityMapping": [],
    "complexEntityVersion": null,
    "hcatTableId": 4,
    "tableName": "spz4_ssz4_Zone4_1",
    "targetSchema": "zonesp",

```

```

        "targetSchemaValue": "zonesp",
        "delimiter": ",",
        "lineDelimiter": null,
        "external": true,
        "externalDataPath": "/user/bedrock/qwerty123",
        "externalDataPathValue": "/user/bedrock/qwerty123",
        "fileFormat": "TextFile",
        "serdeLib": "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
        "libClasspath": "",
        "inputFormatLib": "org.apache.hadoop.mapred.TextInputFormat",
        "outputFormatLib": "org.apache.hadoop.hive ql.io.
↪HiveIgnoreKeyTextOutputFormat",
        "tableProperties":
            {
                "skip.header.line.count": "0",
                "myKey": "myValue"
            },
        "storageDescriptorProperties": {},
        "serdeInfoProperties": {
            "": ""
        },
        "cdcActionable": true,
        "hasInActiveTable": false,
        "validateRequest": false,
        "latestVersion": true,
        "migrate": false,
        "syncJob": false
    },
    "page": null
}

```

- Scenario 2:

For a HDFS entity, when impersonation is enabled:

```

{
    "responseMessage": "Success",
    "restUri": null,
    "result": {
        "entityTypeId": 8,
        "version": 1,
        "ruleSetId": 0,
        "ruleSetName": null,
        "businessName": "Zone8",
        "technicalName": "Zone8",
        "description": "",
        "subjectArea": "",
        "label": "",
        "dataStoreId": 2,
        "dataStoreName": "HDFS",
        "frequencyOfUpdate": "",
        "sourcePlatform": "spz8",
        "sourceSchema": "ssz8",
        "loadType": "",
        "preQueryStatements": null,
        "dataFileFormatId": 1,
        "dataFileFormat": "DELIMITED",
        "metadata": "{ \"hasComplexFields\": false, \"schemaUpload\": ↪
↪false, \"uiFields\": [ { \"name\": \"delimiter\", \"required\": false }, { ↪
↪\"name\": \"lineDelimiter\", \"required\": false } ] }",

```

```

"recordLength": null,
"createdBy": "admin",
"createdTime": "06/06/2016 14:33:39",
"customTable": false,
"modifiedBy": "admin",
"modifiedTime": "06/06/2016 17:02:35",
"ingestionTime": null,
"newVersionRequired": false,
"securityContext": null,
"isDelete": null,
"derived": false,
"fields": [
  {
    "fieldId": 11,
    "position": 1,
    "fieldTechnicalName": "Id",
    "dataTypeId": 10,
    "fieldDataType": "INT",
    "fieldDataTypeSchema": null,
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "Id",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": null,
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,
    "maskingPattern": "",
    "startOffset": 0,
    "endOffset": 0,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "ruleMappings": [
      {
        "ruleSetId": null,
        "ruleId": 8,
        "ruleName": "SimpleRule_API",
        "ruleSetName": null,
        "ruleTag": "",
        "ruleDataType": "STRING",
        "ruleSetDataType": null,
        "ruleSetTag": null,
        "stopExecution": false,
        "executionOrder": 2,
        "variableMapping": [
          {
            "variableName": "primaryVar",
            "fieldId": 11675,
            "fieldTechnicalName": "/Add/STATE",
            "varType": null
          }
        ]
      }
    ]
  },

```

```

        "expressionPayload": "{\\"group\\":{\\"operator\\":\\"AND\\",\
↪\\"rules\\":[{\\"data\\":\\"EQUALS_STRING(primaryVar,\\\\"ZPL\\\\"}\\",\\"firstGroup\\
↪":true}}]"
    },
    "cdcColumn": false,
    "customAttributes": [
        {
            "attributes": [
                {
                    "id": "privRating",
                    "value": "0",
                    "managed": false
                },
                {
                    "id": "alias",
                    "value": "Uniqueid",
                    "managed": false
                }
            ],
            "source": {
                "type": "BR",
                "name": "Zaloni Bedrock"
            }
        }
    ],
    "complexType": false
},
{
    "fieldId": 10,
    "position": 3,
    "fieldTechnicalName": "Target",
    "dataTypeId": 7,
    "fieldDataType": "STRING",
    "fieldDataTypeSchema": null,
    "fieldDataLength": "",
    "dataScale": 0,
    "dataFormat": "",
    "dataFormat": "",
    "fieldBusinessName": "Target",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "tokenized": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": null,
    "tokenizerAlgorithmId": 0,
    "tokenizerAlgorithmName": null,
    "maskingPattern": "",
    "startOffset": 0,
    "endOffset": 0,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "ruleMappings": [

```

```

        {
            "ruleSetId": null,
            "ruleId": 8,
            "ruleName": "SimpleRule_API",
            "ruleSetName": null,
            "ruleTag": "",
            "ruleDataType": "STRING",
            "ruleSetDataType": null,
            "ruleSetTag": null,
            "stopExecution": false,
            "executionOrder": 2,
            "variableMapping": [
                {
                    "variableName": "primaryVar",
                    "fieldId": 11675,
                    "fieldTechnicalName": "/Add/STATE",
                    "varType": null
                }
            ],
            "expressionPayload": "{\"group\":{\"operator\":\"AND\",\\
↪\"rules\":{\"data\":{\"EQUALS_STRING(primaryVar,\\\"ZPL\\\")\"}},\"firstGroup\\
↪\":true}}\"
        },
        {
            "cdcColumn": false,
            "customAttributes": [
                {
                    "attributes": [
                        {
                            "id": "privRating",
                            "value": "0",
                            "managed": false
                        },
                        {
                            "id": "alias",
                            "value": "Uniqueid",
                            "managed": false
                        }
                    ],
                    "source": {
                        "type": "BR",
                        "name": "Zaloni Bedrock"
                    }
                }
            ],
            "complexType": false
        },
        {
            "partitionFields": [],
            "ruleSetEntityMapping": [],
            "complexMetadataEntityMapping": [],
            "complexEntityVersion": null,
            "hcatTableId": 7,
            "tableName": "spz8_ssz8_Zone8_1",
            "targetSchema": "zonesp",
            "targetSchemaValue": "zonesp",

```

```

        "delimiter": ",",
        "lineDelimiter": null,
        "external": true,
        "externalDataPath": "/user/bedrock/qwerty123",
        "externalDataPathValue": "/user/bedrock/qwerty123",
        "fileFormat": "TextFile",
        "serdeLib": "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
        "libClasspath": "",
        "inputFormatLib": "org.apache.hadoop.mapred.TextInputFormat",
        "outputFormatLib": "org.apache.hadoop.hive.ql.io.
↪HiveIgnoreKeyTextOutputFormat",
        "tableProperties":
            {
                "skip.header.line.count": "0",
                "myKey": "myValue"
            },
        "storageDescriptorProperties": {},
        "serdeInfoProperties": {
            "": ""
        },
        "ownerInputConfig": {
            "label": "Owner",
            "inputName": "owner",
            "elementId": "owner",
            "impersonationEnabled": true,
            "changedUser": "sau_pow"
        },
        "owner": "sau_pow",
        "position": 1,
        "targetSchemaCurrent": "${BEDROCK.custom_schema}",
        "externalDataPathCurrent": "",
        "cdcActionable": true
    }
}

```

Response in Case of Failure:

If entity Id is invalid:

```

{
    "responseMessage": "Artifact with Id <Entity_Id> and type ENTITY cannot be_
↪accessed from project with Id <Project_Id>",
    "restUri": "/bedrock-app/services/rest/entities/<Entity_Id>/versions/<Project_
↪Id>",
    "result": null,
    "page": null
}

```

1.6.7 Delete an Entity

This API allows you to delete an entity within a specific ZDP project.

URL:

```

http://<bedrock-host>:port/bedrock-app/services/rest/entities/{entity_Id}?projectId=
↪{project_Id}

```


Note: You must pass the *project_Id* parameter in the API URL. To delete an entity under the public project, set the *project_Id* parameter to 1.

Method:

DELETE

Example Response:

On succesful delete:

```
{
  "responseMessage": "Entity Type deleted successfully",
  "restUri": null,
  "result": "",
  "page": null
}
```

Response in Case of Failure:

- Scenario 1:

If the entity type with the specified *entity_Id* does not exist:

```
{
  "responseMessage": "Artifact with Id <Entity_Id> and type ENTITY cannot_
↪be accessed from project with Id <Project_Id>",
  "restUri": "/bedrock-app/services/rest/entities/<Entity_Id>",
  "result": null,
  "page": null
}
```

- Scenario 2:

If you try to delete a shared entity from a project that you do not have access to:

```
{
  "responseMessage": "User Bennigton doesn't have access to some of these_
↪permissions - md_manage_entity_type",
  "restUri": "/bedrock-app/services/rest/entities/13",
  "result": null,
  "page": null
}
```

1.6.8 Validate Structure of an Entity

This API allows you to validate the structure of an entity (i.e. the metadata of its' fields) within a specific project to compare with the structure defined in a meta file.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/entities/compare/schema/entity?
↪projectIds={project_Id}
```

Note: You must pass the *project_Id* parameter in the API URL. To validate the structure of an entity under the public project, set the *project_Id* parameter to 1.

Method:

POST

Request Payload:

```
{
    *"entityId": "<This is the Id of the entity type.>",
    "version": "<This is the version of the entity type. This is optional. If
↪not provided, the latest version is used.>",
    *"delimiter": "<This is the delimiter that separates the content of the meta
↪file. If the separator in the meta file is a regex character, prepend \\ to escape.>
↪",
    *"metaFilePath": "<This is the local path of the meta file.>"
}
```

Example Request:

- Scenario 1:

For a meta file with comma (,) as the delimiter:

```
{
  "entityId": 126,
  "delimiter": ",",
  "metaFilePath": "/home/brdev/schema/COMP_100.meta"
}
```

- Scenario 2:

For a meta file with a regex character as the delimiter, such as pipe (|):

```
{
  "entityId": 130,
  "delimiter": "\\|",
  "metaFilePath": "/home/brdev/schema/COMP_101.metax"
}
```

Example Response:

```
{
  "responseMessage": "Success",
  "restUri": null,
  "result": {
    "matchStatus": "matched"
  },
  "page": null
}
```

Response in Case of Failure:

If the meta file has issues:

```
{
  "responseMessage": "All mandatory attributes are not present in the metafile.
↪Missing attributes [technicalname,datatype,datascale,dataformat,datalength,primary,
↪position]",
}
```

```
"restUri": "/bedrock-app/services/rest/entities/compare/schema/entity",
"result": null,
"page": null
}
```

Important: This service requires a *schema.xml* file that must be present in the *config* directory of the ZDP platform. This schema file contains the aliases for the metadata attributes which the service checks for doing the comparison. The content of the default schema file is given below:

```
<?xml version="1.0"?>
<fieldschema>
<technicalname>
<value>COLUMN_NAME</value>
<value>FIELD_NAME</value>
</technicalname>
<businessname>
<value>BUSINESS_NAME</value>
</businessname>
<datatype>
<value>DATA_TYPE</value>
<value>FIELD_DATA_TYPE</value>
</datatype>
<datalength>
<value>DATA_LENGTH</value>
<value>FIELD_DATA_LENGTH</value>
</datalength>
<datascale>
<value>DATA_SCALE</value>
</datascale>
<dataformat>
<value>FORMAT</value>
<value>DATA_FORMAT</value>
</dataformat>
<primary>
<value>PRIMARY_KEYS</value>
<value>IS_PRIMARY</value>
</primary>
<position>
<value>COLUMN_ID</value>
</position>
</fieldschema>
</xml>
```

For example, to support *PRIMARY* as the attribute name for the field's primary constraint, add *PRIMARY* as a value under the primary attribute as follows:

```
<primary>
<value>PRIMARY_KEYS</value>
<value>IS_PRIMARY</value>
<value>PRIMARY</value>
</primary>
```

1.6.9 Filter and Fetch Entity Data

The following APIs allow you to fetch the data of specific entity from the Hive table by applying various filters/conditions.

By using the *Fetch data of an entity* API, you can pass the following options:

- Control the number of records that needs to be fetched.
- Set the column based on which the records need to be fetched.

By using the *Fetch entity data using conditions* API, you can set the following options in the request payload (in addition to the options above).

- Set a logical conditions based on which the result is fetched.
- Set additional sampling data type to fetch results.
- Sort the fetched result based on the sorting order and mode.

1.6.9.1 Fetch data of an entity

This API allows you to view the data of an entity based on the certain options (passed in the API URL).

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/entities/{entity_Id}/  
→versions/{entity_version}/data?sampleDataType={sample _type}&numberResults={number_  
→of_records}&submittedBy={username}&fields={comma_separated_field_name}&projectId=  
→{project_Id}
```

URL Parameters:

The *entity_Id*, *entity_version*, and *project_Id* parameters must be passed in the API URL. For example, to fetch sample data for an entity under the public project, set the *project_Id* value to *1*.

Pass the following URL parameters to fetch entity data according to the requirement.

- *sampleDataType*: Indicates the mode of fetching results. Set the value to *FIRSTN* and *UNBOUNDED* to fetch certain number of sample records and to fetch entire sample records (maximum *1000* records supported) respectively.
- *numberResults*: Indicates the number of sample data records to be fetched. This parameter is applicable only when *sampleDataType* is set as *FIRSTN*. If the parameter is not provided in the URL (or passed as URL parameters), the default sample records are fetched (upto *1000* records).
- *submittedBy*: Indicates the user submitting the request. The specified user must have necessary permissions to view the sample data for the specific entity. This parameter is mandatory only when impersonation is enabled.
- *fields*: Indicates the technical name of the entity field to fetch data for the specific column. For multiple fields, provide comma-separated list of values (within parenthesis).

Method:

```
GET
```

Example Response:

- Scenario 1:

For an entity with simple fields (such as Delimited or Fixed Length):

```

{
  "responseMessage": "Success",
  "restUri": null,
  "result": {
    "data": [
      [
        {
          "columnName": "linnn_1.id",
          "data": 1
        },
        {
          "columnName": "linnn_1.name",
          "data": "Heizel"
        }
      ],
      [
        {
          "columnName": "linnn_1.id",
          "data": 5
        },
        {
          "columnName": "linnn_1.name",
          "data": "Peter"
        }
      ]
    ]
  },
  "page": null
}

```

- Scenario 2:

For an entity (of *JSON* datatype) with complex (or nested) fields:

```

{
  "responseMessage": "Success",
  "restUri": null,
  "result": {
    "data": [
      [
        {
          "columnName": "heaven_flavours.batters",
          "data": {
            "batter": [
              {
                "id": "1001",
                "type": "Regular"
              },
              {
                "id": "1002",
                "type": "Chocolate"
              },
              {
                "id": "1003",
                "type": "Blueberry"
              }
            ]
          }
        }
      ]
    ]
  }
}

```

```

    {
        "id": "1004",
        "type": "Devil"
    }
]

},
{
    "columnName": "heaven_flavours.id",
    "data": "0001"
},
{
    "columnName": "heaven_flavours.name",
    "data": "Cake"
},
{
    "columnName": "heaven_flavours.ppu",
    "data": 0.55
},
{
    "columnName": "heaven_flavours.topping",
    "data": [
        {
            "id": "5001",
            "type": "None"
        },
        {
            "id": "5002",
            "type": "Glazed"
        },
        {
            "id": "5005",
            "type": "Sugar"
        },
        {
            "id": "5007",
            "type": "PowderedSugar"
        },
        {
            "id": "5006",
            "type":
→ "ChocolatewithSprinkles"

        },
        {
            "id": "5003",
            "type": "Chocolate"
        },
        {
            "id": "5004",
            "type": "Maple"
        }
    ]
},
{
    "columnName": "heaven_flavours.type",
    "data": "donut"
}

```

```

    ],
    "page": null
}

```

Response in Case of Failure:

- Scenario 1:

If no sample data is available for the specific entity:

```
204 No Content
```

- Scenario 2:

If the entity belongs to a project other than the one specified in the API URL:

```

{
  "responseMessage": "Artifact with Id <Artifact_Id> and type ENTITY cannot_
↪be accessed from project with Id <Entity_Id>",
  "restUri": "/bedrock-app/services/rest/metadata/entities/{Artifact_Id}/
↪versions/{Entity_Id}/sample-data",
  "result": null,
  "page": null
}

```

- Scenario 3:

If the entity field provided is incorrect (or the user does not have necessary permissions to view data):

```

{
  "responseMessage": "Error processing request. Make sure user has_
↪sufficient privilege and provided correct input",
  "restUri": null,
  "result": null,
  "page": null
}

```

- Scenario 4:

If the cluster gets disconnected and the service cannot fetch entity data:

```

{
  "responseMessage": "No connected cluster found",
  "restUri": null,
  "result": null,
  "page": null
}

```

1.6.9.2 Fetch entity data using conditions

This API allows you to fetch the data of an entity based on certain conditions set in the request payload.

URL:

```

http://<bedrock-host>:port/bedrock-app/services/rest/metadata/entities/{entity_Id}/
↪versions/{entity_version}/data/search?projectId={project_Id}&clusterId={cluster_Id}

```

Note: You must pass the *entity_Id*, the *entity_version*, the *project_Id*, and the *cluster_Id* as the *URL* parameters.

Method:

POST

Request Payload :

```
{
    "submittedBy": "<This is the user that owns the entity (or the table in Hive).>"
    ↪ ",
    "sampleDataType": "<This is the mode, based on which the sample data for the ↪
    ↪ entity will be fetched. Probable values can be 'FIRSTN', 'UNBOUNDED', 'RANDOM'. ↪
    ↪ When unspecified, the default ('FIRSTN') value is set.>",
    "fields": [ "<This is the single (or comma) seperated list of column names ↪
    ↪ (string) based on which the record values will be fetched. When unspecified, the ↪
    ↪ fetched result will consist records from all the columns.>" ],
    "maxNumberOfResults": "<This is the integer value limiting the maximum (or ↪
    ↪ limit) number of records that must be fetched. When unspecified, '1000' records are ↪
    ↪ fetched by default.>",
    "seedValue": "<This is the value (in decimal) that indicates the percentage of ↪
    ↪ records that needs to be fetched. This sets a poll to fetch the minimum records by ↪
    ↪ comparing the values set for 'maxNumberOfResults' and 'seedValue' itself.>"
    "samplingCondition": {
        "whereClause": "<This is a condition for the column values using ↪
    ↪ logical operators. Records fulfilling the condition are filtered and fetched ↪
    ↪ accordingly.>",
        "orderByConditions": {
            "columnNames": [
                "<This is the single (or comma) seperated list of ↪
    ↪ column names (string) based on which the fethced results are sorted.>"
            ],
            "orderByClause": "<This is the order mode based on which the ↪
    ↪ fetched result must be sorted. Probable value can be 'ASC' or 'DESC'.>"
        }
    }
}
```

Example Request:

To fetch *FIRSTN* records (where *N=4*) of the specific entity:

```
{
    "submittedBy": "zabber",
    "sampleDataType": "FIRSTN",
    "maxNumberOfResults": 400,
    "seedValue": "0.005",
    "fields": [
        "id", "first_name", "last_name"
    ],
    "samplingCondition": {
        "whereClause": "id>2304",
        "orderByConditions": {
            "columnNames": [
                "id"
            ],
            "orderByClause": "DESC"
        }
    }
}
```



```
}
    }
}
```

Example Response:

```
{
  "responseMessage": "Success",
  "restUri": null,
  "result": {
    "data": [
      [
        {
          "columnName": "id",
          "data": 2813
        },
        {
          "columnName": "f_name",
          "data": "Nick"
        },
        {
          "columnName": "l_name",
          "data": "Mason"
        }
      ],
      [
        {
          "columnName": "id",
          "data": 2812
        },
        {
          "columnName": "f_name",
          "data": "Sydd"
        },
        {
          "columnName": "l_name",
          "data": "Barrett"
        }
      ],
      [
        {
          "columnName": "id",
          "data": 2811
        },
        {
          "columnName": "f_name",
          "data": "Roger"
        },
        {
          "columnName": "l_name",
          "data": "Waters"
        }
      ],
      [
        {
          "columnName": "id",
          "data": 2810
        }
      ]
    ]
  }
}
```

```

        {
            "columnName": "f_name",
            "data": "Richard"
        },
        {
            "columnName": "l_name",
            "data": "Wright"
        }
    ]
},
"page": null
}

```

Response in case of failure:

The following failure responses can be expected due to certain scenarios:

- Scenario 1:

If no records meet the filter condition:

```
204 No Content
```

- Scenario 2:

If the parameters in the request payload have incorrect values:

```

{
    "responseMessage": "Exception while executing query.",
    "restUri": "/bedrock-app/services/rest/metadata/entities/241/versions/2/
↪data/search",
    "result": null,
    "page": null
}

```

1.6.10 Profile Data

1.6.10.1 Fetch entity level profile metrics

This API allows you to fetch the profile metrics of an entity type. The profile metrics is displayed only if the entity has undergone data profiling.

URL:

```

http://<bedrock-host>:port/bedrock-app/services/rest/dataprofiles/entities/{entity_Id}
↪/versions/{version_Id}

```

Note: You must pass the *entity_Id* and *version_Id* parameters in the API URL.

Method:

```
GET
```

Example Response:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "success",
    "result": null
  },
  "result": {
    "businessName": "Dp_sequence44",
    "subjectArea": null,
    "partition": null,
    "sourcePlatform": "Dp_sequence44",
    "sourceSchema": "Dp_sequence44",
    "recordCount": 1000,
    "averageRecordLength": 45,
    "columnCount": 5,
    "fileSizeInBytes": 45576,
    "numberOfFiles": 1,
    "dateOfProfiling": "2016-03-04T07:36:31.203+0000",
    "dateSchemaCreated": "2016-03-03T14:24:28.000+0000",
    "dateSchemaModified": "2016-03-04T07:35:33.000+0000",
    "dateLastIngested": null
  }
}
```

Response in Case of Failure:

If the entity has not been profiled:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 404,
    "responseMessage": "error",
    "result": null
  },
  "result": "No profile data found for the entityId <Entity Id> and version
  ↳<Entity Version>. Profile data will be available only after you have run profiling
  ↳on this entity at least once."
}
```

1.6.10.2 Fetch field summary of profiled entity

This API allows you to fetch the profile metrics of all fields of an entity Type. The metrics is displayed only if the entity has undergone data profiling.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dataprofiles/entities/{entity_Id}
  ↳/versions/{version_Id}/field-summary
```

Note: You must pass the *entity_Id* and *version_Id* parameters in the API URL.

Method:

GET

Example Response:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "success",
    "result": null
  },
  "result": [
    {
      "fieldId": 90,
      "dataType": "INT",
      "fieldBusinessName": "id",
      "fieldTechnicalName": "id",
      "nullCount": 0,
      "distinctCount": 1000
    },
    {
      "fieldId": 89,
      "dataType": "STRING",
      "fieldBusinessName": "name",
      "fieldTechnicalName": "name",
      "nullCount": 0,
      "distinctCount": 198
    },
    {
      "fieldId": 91,
      "dataType": "BOOLEAN",
      "fieldBusinessName": "gender",
      "fieldTechnicalName": "gender",
      "nullCount": 0,
      "distinctCount": null
    },
    {
      "fieldId": 92,
      "dataType": "LONG",
      "fieldBusinessName": "pin",
      "fieldTechnicalName": "pin",
      "nullCount": 0,
      "distinctCount": 1000
    },
    {
      "fieldId": 93,
      "dataType": "DOUBLE",
      "fieldBusinessName": "salary",
      "fieldTechnicalName": "salary",
      "nullCount": 0,
      "distinctCount": 1000
    }
  ]
}
```

Response in Case of Failure:

If the entity fields have no profile statistics indicating that the entity type (or its' version) has not been profiled:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "success",
    "result": null
  },
  "result": []
}
```

1.6.10.3 Fetch profile metrics entity fields

This API allows you to fetch the statistics of an entity field. The metrics are displayed only for the entity that has been profiled earlier.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dataprofiles/entities/{entity_Id}
↪/versions/{version_Id}/fields/{field_Id}
```

Note: You must pass the *entity_Id*, *version_Id*, and the *field_Id* parameters of the profiled entity in the API URL. To get the Id of a field (*field_Id*), use the *Fetch Field Summary of Profiled Entity* API.

Method:

```
GET
```

Example Response:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "success",
    "result": null
  },
  "result": {
    "fieldId": 91,
    "fieldPosition": 3,
    "dataType": "BOOLEAN",
    "fieldBusinessName": "gender",
    "fieldTechnicalName": "gender",
    "nullCount": 0,
    "blankCount": null,
    "invalidCount": 0,
    "distinctCount": null,
    "totalCount": 1000,
    "mostCommonValues": {
      "title": "Most Frequently Occurring Values for gender",
      "namedValueCollection": [
        {
          "label": "true",
          "value": 512
        }
      ]
    }
  }
}
```

```

        "label": "false",
        "value": 488
      }
    ],
    "leastCommonValues": {
      "title": "Least Frequently Occurring Values for gender",
      "namedValueCollection": [
        {
          "label": "false",
          "value": 488
        },
        {
          "label": "true",
          "value": 512
        }
      ]
    },
    "histogram": {
      "title": "Histogram for gender",
      "namedValueCollection": [
        {
          "label": "true",
          "value": 512
        },
        {
          "label": "false",
          "value": 488
        }
      ]
    },
    "dataStatistic": {
      "average": null,
      "standardDeviation": null,
      "median": "true",
      "minimum": null,
      "maximum": null
    }
  }
}

```

Response in Case of Failure:

If the entity field has no records or the entity type (or its' version) has not been profiled:

```

{
  "status": {
    "responseType": "info",
    "responseCode": 404,
    "responseMessage": "error",
    "result": null
  },
  "result": "No profile data found for the entityId <Entity Id> and version  

  ↳<Entity Version>. Profile data will be available only after you have run profiling_  

  ↳on this entity at least once."
}

```

1.6.10.4 Search and fetch profile metrics of entities

This API allows you to search and fetch profile metrics of the multiple entities that are queried.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dataprofiles/search?  
↪filterType=entities_versions
```

Method:

```
POST
```

Input Parameter:

The input parameter comprises an array of entities (along with version number) as follows:

```
*["entityid1_entityVersion1","entityid2_entityVersion1","entityid2_entityVersion3"]
```

Example Request:

```
["16_1","15_1"]
```

Example Response:

```
{  
  "status": {  
    "responseType": "info",  
    "responseCode": 200,  
    "responseMessage": "success",  
    "result": null  
  },  
  "result": {  
    "16_1": {  
      "businessName": "Dp_sequence44",  
      "subjectArea": null,  
      "partition": null,  
      "sourcePlatform": "Dp_sequence44",  
      "sourceSchema": "Dp_sequence44",  
      "recordCount": 1000,  
      "averageRecordLength": 45,  
      "columnCount": 5,  
      "fileSizeInBytes": 45576,  
      "numberOfFiles": 1,  
      "dateOfProfiling": "2016-03-04T07:36:31.203+0000",  
      "dateSchemaCreated": "2016-03-03T14:24:28.000+0000",  
      "dateSchemaModified": "2016-03-04T07:35:33.000+0000",  
      "dateLastIngested": null  
    },  
    "15_1": {  
      "businessName": "Dp_fixedLength66",  
      "subjectArea": null,  
      "partition": null,  
      "sourcePlatform": "Dp_fixedLength66",  
      "sourceSchema": "Dp_fixedLength66",  
      "recordCount": 2,  
      "averageRecordLength": 11,  
      "columnCount": 2,  
      "fileSizeInBytes": 22,  
    }  
  }  
}
```

```
{
  "numberOfFiles": 1,
  "dateOfProfiling": "2016-03-03T14:19:40.263+0000",
  "dateSchemaCreated": "2016-03-03T14:14:50.000+0000",
  "dateSchemaModified": "2016-03-03T14:14:50.000+0000",
  "dateLastIngested": null
}
```

Response in Case of Failure:

If the entity field has no records or the entity type (or its' version) has not been profiled:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "success",
    "result": null
  },
  "result": {
    "12_1": null,
    "2_1": null
  }
}
```

1.6.11 Manage Data Type Mappings

1.6.11.1 Fetch data quality datatypes

This API allows you to fetch the DQ data type metrics supported in ZDP. The result fetched can be used to map DQ HDFS and HCatalog data types.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dq/rules/fetchDataType
```

Method:

```
GET
```

Example Response:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Success",
    "result": null
  },
  "result": [
    {
      "dataTypeId": 4,
      "dataType": "BOOLEAN"
    },
    {

```



```

        "dataTypeId": 8,
        "dataType": "COMPOUND"
      },
      {
        "dataTypeId": 3,
        "dataType": "DATE"
      },
      {
        "dataTypeId": 7,
        "dataType": "ENTITY"
      },
      {
        "dataTypeId": 5,
        "dataType": "INTERVAL"
      },
      {
        "dataTypeId": 2,
        "dataType": "NUMBER"
      },
      {
        "dataTypeId": 1,
        "dataType": "STRING"
      },
      {
        "dataTypeId": 6,
        "dataType": "UNKNOWN"
      }
    ]
  }
}

```

1.6.11.2 Fetch HCatalog datatypes

This API allows you to fetch the HCatalog data type supported in ZDP. The result fetched can be used for DQ, HDFS, and HCatalog data type mapping.

URL:

```

http://<bedrock-host>:port/bedrock-app/services/rest/metadata/entity/
→ fetchHCatalogDataTypes

```

Method:

```
GET
```

Example Response:

```

{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Success",
    "result": null
  },
  "result": [
    "INT",
    "TINYINT",
    "SMALLINT",

```

```
        "BIGINT",
        "BOOLEAN",
        "FLOAT",
        "DOUBLE",
        "STRING",
        "ARRAY",
        "MAP",
        "STRUCT",
        "BINARY",
        "DATE",
        "TIMESTAMP",
        "DECIMAL"
    ]
}
```

1.6.11.3 Fetch datatype mappings

Use this service to fetch all the HCatalog to Data Quality datatype mappings existing in a particular ZDP instance.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/datatypes/mappings
```

Method:

```
GET
```

Example Response:

Displaying the existing datatype mappings across *dqDataType* and *HCatalogDataTypes*.

Note: Datatype mapping Id 0 indicates no mapping exist for the specific type (*dqDataType*/*HCatalogDataTypes*).

```
{
    "responseMessage": "Hcatalog datatype mappings fetched successfully.",
    "restUri": null,
    "result": [
        {
            "dataTypeId": 7,
            "dataType": "STRING",
            "mappedHCatalogDataType": "STRING",
            "dqDataTypeId": 1,
            "dqDataType": "STRING",
            "deletable": false
        },
        {
            "dataTypeId": 8,
            "dataType": "TINYINT",
            "mappedHCatalogDataType": "TINYINT",
            "dqDataTypeId": 2,
            "dqDataType": "NUMBER",
            "deletable": true
        },
        {
            "dataTypeId": 9,
```

```

        "dataType": "SMALLINT",
        "mappedHCatalogDataType": "SMALLINT",
        "dqDataTypeId": 2,
        "dqDataType": "NUMBER",
        "deletable": true
    },
    {
        "dataTypeId": 10,
        "dataType": "INT",
        "mappedHCatalogDataType": "INT",
        "dqDataTypeId": 2,
        "dqDataType": "NUMBER",
        "deletable": false
    },
    {
        "dataTypeId": 11,
        "dataType": "BIGINT",
        "mappedHCatalogDataType": "BIGINT",
        "dqDataTypeId": 2,
        "dqDataType": "NUMBER",
        "deletable": true
    },
    {
        "dataTypeId": 12,
        "dataType": "BOOLEAN",
        "mappedHCatalogDataType": "BOOLEAN",
        "dqDataTypeId": 4,
        "dqDataType": "BOOLEAN",
        "deletable": true
    },
    {
        "dataTypeId": 13,
        "dataType": "FLOAT",
        "mappedHCatalogDataType": "FLOAT",
        "dqDataTypeId": 2,
        "dqDataType": "NUMBER",
        "deletable": true
    },
    {
        "dataTypeId": 14,
        "dataType": "DOUBLE",
        "mappedHCatalogDataType": "DOUBLE",
        "dqDataTypeId": 2,
        "dqDataType": "NUMBER",
        "deletable": true
    },
    {
        "dataTypeId": 15,
        "dataType": "BINARY",
        "mappedHCatalogDataType": "BINARY",
        "dqDataTypeId": 0,
        "deletable": true
    },
    {
        "dataTypeId": 16,
        "dataType": "ARRAY",
        "mappedHCatalogDataType": "ARRAY",
        "dqDataTypeId": 0,

```

```

        "deletable": false
    },
    {
        "dataTypeId": 17,
        "dataType": "MAP",
        "mappedHCatalogDataType": "MAP",
        "dqDataTypeId": 0,
        "deletable": true
    },
    {
        "dataTypeId": 18,
        "dataType": "RECORD",
        "mappedHCatalogDataType": "STRUCT",
        "dqDataTypeId": 0,
        "deletable": false
    },
    {
        "dataTypeId": 19,
        "dataType": "INTEGER",
        "mappedHCatalogDataType": "INT",
        "dqDataTypeId": 2,
        "dqDataType": "NUMBER",
        "deletable": true
    },
    {
        "dataTypeId": 20,
        "dataType": "DATE",
        "mappedHCatalogDataType": "DATE",
        "dqDataTypeId": 3,
        "dqDataType": "DATE",
        "deletable": true
    },
    {
        "dataTypeId": 21,
        "dataType": "TIMESTAMP",
        "mappedHCatalogDataType": "TIMESTAMP",
        "dqDataTypeId": 3,
        "dqDataType": "DATE",
        "deletable": true
    },
    {
        "dataTypeId": 22,
        "dataType": "DECIMAL",
        "mappedHCatalogDataType": "DECIMAL",
        "dqDataTypeId": 2,
        "dqDataType": "NUMBER",
        "deletable": true
    },
    {
        "dataTypeId": 23,
        "dataType": "CHAR",
        "mappedHCatalogDataType": "STRING",
        "dqDataTypeId": 1,
        "dqDataType": "STRING",
        "deletable": true
    },
    {
        "dataTypeId": 24,

```

```

        "dataType": "BYTE",
        "mappedHCatalogDataType": "STRING",
        "dqDataTypeId": 1,
        "dqDataType": "STRING",
        "deletable": true
    },
    {
        "dataTypeId": 25,
        "dataType": "SHORT",
        "mappedHCatalogDataType": "TINYINT",
        "dqDataTypeId": 2,
        "dqDataType": "NUMBER",
        "deletable": true
    },
    {
        "dataTypeId": 26,
        "dataType": "LONG",
        "mappedHCatalogDataType": "BIGINT",
        "dqDataTypeId": 2,
        "dqDataType": "NUMBER",
        "deletable": true
    },
    {
        "dataTypeId": 27,
        "dataType": "BIGINTEGER",
        "mappedHCatalogDataType": "BIGINT",
        "dqDataTypeId": 2,
        "dqDataType": "NUMBER",
        "deletable": true
    },
    {
        "dataTypeId": 28,
        "dataType": "BIGDECIMAL",
        "mappedHCatalogDataType": "DECIMAL",
        "dqDataTypeId": 2,
        "dqDataType": "NUMBER",
        "deletable": true
    },
    {
        "dataTypeId": 29,
        "dataType": "TIME",
        "mappedHCatalogDataType": "STRING",
        "dqDataTypeId": 0,
        "deletable": true
    },
    {
        "dataTypeId": 30,
        "dataType": "OBJECT",
        "mappedHCatalogDataType": "STRING",
        "dqDataTypeId": 0,
        "deletable": true
    },
    {
        "dataTypeId": 31,
        "dataType": "BLOB",
        "mappedHCatalogDataType": "BINARY",
        "dqDataTypeId": 0,
        "deletable": true
    }

```

```

    },
    {
      "dataTypeId": 32,
      "dataType": "CLOB",
      "mappedHCatalogDataType": "STRING",
      "dqDataTypeId": 0,
      "deletable": true
    },
    {
      "dataTypeId": 33,
      "dataType": "XML",
      "mappedHCatalogDataType": "STRING",
      "dqDataTypeId": 0,
      "deletable": true
    },
    {
      "dataTypeId": 34,
      "dataType": "STRUCT",
      "mappedHCatalogDataType": "STRUCT",
      "dqDataTypeId": 0,
      "deletable": true
    }
  ],
  "page": null
}

```

1.6.11.4 Add custom datatype

This API allows you to add a customized HDFS data type in a ZDP server instance. The customized data type (hdfs data type) must be mapped to a corresponding HCatalog and data quality data type supported in ZDP.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/entity/config/datatype/
↪ save
```

Method:

POST

Request Payload:

```

{
  *"hdfsDataType": "<This is the custom data type name that will be added.>",
  *"mappedHCatalogDataType": "<This is the HCatalog datatype for mapping.>",
  *"dqDataType": "<This is the data type of data quality for mapping.>",
  *"dqDataTypeId": "<This is the Id of the selected data quality data type.>"
}

```

- To get the *mappedHCatalogDataType* for HCatalog data type mapping, use the *Fetch HCatalog Datatypes* API.
- To get the *dqDataType* and *dqDataTypeId* mappings, use the *Fetch Data Quality Datatypes* API.

Example Request:

```

{
  "dataType": "MEDRECTYPE",

```

```

    "mappedHCatalogDataType": "INT",
    "dqDataType": "NUMBER",
    "dqDataTypeId": 2
  }

```

Example Response:

```

{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Configuration saved successfully, exited after_
↪business implementation execution.",
    "result": 17
  },
  "result": null
}

```

1.6.12 Manage ZDP Entities

1.6.12.1 Create a ZDP managed entity

This API allows you to create a ZDP managed entity.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/{project_Id}/entities
```

Note:

- You must pass the *project_Id* parameter in the API URL. To create an entity under the public project, set the *project_Id* parameter to *1*.
- For more information on request parameters, refer to the *Create/Update an Entity* section.

Method:

```
POST
```

Example Request:

To create an entity with *BEDROCK* as the data store.

```

{
  "businessName": "BedrockEntity",
  "technicalName": "BRE",
  "description": "",
  "subjectArea": "",
  "label": "",
  "dataStoreId": 1,
  "hcatIntegrated": true,
  "dataFileFormatId": 0,
  "delimiter": null,
  "entityCreateOptions": 1,
  "complexEntityVersion": null,
}

```

```

"recordLength": null,
"frequencyOfUpdate": "",
"sourcePlatform": "BR",
"sourceSchema": "bedrock_mica",
"loadType": "",
"entityTypeId": 0,
"version": 1,
"sharedWith": null,
"shared": null,
"ownerProjectId": null,
"ownerProjectName": null,
"ruleSetName": null,
"dataStoreName": "BEDROCK",
"preQueryStatements": null,
"dataFileFormat": null,
"metadata": null,
"createdBy": "admin",
"createdTime": "01/17/2017 16:37:05",
"modifiedBy": "admin",
"modifiedTime": "01/17/2017 16:37:05",
"newVersionRequired": false,
"securityContext": null,
"isDelete": null,
"blockDerivedEntityUpdate": false,
"derived": false,
"fields": [
  {
    "fieldId": -1,
    "position": 1,
    "fieldTechnicalName": "e1",
    "dataTypeId": 1,
    "fieldDataType": "INTEGER",
    "fieldDataTypeSchema": null,
    "fieldDataLength": "11",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "e1",
    "description": null,
    "primary": false,
    "searchable": false,
    "nonTopLevel": false,
    "managedListId": 0,
    "managedListName": null,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "customAttributes": [
      {
        "attributes": [
          {
            "id": "",
            "value": "",
            "managed": false
          }
        ]
      },
      {
        "source": {
          "type": "BR",
          "name": "Zaloni Bedrock"
        }
      }
    ]
  }
]

```



```

    },
    "complexType": false
  },
  {
    "fieldId": -2,
    "position": 2,
    "fieldTechnicalName": "e2",
    "dataTypeId": 2,
    "fieldDataType": "VARCHAR",
    "fieldDataTypeSchema": null,
    "fieldDataLength": "255",
    "dataScale": 0,
    "dataFormat": "",
    "fieldBusinessName": "e2",
    "description": null,
    "primary": false,
    "searchable": false,
    "masked": false,
    "managedListId": 0,
    "managedListName": null,
    "sensitivityId": 0,
    "sensitivityValue": null,
    "customAttributes": [
      {
        "attributes": [
          {
            "id": "",
            "value": "",
            "managed": false
          }
        ],
        "source": {
          "type": "BR",
          "name": "Zaloni Bedrock"
        }
      }
    ],
    "complexType": false
  },
  ],
  "validateRequest": false,
  "latestVersion": true,
  "migrate": false,
  "import": false,
  "syncJob": false,
  "position": 1
}

```

1.6.12.2 Create an entity instance

This API allows you to create an entity instance of a *BEDROCK* type entity already created in the ZDP instance.

URL:

```

http://<bedrock-host>:port/bedrock-app/services/rest/metadata/publish/
↪ saveEntityInstance

```

Method:

POST

Request Payload:

```
{
  *"entityId": "<This is the entity type Id for which the entity instance is being_
  ↳created. This entity type should have data store defined as BEDROCK.>",
  *"entityVersion": "<This is the version of the entity type for which the entity_
  ↳instance is being created.>",
  *"instanceObject": "<This holds the actual instance data to be stored.>"
                                {
                                "field business name": "<This is the_
  ↳value to be stored for the field.>"
                                }
}
```

Example Request:

```
{
  "entityId": 4,
  "entityVersion": 1,
  "instanceObject": {
    "Total Experiences": "3.23",
    "Taxable": "true",
    "Employee Id": "13",
    "Employee Name": "Mac",
    "Gender": "male",
    "Hire Date": "12/01/2013 01:01:01",
    "Department Id": "1",
    "Department Name": "Admin"
  }
}
```

Example Response:

If the entity instance is saved successfully:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Entity instance saved successfully with entity instance_
  ↳id :<entityInstanceId> and version :<entityInstanceVersion>",
    "result": null
  },
  "result": "Entity instance saved successfully with entity instance id :3 and_
  ↳version :1"
}
```

Response in case of Failure:

- Scenario 1:

If the entity type for which the entity instance is being created does not belong to the *BEDROCK* datastore:

```
{
  "status": {
    "responseType": "ERROR",
```

```
        "responseCode": 500,  
        "responseMessage": "Cannot save entity instance. Trying to create_  
↪entity instance for non compatible datastore",  
        "result": null  
    },  
    "result": ""  
}
```

- Scenario 2:

If the entity type for which the entity instance is being created, does not exist:

```
{  
    "status": {  
        "responseType": "ERROR",  
        "responseCode": 500,  
        "responseMessage": "Cannot save entity instance. Entity type does_  
↪not exist",  
        "result": null  
    },  
    "result": ""  
}
```

- Scenario 3:

If the entity type Id and version is not specified in the request JSON:

```
{  
    "status": {  
        "responseType": "ERROR",  
        "responseCode": 500,  
        "responseMessage": "Cannot save entity instance. Entity type not_  
↪specified",  
        "result": null  
    },  
    "result": ""  
}
```

- Scenario 4:

If the *instanceObject* attribute is missing from the request JSON:

```
{  
    "status": {  
        "responseType": "ERROR",  
        "responseCode": 500,  
        "responseMessage": "Cannot save entity instance. Entity Type_  
↪Instance payload is missing instanceObject data.",  
        "result": null  
    },  
    "result": ""  
}
```

- Scenario 5:

If the *instanceObject* attribute has no values:

```
{  
    "status": {
```

```

        "responseType": "ERROR",
        "responseCode": 500,
        "responseMessage": "Cannot save entity instance. Entity Type_
↪instance has empty or no values",
        "result": null
    },
    "result": ""
}

```

- Scenario 6:

If the *instanceObject* contains field names that are not defined for the entity type and revision:

```

{
    "status": {
        "responseType": "ERROR",
        "responseCode": 500,
        "responseMessage": "Cannot save entity instance. Entity instance_
↪defined with invalid field name(s)",
        "result": null
    },
    "result": ""
}

```

- Scenario 7:

If there is any data validation errors for the data provided in the *instanceObject*.

For example, if non integer value is defined for a field having integer as data type:

```

{
    "status": {
        "responseType": "ERROR",
        "responseCode": 500,
        "responseMessage": "Cannot save entity instance. Error validating_
↪entity Instance: [The field <Field_Name> is of type <Data_Type>. Specify_
↪appropriate value in place of \"fg\"]",
        "result": null
    },
    "result": ""
}

```

1.6.12.3 Edit entity instance

This API allows you to edit an instance of a specific entity type.

URL:

```

http://<bedrock-host>:port/bedrock-app/services/rest/metadata/publish/
↪saveEntityInstance

```

Method:

```
POST
```

Request Payload:

```
{
  "entityId": "<This is the entity type Id for which the entity instance was created.
  ↳ This value is optional for instance edit.>",
  "entityVersion": "<This is the version of the entity type for which the
  ↳ entity instance was created. This value is optional for instance edit.>",
  * "entityInstanceId": "<This is the Id of the entity instance which is to be
  ↳ updated.>",
  * "entityInstanceVersion": "<This is the version of the entity instance which is to
  ↳ be updated.>",
  "newVersionRequired": "<This is the boolean value (True or False) to determine if
  ↳ a new revision needs to be created for the entity instance.>",
  * "instanceObject ": "<This holds the actual instance data to be stored.>"
                                {
                                "field business name": "<This is
  ↳ the value to be stored for the field.>"
                                }
}
```

Example Request:

```
{
  "newVersionRequired": false,
  "entityInstanceId": 15,
  "entityInstanceVersion": 1,
  "instanceObject": {
    "Taxable": "true",
    "Employee Id": "23",
    "Employee Name": "McDonald",
    "Gender": "male",
    "Hire Date": "12/01/2013 01:01:01",
    "Department Id": "1",
    "Department Name": "Admin"
  },
}
```

Example Response:

- Scenario 1:

If the entity instance is updated successfully:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Entity Instance updated successfully",
    "result": null
  },
  "result": "Entity Instance updated successfully"
}
```

- Scenario 2:

If the entity instance is updated to a new revision:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
```

```
        "responseMessage": "Entity Instance saved successfully with new_↵  
↵revision",  
        "result": null  
    },  
    "result": "Entity Instance saved successfully with new revision"  
}
```

Response in Case of Failure:

- Scenario 1:

If you update the non-latest version of entity instance:

```
{  
    "status": {  
        "responseType": "ERROR",  
        "responseCode": 500,  
        "responseMessage": "Cannot save entity instance. Trying to edit a_↵  
↵non latest entity instance",  
        "result": null  
    },  
    "result": ""  
}
```

- Scenario 2:

If the request JSON contains *entityInstanceId* and *entityInstanceVersion* different from the ones for which the instance was created.

```
{  
    "status": {  
        "responseType": "ERROR",  
        "responseCode": 500,  
        "responseMessage": "Cannot save entity instance. Trying to edit_↵  
↵non existing entity instance",  
        "result": null  
    },  
    "result": ""  
}
```

- Scenario 3:

If the entity instance does not exist for the provided entity instance Id and version:

```
{  
    "status": {  
        "responseType": "ERROR",  
        "responseCode": 500,  
        "responseMessage": "Cannot save entity instance. Trying to edit_↵  
↵non existing entity instance",  
        "result": null  
    },  
    "result": ""  
}
```

Note: Any other error response is similar to that defined for the *Create an entity instance* API.

1.6.12.4 Fetch details of entity instance

This API allows you to fetch the details of an entity instance.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/publish/  
→fetchEntityInstanceDetails/{entity_instance_Id}/{entity_instance_version}
```

Note: You must pass the *entity_instance_Id* and *entity_instance_version* parameters in the API URL.

Method:

```
GET
```

Example Response:

If the *entity_instance_version* parameter is provided as 0, the system defaults to the latest version of the entity instance:

```
{  
  "status": {  
    "responseType": "info",  
    "responseCode": 200,  
    "responseMessage": "Success",  
    "result": null  
  },  
  "result": {  
    "entityInstanceId": 7,  
    "entityInstanceVersion": 1,  
    "entityBusinessName": "bedrock",  
    "createdBy": "admin",  
    "createdTime": "11/16/2016 13:40:00",  
    "modifiedBy": "admin",  
    "modifiedTime": "11/16/2016 13:40:00",  
    "instanceObject": {  
      "id": null,  
      "name": null  
    },  
    "latestVersion": true,  
    "newVersionRequired": false,  
    "entityId": 9,  
    "entityVersion": 1,  
    "entityInstanceSearchables": [],  
    "searchableFieldValues": null,  
    "entityInstancePrimaryValues": []  
  }  
}
```

Response in Case of Failure:

If the entity instance does not exist for the given instance Id and version:

```
{  
  "status": {  
    "responseType": "info",  
    "responseCode": 204,  
    "responseMessage": "No Entity Instance details found",  
    "result": null  
  }  
}
```

```
} ,
  "result": null
}
```

1.6.12.5 Search entity instances

This API allows you to search for entity instances.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/publish/
↳searchEntityInstanceDetails
```

Method:

```
POST
```

Request Payload:

Use any one of the following parameters:

```
{
  "entityTypeId": "<This the Id for the entity type for which the instances are_
↳searched.>",
  "entityTypeVersion": "<This is the revision of the entity type for which the_
↳instances are searched. This is optional, if not provided system considers the_
↳latest version>",
  "startDate": "<This is the start date of the date range for the created date._
↳If the end date is provided, the start date is mandatory. The date format must be_
↳mm/dd/yyyy HH:MM:SS.>",
  "endDate": "<This is the end date of the date range for the created date. The_
↳date format must be mm/dd/yyyy HH:MM:SS. If not provided, the system assumes the_
↳current date as the end date.>",
  "searchText": "<This is the text value of entity instances to be searched in_
↳the searchable fields of entity type.>"
}
```

Example Request:

- Scenario 1:

To search for all the entity instances for entity type Id: 4 and version: 1:

```
{
  "entityTypeId": 4,
  "entityTypeVersion": 1,
  "searchText": "",
  "startDate": "",
  "endDate": ""
}
```

- Scenario 2:

To search for all the entity instances of entity Id: 4 and version: 1 and which has the searchable text matching “mac” and were created on 9-Dec-2013 or later:

```
{
  "entityTypeId": 4,
```



```
"entityTypeVersion": 1,  
"searchText": "mac",  
"startDate": "12/09/2013 00:00:00",  
"endDate": ""  
}
```

Example Response:

```
{  
  "result": [  
    {  
      "entityVersion": 1,  
      "instanceObject": {  
        "Total Experience": "2.33",  
        "Taxable": "true",  
        "Employee Id": "23",  
        "Employee Name": "McDonald",  
        "Gender": "Male",  
        "Hire Date": "01-12-2013 01:01:01",  
        "Department Id": "1",  
        "Department Name": "Admin"  
      },  
      "entityInstanceVersion": 1,  
      "latestVersion": false,  
      "entityInstanceId": 15,  
      "createdBy": "john",  
      "createdTime": "12/09/2013 03:29:17",  
      "modifiedBy": "john",  
      "modifiedTime": "12/09/2013 05:35:34",  
      "entityId": 4,  
      "newVersionRequired": false,  
      "entityBusinessName": "Employee"  
    },  
    {  
      "entityVersion": 1,  
      "instanceObject": {  
        "Total Experience": "3.23",  
        "Taxable": "true",  
        "Employee Id": "4",  
        "Employee Name": "Mac",  
        "Gender": "Male",  
        "Hire Date": "01-12-2013 01:01:01",  
        "Department Id": "1",  
        "Department Name": "Admin"  
      },  
      "entityInstanceVersion": 1,  
      "latestVersion": false,  
      "entityInstanceId": 6,  
      "createdBy": "john",  
      "createdTime": "12/08/2013 22:14:43",  
      "modifiedBy": "john",  
      "modifiedTime": "12/08/2013 22:14:43",  
      "entityId": 4,  
      "newVersionRequired": false,  
      "entityBusinessName": "Employee"  
    },  
    {  
      "entityVersion": 1,  

```

```
{
  "instanceObject": {
    "Total Experience": "3.23",
    "Taxable": "true",
    "Employee Id": "2",
    "Employee Name": "Smith",
    "Gender": "Male",
    "Hire Date": "01-12-2013 01:01:01",
    "Department Id": "1",
    "Department Name": "Admin"
  },
  "entityInstanceVersion": 1,
  "latestVersion": true,
  "entityInstanceId": 4,
  "createdBy": "john",
  "createdTime": "12/08/2013 10:20:10",
  "modifiedBy": "john",
  "modifiedTime": "12/08/2013 10:20:10",
  "entityId": 4,
  "newVersionRequired": false,
  "entityBusinessName": "Employee"
},
  "status": {
    "responseCode": 200,
    "result": null,
    "responseMessage": "Success",
    "responseType": "info"
  }
}
```

Response in Case of Failure:

If no entity instances are found matching the search criteria provided in the requested JSON:

```
{
  "result": null,
  "status": {
    "responseCode": 204,
    "result": null,
    "responseMessage": "No Entity Instance details found",
    "responseType": "info"
  }
}
```

1.6.12.6 Export entity instances

This API allows you to export the details of an entity instance.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/publish/
↪exportEntityInstances/{entity_Id}/{entity_version}
```

Note: You must pass the *entity_Id* and *entity_version* parameters in the API URL.

Method:

GET

Example Response:

Depending on the *REST* client used, the response can be a direct *CSV* file download link or the *CSV* file content.

The following is an example response when the *CSV* file content is displayed:

```
Id,Name
5,alex
```

Note: If entity version is provided as *0*, the system defaults to the latest version of the entity type.

Response in Case of Failure:

- Scenario 1:

If the entity type does not exist for the given entity type Id and version:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 404,
    "responseMessage": "Entity type does not exist for entity id :193
  },
  "result": "No matching entity found."
},
"result": null
}
```

- Scenario 2:

If the no instance exists for the entity type Id and Version:

```
{
  "result": null,
  "status": {
    "responseCode": 500,
    "responseMessage": "No entity instances exist for entity type id_
:13 and version :1",
    "responseType": "ERROR"
  }
}
```

1.6.12.7 Import entity instances

This API allows you to import the details of an entity instance.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/publish/
uploadEntityInstanceCSV
```

Method:

POST

Request Payload:

```
{
    *"filePath": "<This is the full file path of the CSV of entity instances to_
    ↳be imported.>",
    *"errorReportDir": "<This is the directory where the error report log is_
    ↳generated if there is any issue in the imported CSV.>",
    *"entityId": "<This is the entity Id for which the instances are being_
    ↳imported.>",
    *"entityVersion": "<This is the version of the entity for which the instances_
    ↳are being imported.>"
}
```

Example Request:

```
{
    "filePath": "/home/brdev/Entity_Instances_For_Entity_Type_1_and_version_1.csv
    ↳",
    "errorReportDir": "/home/brdev/errorReports",
    "entityId": "1",
    "entityVersion": "1"
}
```

Example Response:

- Scenario 1:

If the entity instances are imported successfully:

```
{
    "result": "100 out of 100 entity instances saved successfully",
    "status": {
        "responseCode": 200,
        "result": null,
        "responseMessage": "100 out of 100 entity instances saved_
    ↳successfully",
        "responseType": "info"
    }
}
```

- Scenario 2:

If there is any error validating entity instances defined in the CSV:

```
{
    "result": "Error_Report_42_1_20131212015322.log",
    "status": {
        "responseCode": 406,
        "result": null,
        "responseMessage": "Some of the entity instances could not be_
    ↳imported due to validation error. Details are available in the error report_
    ↳file.",
        "responseType": "info"
    }
}
```

Response in Case of Failure:

- Scenario 1:

If the CSV file does not contain any instance data for the fields:

```
{
  "status": {
    "responseType": "ERROR",
    "responseCode": 400,
    "responseMessage": "com.zaloni.bedrock.metadata.exception.
↳MetadataException: Empty csv file uploaded",
    "result": null
  },
  "result": "com.zaloni.bedrock.metadata.exception.MetadataException: Empty_
↳csv file uploaded"
}
```

- Scenario 2:

If the CSV file is defined with header not matching the field business names of the entity type:

```
{
  "status": {
    "responseType": "ERROR",
    "responseCode": 400,
    "responseMessage": "com.zaloni.bedrock.metadata.exception.
↳MetadataException: Invalid header for uploaded file",
    "result": null
  },
  "result": "com.zaloni.bedrock.metadata.exception.MetadataException:
↳Invalid header for uploaded file"
}
```

1.6.13 Update Sensitivity of an Entity Field

This API allows you to update the value of the *Sensitivity* attribute of a field across all or specific version of a *DELIMITED* and *FIXED LENGTH* entity; as well as the entities that belong to the *BEDROCK* data store.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/entities/sensitivity?projectIds=
↳{project_Id}
```

Note: You must pass the *project_Id* parameter in the API URL. To fetch all the workflows under the public project, set the *project_Id* parameter to 1.

Method:

```
PUT
```

Request Payload:

```
{
  * "entityId": "<This is the Id of the entity type. The data store of the entity can_
↳be HDFS or BEDROCK.>",
  "entityName": "<This is the technical name of the entity type. This is not_
↳required if entityId is provided.>",
```

```
"sourceSchema": "<This is the source schema attribute of the entity type. This is_
↪not required if entityId is provided.>",
  *"entityFieldName": "<This is the technical name of the field.>",
  "versionList": "<This is the revision Id(s) of the entity whose field needs to be_
↪updated with the specific sensitivity value.>",
  "sourcePlatform": "<This is the source platform attribute of the entity type. This_
↪is not required if entityId is provided.>",
  *"sensitivityValue": "<This is the sensitivity value.>"
}
```

Example Request:

```
{
  "entityId": 367,
  "entityName": "BN_1465370468783",
  "sourceSchema": "sop",
  "sourcePlatform": "sos",
  "entityFieldName": "id",
  "versionList": [1, 2],
  "sensitivityValue": "HIGH"
}
```

Example Response:

```
{
  "responseMessage": "Success",
  "restUri": null,
  "result": "Sensitivity value updated successfully for entity type",
  "page": null
}
```

Response Incase of Failure:

If the field name is not provided:

```
{
  "responseMessage": "Bad Request :: field name should be provided to update_
↪sensitivity value",
  "restUri": "/bedrock-app/services/rest/entities/sensitivity",
  "result": null,
  "page": null
}
```

1.6.14 Bulk Import of Entities, Ingestion Processes, and Post Ingestion Workflows

The ZDP platform allows bulk importing of new entities, ingestion, and post-ingestion workflows when adding new datasets to a Hadoop data lake.

By using this API, you can perform the following operations:

- Define entities.
- Create/configure landing zones.
- Associate post-ingestion workflows.
- Specify the masking and tokenization patterns.

Before using the API, data files containing the details must be created. These are two data files and an optional file for tokenization and masking.

Note:

- The bulk import feature creates file patterns or entities under the public project.
- ZDP does not create new workflows but associates the existing workflows with the newly created file patterns.
- For more information on the bulk import feature, refer to the [Bulk Importing of Metadata by using REST API](#) section.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/upload/populateMetadata
```

Method:

```
POST
```

Request Payload:

```
{
  * "metadataDriverFilepath": "<This is the absolute path to the driver file. This_
  ↳ should be a local path with filename. The driver file contains the data about the_
  ↳ meta files and file patterns.>",
  * "metaFilepath": "<This is the directory to the meta file. This should be a local_
  ↳ path. The meta files contain the data of the fields.>",
  "piiFieldInfoFilePath": "<This is the absolute path to the PII file with_
  ↳ filename. This is optional. This should be a local path. The PII file contains_
  ↳ metadata related to the tokenization and masking options.>",
  * "lzServerIP": "<This is the IP address of the landing server. This should be_
  ↳ already added in ZDP.>"
}
```

Example Request:

```
{
  "metadataDriverFilepath": "/home/bedrock/BEDROCK_DEV/Test_Driver_file.csv",
  "metaFilepath": "/home/bedrock/BEDROCK_DEV",
  "lzServerIP": "192.168.1.247",
  "piiFieldInfoFilePath": "/home/bedrock/BEDROCK_DEV/DUMMY_PII.csv"
}
```

Example Response:

```
{
  "result": "Bulk import successfully completed",
  "status": {
    "responseCode": 200,
    "result": null,
    "responseMessage": "Success",
    "responseType": "info"
  }
}
```

Response in Case of Failure:

- Scenario 1:

If the driver file does not exist:

```
{
  "status": {
    "responseType": "ERROR",
    "responseCode": 500,
    "responseMessage": "Driver file does not exist ",
    "result": null
  },
  "result": null
}
```

- Scenario 2:

If the driver file is not in the CSV format:

```
{
  "status": {
    "responseType": "ERROR",
    "responseCode": 500,
    "responseMessage": "Driver file must be a .CSV file ",
    "result": null
  },
  "result": null
}
```

- Scenario 3:

If a meta file extension is not *.meta*:

```
{
  "status": {
    "responseType": "ERROR",
    "responseCode": 406,
    "responseMessage": "Bulk import completed with errors importing_
↪data for following rows of driver file :: Driver file Row-1 --> Meta  file must_
↪be of .meta type  ; Driver file Row-2 --> Meta  file must be of .meta type  ; ",
    "result": null
  },
  "result": null
}
```

- Scenario 4:

If meta file location is not reachable:

```
{
  "status": {
    "responseType": "ERROR",
    "responseCode": 406,
    "responseMessage": "Bulk import completed with errors importing_
↪data for following rows of driver file :: Driver file Row-1 --> Meta file does_
↪not exist  ; Driver file Row-2 --> Meta file does not exist  ; ",
    "result": null
  },
  "result": null
}
```


- Scenario 5:

If other errors are encountered inside the driver file:

```
{
  "result": "Bulk import completed with errors importing data for following
  ↳rows of driver file ::<errors>",
  "status": {
    "responseCode": 200,
    "result": null,
    "responseMessage": "Success",
    "responseType": "info"
  }
}
```

EXAMPLES:

- Content of an example driver file for Delimited files:

```
Meta File Name,Data File Format,Integrate With HCatalog,Table Delimiter,Target
↳Schema,External Table,External Data Path,Stored As,Table Properties,Change Data
↳Capture Field,Change Data Capture Inactive Table,File System,File Pattern
↳Prefix,File Pattern Suffix,Trigger file Suffix,Delimiter,Workflow Name,Global
↳Parameter,Workflow Level Parameter,Workflow Namespace Parameter,Additional
↳Files extensions,Destination Path,Landing Zone Directory,Notification
↳Distribution List,Script Location,Script TimeOut,Queue Name,ByPass Ingestion,
↳RuleSet Name,Partition File Name,Control File Suffix,Meta File Suffix
20130507T192550_81_HSR_HSR_MBR_DIAG_FULLL.meta,DELIMITED,TRUE,|,testSchema,TRUE,/
↳hdfs/test/path,TextFile,"key1:value1,key2:value2",FIELD_FROM_META_FILE,TRUE,
↳LOCAL,BULK_[0-9][0-9],.dat,.done,|,WF_NAME,"WF_NAME,,,,,,","wf_level_key1:wf_
↳level_value1,wf_level_key2:wf_level_value2","wf_name_space_key2:value,wf_name_
↳space_key3:value3",".ddl",/hdfs/test/DestBulk,/bulk/demo/lzdir,Notification
↳demo_list,/bulk/pre_ingestion/script.sh,1000,QUEUE_NAME,FALSE,DEMO_RULE_SET,
↳basePartition.csv,.ctl,.meta
```

- Content of an example driver file for Fixed Length files:

```
Meta File Name,Data File Format,Integrate With HCatalog,Table Delimiter,Target
↳Schema,External Table,External Data Path,Stored As,Table Properties,Change Data
↳Capture Field,Change Data Capture Inactive Table,File System,File Pattern
↳Prefix,File Pattern Suffix,Trigger file Suffix,Delimiter,Workflow Name,Global
↳Parameter,Workflow Level Parameter,Workflow Namespace Parameter,Additional
↳Files extensions,Destination Path,Landing Zone Directory,Notification
↳Distribution List,Script Location,Script TimeOut,Queue Name,ByPass Ingestion,
↳RuleSet Name,Partition File Name,Control File Suffix,Meta File Suffix
20130507T192550_81_HSR_HSR_MBR_DIAG_FULLL.meta,FIXED LENGTH,TRUE,|,testSchema,TRUE,
↳/hdfs/test/path,TextFile,"key1:value1,key2:value2",FIELD_FROM_META_FILE,TRUE,
↳LOCAL,BULK_[0-9][0-9],.dat,.done,|,WF_NAME,"WF_NAME,,,,,,","wf_level_key1:wf_
↳level_value1,wf_level_key2:wf_level_value2","wf_name_space_key2:value,wf_name_
↳space_key3:value3",".ddl",/hdfs/test/DestBulk,/bulk/demo/lzdir,Notification
↳demo_list,/bulk/pre_ingestion/script.sh,1000,QUEUE_NAME,FALSE,DEMO_RULE_SET,
↳basePartition.csv,.ctl,.meta
```

- Content of an example Meta file:

```
Source_Name|Schema_Name|TABLE_NAME|COLUMN_NAME|DATA_TYPE|DATA_LENGTH|DATA_
↳SCALE|Format|Primary_Keys|COLUMN_ID|SENSITIVITY
HSR|HSR|UR_JURDC|UR_JURDC_CD|VARCHAR2|2|||Y|1|PII
HSR|HSR|UR_JURDC|CREAT_USER_ID|VARCHAR2|20|||N|3
```

```

HSR|HSR|UR_JURDC|CHG_DTTM|TIMESTAMP(3)|11||yyyy-MM-dd HH:mm:ss.SSSSSS|N|4
HSR|HSR|UR_JURDC|CHG_USER_ID|VARCHAR2(20)|||N|5
HSR|HSR|UR_JURDC|UPDT_VER_NBR|NUMBER(10|0)||N|6
HSR|HSR|UR_JURDC|SOS_APPL_IND|NUMBER(1|0)||N|7
HSR|HSR|UR_JURDC|SOR_APPL_IND|NUMBER(1|0)||N|8
HSR|HSR|UR_JURDC|SOI_APPL_IND|NUMBER(1|0)||N|9
HSR|HSR|UR_JURDC|XCPT_IND|NUMBER(1|0)||N|10

```

- Contents of an example PII file:

```

Name,Visible,Mask,Masking Pattern,Full Token,Partial Token,Tokenization Algorithm_
↪Name
SSN,Y,Y,1-3:##;5-8:$;10-12:##;,Y,Y,NA
DOB,N,Y,1-3:##;5-8:$;10-12:##;,Y,Y,SHA1
MBR_ID,N,Y,1-3:##;5-8:$;10-12:##;,Y,Y,SHA1
MBR_LAST_NM,N,Y,1-3:##;5-8:$;10-12:##;,N,N,NA
MBR_ADDR,N,Y,1-3:##;5-8:$;10-12:##;,N,N,NA
STATE_CD,N,Y,1-3:##;5-8:$;10-12:##;,N,N,NA
ST_CD,N,Y,1-3:##;5-8:$;10-12:##;,N,N,NA
MBR_ZIP,N,Y,1-3:##;5-8:$;10-12:##;,Y,N,SHA1
MBR_Phone_Nbr,N,Y,1-3:##;5-8:$;10-12:##;,Y,N,SHA1
DIAG_CD,N,Y,1-3:##;5-8:$;10-12:##;,N,N,NA
DIAG_PRIM_CD,N,Y,1-3:##;5-8:$;10-12:##;,N,N,NA
PROC_CD,N,Y,1-3:##;5-8:$;10-12:##;,N,N,NA

```

1.7 Data Quality API

The following APIs allow you to perform various operations and across the quality module in ZDP. Each API in the subsequent sections serves a specific purpose to fulfill in setting up the quality conditions (such as rules, rule sets).

1.7.1 Add/Update Simple or Compound Rule (JSON Format)

This API allows you to add or update a simple or compound rule. All the details of a rule can be updated except the rule name. If a rule is not associated with any field, all the details of a rule can be updated, except the rule name.

However, if a rule is associated with a field, for updating the validation function, the new function's data type must be one of the following:

- Same as the earlier data type
- Unknown data type

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dq/rules/addOrUpdate
```

Method:

```
POST
```

Request Payload:

```

{
  "ruleName": "<This is the name of the rule.>",
  "ruleTag": "<This is the rule tag.>",

```

```

"ruleDesc": "<This is the rule description.>",
* "functionName": "<This is the function name.>",
* "ruleExpression": "<This the whole rule expression.>",
"failureMessage": "<This is the error message to return when this rule fails.
↪>",
"failureErrorCode": "<This is the error code to return when this rule fails.>
↪",
"effectiveDate": "<This is the date from which the rule is effective.>",
"terminationDate": "<This is the date till which the rule is effective.>",
"ruleStatus": "<This is the status of the rule.>",
"invertFunction": "<If set to 1, invert-validation is performed. If set to 0,
↪ regular validation is performed.>",
"dateFormat": "<This is the format of the date. This parameter is mandatory.
↪when DATE function is selected.>",
* "Variables": "<This are the variables that are used in the expression.>",
* "ruleExpression": "<{
                                This is the whole expression. If two
↪expressions must be defined, use the desired conditional operator (AND/OR) to set
↪the rule condition (refer to the Example Request 2 below)
                                }>",
* "ruleExpressionJson": "<This is the whole expression JSON. For simple rule,
↪the default conditional operator is AND (refer to the Example Request 1 below).>"
}

```

Example Request:

- Scenario 1:

To add a simple rule. If this rule already exists, update parameters as per the requirement.

```

{
  "ruleName": "SimpleRule_API",
  "ruleDesc": "",
  "failureMessage": "",
  "failureErrorCode": "",
  "ruleTag": "",
  "effectiveDate": "",
  "terminationDate": "",
  "dateFormat": "",
  "ruleStatus": "ACTIVE",
  "invertFunction": 0,
  "parameterValue": null,
  "Variables": [
    {
      "name": "primaryVar",
      "type": "STRING",
      "isPrimary": true
    }
  ],
  "ruleExpression": "{EQUALS_STRING(primaryVar,\"ZPL\")}",
  "ruleExpressionJson": {
    "group": {
      "operator": "AND",
      "rules": [
        {
          "data": "EQUALS_STRING(primaryVar,\"ZPL\")"
        }
      ],
      "firstGroup": true
    }
  }
}

```

```

    }
  }
}

```

- Scenario 2:

To add a compound rule. If this rule already exists, update parameters as per the requirement.

```

{
  "ruleName": "CompundRule_API",
  "ruleDesc": " Compound Rule using API",
  "failureMessage": "Error from- CompundRule_API Rule ",
  "failureErrorCode": "13",
  "ruleTag": "RestAPI",
  "effectiveDate": "2015-12-01 00:00:00",
  "terminationDate": "2015-12-31 00:00:00",
  "dateFormat": "",
  "ruleStatus": "ACTIVE",
  "invertFunction": 0,
  "parameterValue": null,
  "Variables": [
    {
      "name": "primaryVar",
      "type": "NUMBER",
      "isPrimary": true
    },
    {
      "name": "secondaryVar",
      "type": "NUMBER",
      "isPrimary": false
    }
  ],
  "ruleExpression": "{EQUALS_NUMBER(primaryVar,10) OR GREATER_THAN_
↪NUMBER(secondaryVar,20)}",
  "ruleExpressionJson": {
    "group": {
      "operator": "OR",
      "rules": [
        {
          "data": "EQUALS_NUMBER(primaryVar,10) "
        },
        {
          "data": "GREATER_THAN_NUMBER(secondaryVar,
↪20) "
        }
      ],
      "firstGroup": true
    }
  }
}

```

Example Response:

```

{
  "result": null,
  "status": {
    "responseCode": 200,
    "result": null,

```

```

    "responseMessage": "Rule added successfully: CompundRule_API",
    "responseType": "info"
  }
}

```

Once a rule is defined, it can be associated with a rule set.

Rule and Rule Set Association

- A rule set is defined as a set of rules with the added constraint that all the rules in a rule set must operate on the same data type for a field.
- A rule can be part of multiple rule sets.
- A rule set name is not case sensitive and is unique.
- A rule with a validation function that has the *UNKNOWN* field data type can be associated with any rule set.
- However, the reverse is not true, i.e a rule set with field data type as *UNKNOWN* can only be associated with a rule containing a validation function that has the *UNKNOWN* field data type.

Execution Order:

When there are multiple rules grouped into a rule set and that rule set is associated with a metadata field, internally all those rules get associated with that field. When there are multiple rules associated a field, you can define the order in which the rule gets executed. This execution order value is numeric.

Stop Execution:

When there are multiple rules associated a field, the rules are executed in the order you have defined through the *Execution Order* attribute. In this process, you can set the *stop execution* flag against each rule. If this is set to *True* for a rule and if that rule fails while execution, the remaining rules do not get executed. For example, SSN has three rules: rule 1 (order 1), rule 2 (order 2), and rule 3 (order 3). Here, if the stop execution flag is set to *True* for the rule 2, and while execution, if this rule fails, the last rule: rule 3 does not get executed. However, if the flag is set to *False*, even if rule 2 fails, the last rule “rule 3” is executed.

1.7.2 Add Rule Set (JSON Format)

This API allows you to create rule set and to associate rule with that rule set.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dq/rules/addruleSet
```

Method:

POST

Request Payload:

```

{
  *"ruleSetName": "<This is the rule set name.>",
    "ruleSetDesc": "<This is the rule set description.>",
  *"dataType": "<This is the rule set data type.>",
  *"dqRuleSetRuleMappingList": [
    {
      "ruleName": "<This is the rule name.>",
      "executionOrder": "<This is the rule execution order.>",
      "stopExecution": "<This is the stop execution flag.>"
    },
    {

```

```
{
  {
    "ruleName": "<This is the rule name.>",
    "executionOrder": "<This is the rule execution order.>",
    "stopExecution": "<This is the stop execution flag.>"
  }
}
```

Example Request:

```
{
  "ruleSetName": "SSN_CHECKS",
  "ruleSetDesc": "Used to validate the SSN",
  "dataType": "STRING",
  "dqRuleSetRuleMappingList": [
    {
      "ruleName": "SSN_Pattern_Check_Rule1",
      "executionOrder": 1,
      "stopExecution": false
    },
    {
      "ruleName": "SSN_Pattern_Check_Rule2",
      "executionOrder": 2,
      "stopExecution": true
    }
  ]
}
```

Example Response:

```
{
  "result": null,
  "status": {
    "responseCode": 200,
    "result": null,
    "responseMessage": "rule set saved successfully with name: SSN_CHECKS",
    "responseType": "info"
  }
}
```

1.7.3 Update rule set (JSON Format)

This API allows you to update a rule set, add or remove new rules, or change the *execution order* and *stop execution* flags. You cannot update the rule set name.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dq/rules/updateruleSet
```

Method:

```
POST
```

Request Payload:

```
{
  *"ruleSetName": "<This is the rule set name.>",
    "ruleSetDesc": "<This is the rule set description.>",
  *"dataType": "<This is the rule set data type.>",
  *"dgruleSetruleMappingList": [
    {
      "ruleName": "<This is the rule name.>",
      "executionOrder": "<This is the rule execution order.>",
      "stopExecution": "<This is the stop execution flag.>"
    },
    {
      "ruleName": "<This is the rule name.>",
      "executionOrder": "<This is the rule execution order.>",
      "stopExecution": "<This is the stop execution flag.>"
    }
  ]
}
```

Example Request:

```
{
  "ruleSetName": "SSN_CHECKS",
  "ruleSetDesc": "Used to validate the SSN",
  "dataType": "STRING",
  "dgruleSetruleMappingList": [
    {
      "ruleName": "SSN_Pattern_Check_Rule1",
      "executionOrder": 1,
      "stopExecution": false
    },
    {
      "ruleName": "SSN_Pattern_Check_Rule2",
      "executionOrder": 2,
      "stopExecution": true
    }
  ]
}
```

Example Response:

```
{
  "result": null,
  "status": {
    "responseCode": 200,
    "result": null,
    "responseMessage": "Ruleset updated successfully with name: SSN_CHECKS",
    "responseType": "info"
  }
}
```

Associate rule and rule set with metadata field

Once rule and rule set are defined, they can be associated with the metadata field.

Rule and Entity Field Association - One field can be associated with many rules. - One rule can be applicable to many fields. - A rule name is not case sensitive and is unique. - A rule with a validation function that has the *UNKNOWN* field data type can be associated with any field. - However, the reverse is not true. Hence, a field with

data type as *UNKNOWN* can only be associated with a rule containing a validation function that has the *UNKNOWN* field data type.

When a rule set is associated with the metadata field, all the rules belonging to that rule set are automatically associated with the field.

1.7.4 View Rules (JSON Format)

This API allows you to view rule details that are filtered by rule and rule set details.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dq/rules/viewRuleDetails
```

Method:

```
POST
```

Request Payload:

Use any one of the below parameters to view the results:

```
{
  "ruleSetName": "<This is the name of the rule set.>",
  "ruleName": "<This is the name of the rule.>",
  "ruleTag": "<This is the name of the rule tag.>",
  "ruleStatus": "<This is the status of the rule.>"
}
```

Example Request:

```
{
  "ruleSetName": "Standard Entity DQ Ruleset",
  "ruleName": "",
  "ruleTag": "",
  "ruleStatus": ""
}
```

Example Response:

```
{
  "result": [
    {
      "ruleCount": 1,
      "ruleDefinitions": [
        {
          "ruleName": "test1",
          "ruleTag": "",
          "ruleDesc": "",
          "functionName": null,
          "functionDataType": null,
          "ruleExpression": "{EQUALS_STRING(primaryVar,\n↪\"test\")}",
          "failureMessage": "Error in rule: test1",
          "failureErrorCode": "1",
          "effectiveDate": null,
          "terminationDate": null,
          "ruleStatus": "ACTIVE",
        }
      ]
    }
  ]
}
```



```

        "invertFunction": 0,
        "createdBy": "admin",
        "modifiedBy": "admin",
        "createdDate": "2016-08-09 16:24:50",
        "modifiedDate": "2016-08-09 16:24:50",
        "dateFormat": "",
        "ruleId": 7
      }
    ]
  },
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Success",
    "result": null
  }
}

```

1.7.5 View Rule by Entity (JSON Format)

This API allows you to view rules that are filtered by entity details (by using source platform, source schema, entity, field names information).

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dq/rules/viewRuleByEntity
```

Method:

```
POST
```

Request Payload:

Use any one of the following parameters to view the associated entities:

```

{
  "fieldName": "<This is the name of the entity field. To fetch all the rules,
  ↳ associated with the fields matching the fieldName, across ZDP metadata, set only.
  ↳ this parameter in the payload. Alternately, use this parameter as combination.>",
  "entityName": "<This is the business name of the entity.>",
  "sourceSchema": "<This is the name of the source schema.>",
  "sourcePlatform": "<This is the name of the source platform.>"
}

```

Example Request:

```

{
  "fieldName": "F1",
  "entityName": "Rest API Entity Type",
  "sourceSchema": null,
  "sourcePlatform": null
}

```

Example Response:

```

{
  "result": {
    "chunkSize": 0,
    "currentPage": 0,
    "resultList": [
      {
        "fieldName": "F1",
        "entityName": "Rest API Entity Type",
        "sourceSchema": "demo",
        "sourcePlatform": "demo",
        "ruleCount": 1,
        "ruleDefinitions": [
          {
            "ruleName": "NewRuleAPI2",
            "ruleTag": null,
            "ruleDesc": null,
            "functionName": null,
            "functionDataType": null,
            "ruleExpression": "{EQUALS_
↪NUMBER(primaryVar,65)}",
            "failureMessage": "Error in rule: ↪
↪NewRuleAPI2",
            "failureErrorCode": "1",
            "effectiveDate": null,
            "terminationDate": null,
            "ruleStatus": "ACTIVE",
            "invertFunction": 1,
            "createdBy": "admin",
            "modifiedBy": "admin",
            "createdDate": "2015-12-16 19:19:07",
            "modifiedDate": "2015-12-16 19:27:31",
            "dateFormat": ""
          }
        ]
      }
    ],
    "totalRecords": 0
  },
  "status": {
    "responseType": "info",
    "responseMessage": "Success",
    "result": null,
    "responseCode": 200
  }
}

```

1.7.6 Delete Rule (JSON Format)

This API allows you to delete the rule by using the rule name. When a rule is deleted, associations with any rule sets and fields are also deleted.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dq/rules/deleteRule
```

Method:

POST

Request Payload:

```
{
  "ruleNameList": [
    "<Provide the rule names.>"
  ]
}
```

Example Request:

```
{
  "ruleNameList": [
    "SSN_Pattern_Check_Rule1",
    "SSN_Pattern_Check_Rule2"
  ]
}
```

Example Response:

```
{
  "result": null,
  "status": {
    "responseCode": 200,
    "result": null,
    "responseMessage": "Rule: SSN_Pattern_Check_Rule1, SSN_Pattern_Check_
↪Rule2 deleted successfully",
    "responseType": "info"
  }
}
```

1.7.7 Delete Rule Set (JSON Format)

This API allows you to delete the rule set(s) by using the rule set name. When a rule set is deleted, associations with any fields are also deleted.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dq/rules/deleteRuleSet
```

Method:

POST

Request Payload:

```
{
  "ruleSetNameList": [
    "DATE_CHECKS",
    "SSN_CHECKS"
  ]
}
```

Example Request:

```
{
  "ruleSetNameList": [
    "qwertyuio",
    "qwewertyuio"
  ]
}
```

Example Response:

```
{
  "result": null,
  "status": {
    "responseType": "info",
    "responseMessage": "Rule set: qwertyuio deleted successfully. Rule_
↪set: qwewertyuio not exists",
    "result": null,
    "responseCode": 200
  }
}
```

1.7.8 Add Rules (CSV Format)

This API allows you to add bulk rules by using the CSV format. If a rule name is already available in the database, the new rule is not added.

URL:

```
http://<bedrock-host>:<port>/bedrock-app/services/rest/dq/rules/csv/add
```

Method:

```
POST
```

Example CSV:**Example Response:**

```
{
  "result": null,
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "File '<csv_filename>.csv' has been processed and_
↪uploaded successfully.",
    "result": null
  }
}
```

1.7.9 Add or Update rules (CSV Format)

This API allows you to add/update bulk rules by using the CSV format. If any existing rule is available in the database, it is updated based on the rule name. However, the rule name cannot be updated.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dq/rules/csv/addOrUpdate
```

Method:

```
POST
```

Input Parameters: For this REST call, the parameter type must be *form-data* and format must be *file* with the following key-values:

Key	Value
<i>file</i>	Choose the .csv file storing the rule definitions file from the local system.

Example CSV:**Example Response:**

```
{
  "result": null,
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "File '<csv_filename>.csv' has been processed and
    ↪uploaded successfully.",
    "result": null
  }
}
```

1.7.10 Fetch Field/File Level DQ Aggregations

This API allows you to fetch the valid and invalid counts performed by file and field level data quality rules in ZDP using various filters called as *bucketType*. Each aggregate in the response is a collection of total, invalid, and valid counts headed by the queried *bucketType*.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/dq/reports/aggregates
```

Method:

```
POST
```

Request Payload:

```
{
  *"searchType": "<This is the type of search: field level or entity level. Probable
  ↪values can be FIELD_DQ or ENTITY_DQ.>",
  *"bucketType": "<This indicates the type of the desired bucket. Probable values
  ↪can be DATE, RULE, RULESET, FIELD, ENTITY_ID_VERSION, ENTITY_NAME_VERSION, or FILE.
  ↪For example, set this value to DATE to fetch the date histogram.>"
  "intervalType": "<This is an optional property and applies only when when
  ↪bucketType is set as DATE. The histogram buckets are fetched based on the set
  ↪interval type. The probable values can be DAY, MONTH, or YEAR. The default value is
  ↪DAY.>",
  "entityList": [ "<This is the list of entities for which the aggregates are
  ↪retrieved. An empty list indicates all the entities. To set multiple entities, use
  ↪the format:
```

```

        "<EntityId_1.versionId>",
        "<EntityId_2.versionId>"
    >"],
    "dateRange": {
        "<This is an optional block. Set the start and end_
    ↪ dates to fetch aggregates within the specified interval, in the following format:
        "fromDateTime": "<Use the format: <yyyy-mm-dd>T
    ↪<hh:mm:ss:sss>+<time_zone>.>",
        "toDateTime": "<Use the format: <yyyy-mm-dd>T
    ↪<hh:mm:ss:sss>+<time_zone>.>"
    >}
}

```

Example Request:

```

{
    "entityList": [],
    "searchType": "ENTITY_DQ",
    "bucketType": "ENTITY_NAME_VERSION",
    "intervalType": "YEAR",
    "dateRange": {
        "fromDateTime": "2016-04-01T00:00:00.000+05:30",
        "toDateTime": "2016-05-13T23:00:00.000+05:30"
    }
}

```

Example Response:

- Scenario 1:

If *searchType* is *ENTITY_DQ* and *bucketType* is *ENTITY_NAME_VERSION*:

```

{
    "responseMessage": "Aggregations fetched successfully",
    "restUri": null,
    "result": {
        "records": [
            {
                "label": "Chaw.1",
                "totalCount": 1,
                "invalidCount": 0,
                "validCount": 1
            },
            {
                "label": "Fanwebglobal RAW.1",
                "totalCount": 2,
                "invalidCount": 0,
                "validCount": 2
            },
            {
                "label": "TestEntity.1",
                "totalCount": 6,
                "invalidCount": 0,
                "validCount": 6
            }
        ],
        "totalCount": 9,
        "totalInvalidCount": 0,
        "totalValidCount": 9
    }
}

```

```
    },  
    "page": null  
}
```

- Scenario 2:

If *searchType* is *ENTITY_DQ* and *bucketType* is *RULE*:

```
{  
  "responseMessage": "Aggregations fetched successfully",  
  "restUri": null,  
  "result": {  
    "records": [  
      {  
        "label": "RECORD COUNT CHECK",  
        "totalCount": 5,  
        "invalidCount": 0,  
        "validCount": 5  
      },  
      {  
        "label": "SCHEMA VALIDATION",  
        "totalCount": 4,  
        "invalidCount": 0,  
        "validCount": 4  
      }  
    ],  
    "totalCount": 9,  
    "totalInvalidCount": 0,  
    "totalValidCount": 9  
  },  
  "page": null  
}
```

- Scenario 3:

If no buckets are available:

```
{  
  "responseMessage": "No buckets found",  
  "restUri": null,  
  "result": {  
    "records": [],  
    "totalCount": 0,  
    "totalInvalidCount": 0,  
    "totalValidCount": 0  
  },  
  "page": null  
}
```

Response in Case of Failure:

If the input is incorrect:

```
{  
  "responseMessage": "<Error message>",  
  "restUri": "/bedrock-app/services/rest/dq/reports/aggregates",  
  "result": null,  
  "page": null  
}
```

1.8 Bedrock Namespace API

These APIs allow you to fetch, save, modify, and delete Bedrock namespaces.

1.8.1 Fetch Bedrock Namespace

This API allows you to fetch the Bedrock namespaces in the *JSON* format.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/admin/namespace/bedrock/  
↪ fetchBedrockNamespace
```

Method:

```
GET
```

Example Response:

```
{  
  "result": [  
    {  
      "value": "value1",  
      "key": "key1",  
      "id": 22,  
      "overridable": false  
    },  
    {  
      "value": "value2",  
      "key": "key2",  
      "id": 23,  
      "overridable": false  
    }  
  ],  
  "status": {  
    "responseCode": 200,  
    "result": null,  
    "responseMessage": "Success",  
    "responseType": "info"  
  }  
}
```

Response in Case of Failure:

If no Bedrock namespace is available:

```
{  
  "status": {  
    "responseType": "SUCCESS",  
    "responseCode": 200,  
    "responseMessage": "No record found!",  
    "result": null  
  },  
  "result": null  
}
```


1.8.2 Save Bedrock Namespace

This API allows you to save the Bedrock namespaces.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/admin/namespace/bedrock/  
↪saveVariables
```

Method:

```
POST
```

Request Payload:

```
{  
  "systemNamespaceList": [  
    {  
      *"key": "bedrock_namespace_key",  
      *"value": "bedrock_namespace_value",  
      "id": 0,  
      "overridable": True or false (if not specified)  
    }  
  ]  
}
```

Example Request:

```
{  
  "systemNamespaceList": [  
    {  
      "key": "path",  
      "value": "/home/bedrock/examplnamespace",  
      "id": 0,  
      "overridable":""  
    }  
  ]  
}
```

Example Response:

```
{  
  "status": {  
    "responseType": "INFO",  
    "responseCode": 200,  
    "responseMessage": "Namespace variable 'SB' has been added_  
↪successfully , ",  
    "result": null  
  },  
  "result": null  
}
```

Response in Case of Failure:

If the Bedrock namespace with the same key exists:

```
{  
  "status": {  
    "responseType": "INFO",
```

```
{
  "responseCode": 412,
  "responseMessage": "Namespace key MyPath already exists , ",
  "result": null
},
"result": null
}
```

1.8.3 Modify Bedrock Namespace

This API allows you to modify the Bedrock namespaces.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/admin/namespace/bedrock/
↪modifyVariables
```

Method:

POST

Example Request:

```
{
  "systemNamespaceList": [
    {
      *"key": "path",
      *"value": "/home/bedrock/datafiles/",
      *"id": 2,
      "overridable": "true"
    }
  ]
}
```

Example Response:

```
{
  "status": {
    "responseType": "INFO",
    "responseCode": 200,
    "responseMessage": "Namespace variable 'SB_1121' has been updated_
↪successfully , ",
    "result": null
  },
  "result": null
}
```

Response in Case of Failure:

```
{
  "result": null,
  "status": {
    "responseCode": 500,
    "result": null,
    "responseMessage": "Could not update bedrock namespace with key
↪'bedrock_namespace_key' ",
    "responseType": "ERROR"
  }
}
```

```
}  
}
```

1.8.4 Delete Bedrock Namespace

This API allows you to delete the Bedrock namespaces.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/admin/namespace/bedrock/  
↪deleteVariables
```

Method:

```
POST
```

Request Payload:

```
{  
  *"systemNamespaceList": [  
    {  
      *"key": "bedrock_namespace_key"  
    }  
  ]  
}
```

Example Response:

```
{  
  "status": {  
    "responseType": "INFO",  
    "responseCode": 200,  
    "responseMessage": "Namespace variable 'bedrock_namespace_key' has_  
↪been deleted successfully , ",  
    "result": null  
  },  
  "result": null  
}
```

Response in Case of Failure:

If Bedrock namespace does not exist:

```
{  
  "status": {  
    "responseType": "INFO",  
    "responseCode": 200,  
    "responseMessage": "Error while deleting bedrock namespace variable  
↪'namespace_212' , ",  
    "result": null  
  },  
  "result": null  
}
```

1.9 Category Namespace API

These APIs allow you to fetch, save, modify, and delete category namespaces.

1.9.1 Fetch Category Namespace

This API allows you to fetch category namespaces in the JSON format.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/namespace/category/  
↪ fetchCategoryNamespace
```

Method:

```
POST
```

Request Payload:

```
{  
  "categoryId": 1  
}
```

Example Response:

```
{  
  "result": [  
    {  
      "value": "categoryvalue",  
      "key": "categorynamespacekey",  
      "id": 3,  
      "overridable": true,  
      "categoryId": 0  
    },  
    {  
      "value": "value2",  
      "key": "key2",  
      "id": 4,  
      "overridable": false,  
      "categoryId": 0  
    }  
  ],  
  "status": {  
    "responseCode": 200,  
    "result": null,  
    "responseMessage": "Success",  
    "responseType": "info"  
  }  
}
```

Response in Case of Failure:

If no category namespace is available:

```
{  
  "result": null,  
  "status": {
```

```

        "responseCode": 200,
        "result": null,
        "responseMessage": " No category namespace found for the category id 1
    ↪",
        "responseType": "SUCCESS"
    }
}

```

1.9.2 Save Category Namespace

This API allows you to save the category namespaces.

URL:

```

http://<bedrock-host>:port/bedrock-app/services/rest/workflow/namespace/category/
    ↪saveVariables

```

Method:

POST

Example Request:

```

{
    "categoryNamespaceList": [
        {
            *"categoryId": "1",
            *"key": "testcatnamespace1",
            *"value": "testvalue",
            "overridable": false
        }
    ]
}

```

Example Response:

```

{
    "result": null,
    "status": {
        "responseType": "INFO",
        "responseMessage": "Category namespace testcatnamespace1 is added_
    ↪successfully for the given category., ",
        "result": null,
        "responseCode": 200
    }
}

```

Response in Case of Failure:

If the category Id does not exist:

```

{
    "status": {
        "responseType": "INFO",
        "responseCode": 200,
        "responseMessage": "No category is available for category namespace_
    ↪SB !, ",
    }
}

```

```
        "result": null
    },
    "result": null
}
```

1.9.3 Modify Category Namespace

This API allows you to modify the category namespaces.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/namespace/category/  
→modifyVariables
```

Method:

```
POST
```

Example Request:

The *id* and *categoryId* parameters are mandatory to identify and modify the specific category namespace. The *key*, *value*, or *overridable* parameters can be updated.

To modify category namespace, perform the following steps:

1. Invoke the *Fetch Category Namespace* API to get the details for the desired category.
2. Modify the category variables in the JSON fetched by the *Fetch Category Namespace* API.

```
{
    "categoryNamespaceList": [
        {
            *"id": 3,
            *"categoryId": 1,
            *"key": "testcatnamespace3",
            *"value": "samplevalue_category3",
            "overridable": false
        }
    ]
}
```

Example Response:

```
{
    "result": null,
    "status": {
        "responseType": "INFO",
        "responseMessage": "Category namespace testcatnamespace3 is modified_
→successfully for the given category., ",
        "result": null,
        "responseCode": 200
    }
}
```

Response in Case of Failure:

```
{
  "result": null,
  "status": {
    "responseCode": 500,
    "result": null,
    "responseMessage": "Could not modify category namespace with key 'testcatnamespace3' ",
    "responseType": "ERROR"
  }
}
```

1.9.4 Delete Category Namespace

This API allows you to delete the category namespaces.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/namespace/category/
deleteVariables
```

Method:

POST

Example Request:

```
{
  "categoryNamespaceList": [
    {
      "id": 1
    }
  ]
}
```

Example Response:

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Category namespace with namespace id 17 deleted successfully. ",
    "result": null
  },
  "result": null
}
```

Response in Case of Failure:

If the category namespace Id is invalid (or does not exist):

```
{
  "status": {
    "responseType": "info",
    "responseCode": 200,
    "responseMessage": "Category namespace with namespace id 171 could not be deleted. ",
  },
}
```

```
        "result": null
    },
    "result": null
}
```

1.10 Project Specific API

A ZDP project can be considered as a logical workspace that can be shared across multiple users (internal or LDAP) or LDAP groups; to perform various operations on the set of project-specific artifacts.

An artifact is any ZDP object that is created, used, and modified by the ZDP users. These are different from operational data, such as workflow history, ingestion history, etc.

Example of artifacts are file patterns, entities, workflows, and transformations.

By using the Role Based Access Control (RBAC) feature, you can control access to the various ZDP artifacts based on projects and user roles. This approach ensures that the project-specific artifacts are available only to authorized users.

This section describes the project-specific services that are available across the ZDP platform based on the set role permissions. By using the project-specific APIs, you can get detailed information as follows:

- Fetch the list of existing projects.
- Manage artifacts across multiple ZDP projects.
- Transfer artifacts from one ZDP project to another.

For more information on projects, refer to the `role_based_access_control` section.

1.10.1 Fetch the List of Allocated Projects

This API allows you to fetch the list of projects allocated to the currently logged-in user.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects
```

Method:

```
GET
```

Example Response:

```
{
  "responseMessage": "SUCCESS",
  "restUri": "/projects",
  "result": [
    {
      "projectId": 34,
      "projectName": "SB_2304_AS_BP",
      "createdBy": "admin",
      "createdDateTime": 1487834629000,
      "canManageUserRoles": true,
      "isDefaultProject": false,
      "roleNames": [
        "Project Administrator"
      ]
    }
  ]
}
```



```

    },
    {
      "projectId": 33,
      "projectName": "SB_2304_AS",
      "createdBy": "admin",
      "createdDateTime": 1487834106000,
      "canManageUserRoles": true,
      "isDefaultProject": false,
      "roleNames": [
        "Project Administrator"
      ]
    },
    {
      "projectId": 32,
      "projectName": "project09",
      "createdBy": "admin",
      "createdDateTime": 1487833408000,
      "canManageUserRoles": true,
      "isDefaultProject": false,
      "roleNames": [
        "Project Administrator"
      ]
    },
    {
      "projectId": 31,
      "projectName": "project06",
      "createdBy": "admin",
      "createdDateTime": 1487833097000,
      "canManageUserRoles": true,
      "isDefaultProject": false,
      "roleNames": [
        "Project Administrator"
      ]
    },
    {
      "projectId": 30,
      "projectName": "project_dpaul1",
      "createdBy": "admin",
      "createdDateTime": 1487831390000,
      "canManageUserRoles": true,
      "isDefaultProject": false,
      "roleNames": [
        "Project Administrator"
      ]
    },
    {
      "projectId": 2,
      "projectName": "project05",
      "createdBy": "admin",
      "createdDateTime": 1487058503000,
      "canManageUserRoles": true,
      "isDefaultProject": false,
      "roleNames": [
        "Project Administrator"
      ]
    },
    {
      "projectId": 1,

```

```
        "projectId": 23,
        "projectName": "SB_2304_BP_457",
        "createdBy": "admin",
        "createdDateTime": 1487753725000,
        "canManageUserRoles": true,
        "isDefaultProject": false,
        "roleNames": [
            "Project Administrator"
        ]
    },
    {
        "projectId": 22,
        "projectName": "SB_2304_BP12345",
        "createdBy": "admin",
        "createdDateTime": 1487753392000,
        "canManageUserRoles": true,
        "isDefaultProject": false,
        "roleNames": [
            "Project Administrator"
        ]
    },
    {
        "projectId": 17,
        "projectName": "SB_2304",
        "createdBy": "admin",
        "createdDateTime": 1487750097000,
        "canManageUserRoles": true,
        "isDefaultProject": false,
        "roleNames": [
            "Project Administrator"
        ]
    },
    {
        "projectId": 3,
        "projectName": "mica_project",
        "createdBy": "admin",
        "createdDateTime": 1487082410000,
        "canManageUserRoles": true,
        "isDefaultProject": false,
        "roleNames": [
            "Project Administrator"
        ]
    }
],
"page": null
}
```

1.10.2 Search Projects

This API allows you to search for projects allocated to the currently logged-in user in ZDP.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/search
```

Method:

```
POST
```

Example Request:

To search for the project(s) that matches a specific string:

```
{
  "page": {
    "chunkSize": 5,
    "currentPage": 1,
    "sortBy": "createdTime",
    "sortOrder": "ASC",
    "totalRecords": 0
  },
  "projectName": "SB_"
}
```

Example Response:

- Scenario 1:

If the project name matches with the searched string:

```
{
  "responseMessage": "SUCCESS",
  "restUri": "/projects/search",
  "result": [
    {
      "id": 16,
      "name": "SB",
      "description": "ZDP project for 2304",
      "createdTime": 1487749808000,
      "createdBy": "admin",
      "modifiedTime": 1487749808000,
      "modifiedBy": "admin",
      "roleAssociations": null
    },
    {
      "id": 17,
      "name": "SB_2304",
      "description": "This is a demo based ZDP project",
      "createdTime": 1487750097000,
      "createdBy": "evan",
      "modifiedTime": 1487750097000,
      "modifiedBy": "evan",
      "roleAssociations": null
    },
    {
      "id": 19,
      "name": "SB_2304_BP",

```

```
        "description": "This is a sample ZDP project",
        "createdTime": 1487752724000,
        "createdBy": "evan",
        "modifiedTime": 1487752724000,
        "modifiedBy": "admin",
        "roleAssociations": null
    },
    {
        "id": 20,
        "name": "SB_2304_BP123",
        "description": null,
        "createdTime": 1487753253000,
        "createdBy": "evan",
        "modifiedTime": 1487753253000,
        "modifiedBy": "evan",
        "roleAssociations": null
    },
    {
        "id": 21,
        "name": "SB_2304_BP1234",
        "description": null,
        "createdTime": 1487753351000,
        "createdBy": "admin",
        "modifiedTime": 1487753351000,
        "modifiedBy": "admin",
        "roleAssociations": null
    }
],
"page": {
    "chunkSize": 5,
    "currentPage": 1,
    "sortBy": "createdTime",
    "sortOrder": "ASC",
    "totalRecords": 9
}
}
```

- Scenario 2:

If no project matches for the searched query:

```
{
    "responseMessage": "SUCCESS",
    "restUri": "/projects/search",
    "result": [],
    "page": {
        "chunkSize": 2,
        "currentPage": 1,
        "sortBy": "createdTime",
        "sortOrder": "ASC",
        "totalRecords": 0
    }
}
```

Response in Case of Failure:

If the value for a parameter is invalid in the request payload:

```
{
  "responseMessage": "org.hibernate.QueryException: could not resolve property:↵
↵createdDate of: com.zaloni.bedrock.security.entity.SecProject",
  "restUri": "/bedrock-app/services/rest/projects/search",
  "result": null,
  "page": null
}
```

1.10.3 Fetch Roles and Users Details

Use this service to fetch the information of all roles and users existing in the ZDP instance.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/roles/configuration
```

Method:

```
GET
```

Example Response:

```
{
  "responseMessage": "SUCCESS",
  "restUri": null,
  "result": {
    "roles": [
      {
        "roleId": 9,
        "roleName": "Project Administrator",
        "roleType": "SYSTEM_DEFINED",
        "isGlobal": false
      },
      {
        "roleId": 10,
        "roleName": "Project Designer",
        "roleType": "USER_DEFINED",
        "isGlobal": false
      },
      {
        "roleId": 11,
        "roleName": "Project Operations",
        "roleType": "USER_DEFINED",
        "isGlobal": false
      },
      {
        "roleId": 12,
        "roleName": "Project Business",
        "roleType": "USER_DEFINED",
        "isGlobal": false
      },
      {
        "roleId": 13,
        "roleName": "Project Data Provider",
        "roleType": "USER_DEFINED",
        "isGlobal": false
      }
    ]
  }
}
```

```
{
  "roleId": 14,
  "roleName": "Project BDCA",
  "roleType": "BEDROCK_INTERNAL",
  "isGlobal": false
},
{
  "roleId": 15,
  "roleName": "Project Executor",
  "roleType": "BEDROCK_INTERNAL",
  "isGlobal": false
}
],
"userGroupIds": [
  {
    "id": 1,
    "name": "admin",
    "isUserId": true,
    "mappingId": 0,
    "sequenceId": 1
  },
  {
    "id": 2,
    "name": "workflowexecutor",
    "isUserId": true,
    "mappingId": 0,
    "sequenceId": 2
  },
  {
    "id": 3,
    "name": "bdca",
    "isUserId": true,
    "mappingId": 0,
    "sequenceId": 3
  },
  {
    "id": 11,
    "name": "shiva",
    "isUserId": true,
    "mappingId": 0,
    "sequenceId": 5
  },
  {
    "id": 12,
    "name": "truser7",
    "isUserId": true,
    "mappingId": 0,
    "sequenceId": 6
  },
  {
    "id": 13,
    "name": "truser8",
    "isUserId": true,
    "mappingId": 0,
    "sequenceId": 7
  },
  {
    "id": 14,
```

```

        "name": "truser9",
        "isUserId": true,
        "mappingId": 0,
        "sequenceId": 8
      },
      {
        "id": 15,
        "name": "truser10",
        "isUserId": true,
        "mappingId": 0,
        "sequenceId": 9
      }
    ],
    "page": null
  }
}

```

Response in case of failure:

If the user does not have necessary permission to access this service.

```

{
  "responseMessage": "User {username} doesn't have access to some of these_
↳ permissions - admin_manage_project,admin_manage_users",
  "restUri": "/bedrock-app/services/rest/projects/roles/configuration",
  "result": null,
  "page": null
}

```

1.10.4 Create a ZDP Project

This API allows you to create a project in ZDP. To use this API, you must be a ZDP-level administrator having at least the *Manage Projects* permission.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects
```

Method:

```
POST
```

Request Payload:

```

{
  *"name": "<This is the unique name for the project, that you want to create.>",
  "description": "<This is the description about the project. This is an_
↳ optional parameter that supports upto 255 characters.>",
  *"roleAssociations": [ "<This is an array to set role to user mappings within the_
↳ project.>"
    {
      *"roleId": "<This is the unique Id of the role maintained in ZDP;_
↳ to which users within a project need to be mapped.>",
      "roleName": "<This is the name of the role existing in ZDP.>",
      "userGroupIds": [
        {
          *"category": "<This is the category type. For role_
↳ to user mapping, set the value as: user. To map the role to a LDAP group, set the_
↳ value as: group.>",

```

```

        *"id": "<This is the user Id (or group Id)>",
        *"name": "<This is the user (or group name)>",
        *"isUserId": "<Set the value of this parameter as
        ↳true. set the value as false, if the value against the category parameter is set as
        ↳group.>"
    }
  ]
}

```

Example Request:

To create a project with users and groups mapped to different roles:

```

{
  "name": "Schivas",
  "description": "This is a sample ZDP project",
  "roleAssociations": [
    {
      "roleId": 9,
      "roleName": "Project Administrator",
      "checked": true,
      "userGroupIds": [
        {
          "category": "user",
          "id": 1,
          "name": "admin",
          "isUserId": true
        }
      ]
    },
    {
      "roleId": 13,
      "roleName": "Project Data Provider",
      "checked": true,
      "userGroupIds": [
        {
          "category": "user",
          "id": 16,
          "name": "Eva",
          "isUserId": true
        },
        {
          "category": "group",
          "id": 1,
          "name": "cn=test_group,dc=zalonilabs,dc=com",
          "isUserId": false
        },
        {
          "category": "group",
          "id": 2,
          "name": "cn=qa,dc=zalonilabs,dc=com",
          "isUserId": false
        }
      ]
    }
  ]
}

```



```
    ]
  }
}
```

Example Response:

If a project is created successfully with the Response Code - 200:

```
{
  "responseMessage": "Project created successfully",
  "restUri": "/projects",
  "result": {
    "id": 19,
    "name": "SB_2304_BP",
    "description": "This is a sample ZDP project",
    "createdTime": 1487752724788,
    "createdBy": "admin",
    "modifiedTime": 1487752724788,
    "modifiedBy": "admin",
    "roleAssociations": [
      {
        "roleId": 9,
        "roleName": "Project Administrator",
        "userGroupIds": [
          {
            "id": 1,
            "name": "admin",
            "isUserId": true,
            "mappingId": 38,
            "sequenceId": 0
          }
        ]
      },
      {
        "roleId": 13,
        "roleName": "Project Data Provider",
        "userGroupIds": [
          {
            "id": 16,
            "name": "tulya",
            "isUserId": true,
            "mappingId": 39,
            "sequenceId": 0
          },
          {
            "id": 1,
            "name": "cn=test_group,dc=zalonilabs,dc=com",
            "isUserId": false,
            "mappingId": 0,
            "sequenceId": 0
          },
          {
            "id": 2,
            "name": "cn=qa,dc=zalonilabs,dc=com",
            "isUserId": false,
            "mappingId": 0,
            "sequenceId": 0
          }
        ]
      }
    ]
  }
}
```

```
    ],
    "page": null
}
```

Response in Case of Failure:

If the role Id is invalid:

```
{
  "responseMessage": "Role with id:102 does not exist",
  "restUri": "/bedrock-app/services/rest/projects",
  "result": null,
  "page": null
}
```

1.10.5 Fetch the Details of a Project

This API allows you to fetch the details of a project. You can fetch the roles to user mapping details along with additional project information.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/{project_Id}
```

Note: You must pass the *project_Id* parameter in the API URL. To fetch the details of the public project, set the *project_Id* parameter to 1.

Method:

```
GET
```

Example Response:

```
{
  "responseMessage": "SUCCESS",
  "restUri": "/projects/19",
  "result": {
    "id": 19,
    "name": "SB_2304_BP",
    "description": "This is a sample ZDP project",
    "createdTime": 1487752724000,
    "createdBy": "admin",
    "modifiedTime": 1487752724000,
    "modifiedBy": "admin",
    "roleAssociations": [
      {
        "roleId": 9,
        "roleName": "Project Administrator",
        "userGroupIds": [
          {
            "id": 1,
            "name": "admin",
```

```
        "isUserId": true,
        "mappingId": 38,
        "sequenceId": 0
      }
    ],
    {
      "roleId": 13,
      "roleName": "Project Data Provider",
      "userGroupIds": [
        {
          "id": 16,
          "name": "Eva",
          "isUserId": true,
          "mappingId": 39,
          "sequenceId": 0
        },
        {
          "id": 1,
          "name": "cn=test_group,dc=zalonilabs,dc=com",
          "isUserId": false,
          "mappingId": 0,
          "sequenceId": 0
        },
        {
          "id": 2,
          "name": "cn=qa,dc=zalonilabs,dc=com",
          "isUserId": false,
          "mappingId": 0,
          "sequenceId": 0
        }
      ]
    }
  ],
  "page": null
}
```

Response in Case of Failure:

If the project Id specified in the URL does not exist.

```
{
  "responseMessage": "Project with id: {project_Id} does not exist",
  "restUri": "/bedrock-app/services/rest/projects/{project_Id}",
  "result": null,
  "page": null
}
```

1.10.6 Update Role-User Mapping within a Project

Use this API to update the Role to User mapping for an existing project.

Note: This API overwrites the entire role to user mapping for the project, based on the request payload.

1. Use the *Fetch the Details of a Project* API to get the details of the project for which the Role to User mapping needs to be updated.
2. Use the following service.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/{project_Id}/  
↪roleassociation
```

Note: You must pass the *project_Id* in the request URL.

Method:

```
PUT
```

Request Payload:

To update an existing role to user mapping as well as to include a new user to an existing role within the project.

```
[  
  {  
    "roleId": 9,  
    "roleName": "Project Administrator",  
    "userGroupIds": [  
      {  
        "id": 1,  
        "name": "admin",  
        "isUserId": true,  
        "mappingId": 44  
      },  
      {  
        "id": 11,  
        "name": "Evan",  
        "isUserId": true,  
        "mappingId": 78  
      }  
    ]  
  },  
  {  
    "roleId": 26,  
    "roleName": "Assistant Administrator",  
    "userGroupIds": [  
      {  
        "id": 31,  
        "name": "asst_admin",  
        "isUserId": true,  
        "mappingId": 39  
      },  
      {  
        "id": 37,  
        "name": "Avean",  
        "isUserId": true,  
        "mappingId": 0  
      }  
    ]  
  }  
]
```

Response in Case of Failure:

If the *project_Id* passed in the URL does not exist:

```
{
  "responseMessage": "Project with Id 91 doesn't exist",
  "restUri": "/bedrock-app/services/rest/projects/91/roleassociation",
  "result": null,
  "page": null
}
```

1.10.7 Update a Project

This API allows you to add new user(s), update role to user mapping, or include additional information of a project; already existing in ZDP. As a good practice, you can follow these steps to use this service efficiently.

1. Use the *Fetch the Details of a Project* API to get the details of the project that needs to be updated.
2. Use the following service thereafter.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/{project_Id}
```

Note: You must pass the *project_Id* parameter in the API URL.

Method:

```
PUT
```

Request Payload:

```
{
  *"name": "<This is the name of the project that needs to be updated.>",
  "description": "<This is the updated description about the project. This_
↳parameter can support upto 255 characters.>",
  *"roleAssociations": [ "<This is an array to set role to user mappings within the_
↳project.>"
    {
      *"roleId": "<This is the unique Id of the role maintained within_
↳ZDP; to which users within a project are to be mapped.>",
      "roleName": "<This is the name of the role, existing in ZDP.>",
      *"userGroupIds": [
        {
          *"category": "<This is the category type. For role_
↳to user mapping, set the value as: user. To map the role to a LDAP group, set the_
↳value as: group.>",
          *"id": "<This is the user Id (or group Id)_
↳maintained within ZDP.>",
          "name": "<This is the username (or group_
↳name) that may be updated.>",
          *"isUserId": "<Set the value as true when the_
↳category parameter is set as user. When the category parameter is set as group, set_
↳the value as false.>",
          *"mappingId": "<This is an integer value (unique_
↳within a specific project) that acts as an identifier to indicate the mapping of_
↳the role to a user. This must be passed in the request payload only when an_
↳existing user needs to be updated. This parameter is not applicable for LDAP groups_
↳or while adding new user(s) to an existing project (and the value can be set as_
```

```

    }
  ]
}

```

Note: This API overwrites the entire role to user mapping for the project, based on the request payload.

Request Payload:

To update an existing role to user mapping as well as to include a new user to an existing role for a specific project:

```

{
  "name": "Sector_431",
  "description": "This is an sample project that is being updated",
  "roleAssociations": [
    {
      "roleId": 9,
      "roleName": "Project Administrator",
      "userGroupIds": [
        {
          "id": 1,
          "name": "admin",
          "isUserId": true,
          "mappingId": 5
        },
        {
          "id": 5,
          "name": "do2user2",
          "isUserId": true,
          "mappingId": 6
        }
      ]
    },
    {
      "roleId": 10,
      "roleName": "Project Designer",
      "userGroupIds": [
        {
          "id": 1,
          "name": "CN=ad2,OU=zaloni,DC=zaloni,DC=com",
          "isUserId": false
        },
        {
          "id": 3,
          "name": "CN=ad1,OU=zaloni,DC=zaloni,DC=com",
          "isUserId": false,
          "mappingId": null
        },
        {
          "id": 17,
          "name": "user221_new",
          "isUserId": true,
          "mappingId": null
        }
      ]
    }
  ]
}

```

```

    },
    {
        "id": 14,
        "name": "user220_new",
        "isUserId": true
    }
]
}

```

Response in Case of Failure:

If the user Id/role Id is incorrect:

```

{
    "responseMessage": "org.hibernate.ObjectNotFoundException: No row with the_
↳given identifier exists: [com.zaloni.bedrock.security.ldap.entity.
↳LdapGroupRoleMapping#322]",
    "restUri": "/bedrock-app/services/rest/projects/5",
    "result": null,
    "page": null
}

```

1.10.8 Share Artifact with Project(s)

The following API allows users to share an artifact with one (or multiple) project.

URL:

```

http://<bedrock-host>:port/bedrock-app/services/rest/projects/{source_project_Id}/
↳artifacts/share/batch

```

Request Payload:

```

[
    {
        *"artifactId": "<This is an integer value that represents the Id of the_
↳artifact that needs to be shared.>",
        *"artifactType": "<This is the type of the artifact in ZDP. Probable_
↳values can be 'ENTITY', 'WORKFLOW', 'DASHBOARD', 'WORKFLOW_SCHEDULE',
↳'TRANSFORMATION', or 'FILE_PATTERN'>",
        *"projectList": ["<This is the project or array of projects across which_
↳the artifacts needs to be shared.>"
            {
                *"projectId": "<This is an integer value that represents_
↳the Id of the project with which the artifact needs to be shared.>",
                "projectName": "<This is the name of the project with_
↳which the artifact needs to be shared.>"
            }
        ]
    }
]

```

Example Request:

- Scenario 1:

To share an entity with another ZDP project:

```
[
  {
    "artifactId": 78,
    "artifactType": "ENTITY",
    "projectList": [
      {
        "projectId": 2,
        "projectName": "Sector 7"
      }
    ]
  }
]
```

- Scenario 2:

To share an entity with multiple ZDP projects:

```
[
  {
    "artifactId": 121,
    "artifactType": "ENTITY",
    "projectList": [
      {
        "projectId": 4
      },
      {
        "projectId": 8,
        "projectName": "Zone Zabber"
      }
    ]
  }
]
```

- Scenario 3:

To share a workflow with multiple ZDP projects:

```
[
  {
    "artifactId": 21,
    "artifactType": "WORKFLOW",
    "projectList": [
      {
        "projectId": 4,
        "projectName": "Designers Cat1"
      },
      {
        "projectId": 8,
        "projectName": "Designers Cat2"
      }
    ]
  }
]
```

- Scenario 4:

To share a File Pattern with multiple ZDP projects.


```
[
  {
    "artifactId": 17,
    "artifactType": "FILE_PATTERN",
    "projectList": [
      {
        "projectId": 4,
        "projectName": "Data Monitor Cat1"
      },
      {
        "projectId": 8,
        "projectName": "Data Monitor Cat2"
      },
      {
        "projectId": 12,
        "projectName": "Data Monitor Cat3"
      }
    ]
  }
]
```

Example Response:

If an artifact is successfully shared.

```
{
  "responseMessage": "Artifacts shared with selected projects",
  "restUri": null,
  "result": true,
  "page": null
}
```

Response in case of failure:

- Scenario 1:

If the artifact does not belong to the current project:

```
{
  "responseMessage": "Artifact with Id 1323 and type ENTITY doesn't belong_
↳to project with Id 4",
  "restUri": "/bedrock-app/services/rest/projects/7/artifacts/share/batch",
  "result": null,
  "page": null
}
```

- Scenario 2:

If you do not have permissions to share artifacts:

```
{
  "responseMessage": "User john doesn't have access to some of these_
↳permissions - pm_share_artifacts",
  "restUri": "/bedrock-app/services/rest/projects/4/artifacts/share/batch",
  "result": null,
  "page": null
}
```

1.10.9 Unshare Artifact(s) from Project

This API allows you to unshare artifact(s) from a particular project.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/{source_project_Id}/  
↪artifacts/unshare/batch
```

Note: Specify the parameter: *source_project_Id* where the artifacts currently exist and need to be unshared (or removed).

Example Request:

- Scenario 1:

To unshare an entity from a project (shared earlier).

```
[  
  {  
    "artifactId": 28,  
    "artifactType": "ENTITY"  
  }  
]
```

- Scenario 2:

To unshare multiple workflows from a project:

```
[  
  {  
    "artifactId": 121,  
    "artifactType": "WORKFLOW"  
  },  
  {  
    "artifactId": 122,  
    "artifactType": "WORKFLOW"  
  },  
  {  
    "artifactId": 123,  
    "artifactType": "WORKFLOW"  
  },  
  {  
    "artifactId": 124,  
    "artifactType": "WORKFLOW"  
  }  
]
```

- Scenario 3:

To unshare multiple file patterns from a project:

```
[  
  {  
    "artifactId": 121,  
    "artifactType": "FILE_PATTERN"  
  },  
  {  
    "artifactId": 122,  
    "artifactType": "FILE_PATTERN"  
  }  
]
```

```

        "artifactId": 122,
        "artifactType": "FILE_PATTERN"
    },
    {
        "artifactId": 123,
        "artifactType": "FILE_PATTERN"
    }
]

```

Response in case of failure:

• Scenario 1:

If you try to unshare artifacts of different types (such as *WORKFLOW*, *ENTITY*):

```

{
    "responseMessage": "All artifacts are not of the same type",
    "restUri": "/bedrock-app/services/rest/projects/7/artifacts/unshare/batch
↪",
    "result": null,
    "page": null
}

```

• Scenario 2:

If the artifacts do not exist in the specified project:

```

{
    "responseMessage": "Specified artifacts not found in given project",
    "restUri": "/bedrock-app/services/rest/projects/12/artifacts/unshare/batch
↪",
    "result": null,
    "page": null
}

```

1.10.10 Transfer Project Artifacts

This API allows you to transfer ingestion, metadata, and workflow level artifacts from one project to another based on the role permission set for the currently logged-in user.

URL:

```

http://<bedrock-host>:port/bedrock-app/services/rest/projects/{source_project_Id}/
↪artifacts/transfer/batch?destinationProject={destination_project_Id}

```

In the URL, specify:

- *source_project_Id*: The Id of the project where the artifacts exist currently.
- *destination_project_Id*: The Id of the project to which you want to transfer the artifacts.

Method:

```
POST
```

Request Payload:

```
[
  {
    *"artifactId": "<This is the artifact Id based on the defined artifactType.
    →>"
    *"artifactType": "<This is the type of the artifact that needs to be_
    →transferred from one project to another. Probable value can be FILE_PATTERN, ENTITY,
    → WORKFLOW, or TRANSFORMATION.>"
  }
]
```

Example Request:

- Scenario 1:

To transfer entity from one project to another:

```
[
  {
    "artifactId": 62,
    "artifactType": "ENTITY",
    "includeDependencies": false
  }
]
```

- Scenario 2:

To transfer a workflow and the associated schedules from one project to another:

```
[
  {
    "artifactId": 10,
    "artifactType": "WORKFLOW"
  },
  {
    "artifactId": 40,
    "artifactType": "SCHEDULE"
  },
  {
    "artifactId": 39,
    "artifactType": "SCHEDULE"
  }
]
```

- Scenario 3:

To transfer a transformation from one project to another:

```
[
  {
    "artifactId": 31,
    "artifactType": "TRANSFORMATION"
  }
]
```

- Scenario 4:

To transfer multiple artifacts (below) from one project to another:

- A workflow and associated schedules
- An independent workflow

- File patterns
- Entities

```
[
  {
    "artifactId": 10,
    "artifactType": "WORKFLOW"
  },
  {
    "artifactId": 51,
    "artifactType": "SCHEDULE"
  },
  {
    "artifactId": 52,
    "artifactType": "SCHEDULE"
  },
  {
    "artifactId": 19,
    "artifactType": "WORKFLOW"
  },
  {
    "artifactId": 22,
    "artifactType": "ENTITY"
  },
  {
    "artifactId": 66,
    "artifactType": "ENTITY"
  },
  {
    "artifactId": 77,
    "artifactType": "FILE_PATTERN"
  },
  {
    "artifactId": 78,
    "artifactType": "FILE_PATTERN"
  }
]
```

Example Response:

If the artifacts are transferred successfully:

```
{
  "responseMessage": "SUCCESS",
  "restUri": "/projects/1/artifacts/transfer/batch",
  "result": true,
  "page": null
}
```

Response in Case of Failure:

If the artifact cannot be transferred:

```
{
  "responseMessage": "<Failure Message>",
  "restUri": "/bedrock-app/services/rest/projects/33/artifacts/transfer/batch",
  "result": null,
  "page": null
}
```

1.10.10.1 Delete a project

This API allows you to delete a project that already exists in ZDP.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/{project_Id}
```

Note: You must pass the *project_Id* parameter in the API URL.

Method:

```
DELETE
```

Example Response:

If the project is deleted successfully:

```
{
  "responseMessage": "Project deleted successfully",
  "restUri": "/projects/5",
  "result": null,
  "page": null
}
```

Response in Case of Failure:

- Scenario 1:

If the user does not have permissions to delete project in ZDP:

```
{
  "responseMessage": "User {username} doesn't have access to some of these_
↪permissions - admin_manage_project",
  "restUri": "/bedrock-app/services/rest/projects/58",
  "result": null,
  "page": null
}
```

- Scenario 2:

If the project Id specified in the URL does not exist in the ZDP instance:

```
{
  "responseMessage": "java.lang.IllegalArgumentException: attempt to create_
↪delete event with null entity",
  "restUri": "/bedrock-app/services/rest/projects/51",
  "result": null,
  "page": null
}
```

1.10.11 Sync Project Policies in Ranger

This service can be invoked only when ZDP is integrated with Apache Ranger™.

This API allows you to sync all the project policies, replace the existing policies for the projects, and delete policies for the projects that do not have any resources or user/group associations.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/security/policies/  
↪refresh
```

Method:

```
GET
```

Example Response:

```
{  
  "responseMessage": "Synced all projects with cluster security managers",  
  "restUri": null,  
  "result": true,  
  "page": null  
}
```

1.11 Administration API

1.11.1 Get System Configuration

ZDP provides a host of system variables that are required for setting system level configurations across the various modules.

This API allows you to fetch all the system variables set for the ZDP instance.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/admin/getSystemConfig
```

Method:

```
GET
```

Example Response:

```
{  
  "list": [{  
    "description": "Is the administration email address; ZDP uses this_  
↪email address for sending the notification mails",  
    "id": 58,  
    "required": 1,  
    "sysKey": "ADMIN_EMAIL_ADDRESS",  
    "sysValue": "bedrock-admin@zaloni.com",  
    "systemConfigTagList": [{  
      "id": 1,  
      "tag": "GENERAL"  
    }],  
    "type": "text"  
  }, {  
    "description": "This Represents the base log directory, where_  
↪workflow?s instance log will be generated",
```

```

        "id": 1,
        "required": 1,
        "sysKey": "APP_LOGS",
        "sysValue": "/share/applogs441_29Oct",
        "systemConfigTagList": [{
            "id": 1,
            "tag": "GENERAL"
        }],
        "type": "text"
    }, {
        "description": "HDFS location where ZDP libraries are stored.",
        "id": 11,
        "required": 1,
        "sysKey": "BEDROCK_HDFS_LIB_DIR",
        "sysValue": "/user/bedrock/lib",
        "systemConfigTagList": [{
            "id": 1,
            "tag": "GENERAL"
        }], {
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "The ZDP application URL i.e. https://bedrock-host:port",
        "id": 2,
        "required": 1,
        "sysKey": "BEDROCK_URL",
        "sysValue": "http://192.168.2.128:9090",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }], {
            "id": 1,
            "tag": "GENERAL"
        }],
        "type": "text"
    }, {
        "description": "Directory path where ZDP is expecting custom_
tokenizer plugin jars",
        "id": 100,
        "required": 0,
        "sysKey": "CUSTOM_TOKENIZER_JAR_PATH_APP",
        "sysValue": "",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "Set TRUE if a checkpoint needs to be set for Data_
Inventory (DI) Action; by default it is FALSE",
        "id": 74,
        "required": 1,
        "sysKey": "DATA_INVENTORY_CHECKPOINT_ENABLED",
        "sysValue": "false",
        "systemConfigTagList": [{

```



```

        "id": 3,
        "tag": "WORKFLOW"
    }],
    "type": "flag"
}, {
    "description": "Is the count of files after which a checkpoint will_
↪be created to avoid reprocessing of files in an event of unavoidable error/ failure_
↪of the DI Action",
    "id": 76,
    "required": 1,
    "sysKey": "DATA_INVENTORY_CHECKPOINT_INTERVAL",
    "sysValue": "50",
    "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
    }],
    "type": "text"
}, {
    "description": "The DFS directory that ADI scans for data.",
    "id": 13,
    "required": 1,
    "sysKey": "DATA_INVENTORY_DFS_BASE_DIR",
    "sysValue": "/user/bedrock/data-inventory",
    "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
    }],
    "type": "text"
}, {
    "description": "Is the directory where heuristics information to_
↪speed up DFS scanning will be stored",
    "id": 14,
    "required": 1,
    "sysKey": "DATA_INVENTORY_DFS_HEURISTICS_DIR",
    "sysValue": "/user/bedrock/data-inventory/heuristics",
    "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
    }],
    "type": "text"
}, {
    "description": "Is the output directory for Data Inventory Action_
↪where error, report files generated during execution will be stored",
    "id": 15,
    "required": 1,
    "sysKey": "DATA_INVENTORY_DFS_OUTPUT_BASE_DIR",
    "sysValue": "/user/bedrock/data-inventory/output",
    "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
    }],
    "type": "text"
}, {
    "description": "Is the DFS directory where transient ADI data will_
↪reside",
    "id": 16,
    "required": 1,
    "sysKey": "DATA_INVENTORY_DFS_WORK_DIR",

```

```

        "sysValue": "/user/bedrock/data-inventory/working-dir",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "Is the Distributed File System (DFS) directory where ↵
↵ADI data will reside",
        "id": 17,
        "required": 1,
        "sysKey": "DATA_INVENTORY_DFS_WORK_DIR_TEMP",
        "sysValue": "/user/bedrock/data-inventory/working-dir/tmp",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "Is to speed up scanning of files; a map task is ↵
↵spawned for each directory that has more files than the count set against this ↵
↵parameter",
        "id": 72,
        "required": 1,
        "sysKey": "DATA_INVENTORY_DIR_SIZE",
        "sysValue": "10000",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "Is the threshold count of the number of internal ↵
↵errors encountered during execution of DI Action; reaching this shall forcefully ↵
↵abort the job",
        "id": 73,
        "required": 1,
        "sysKey": "DATA_INVENTORY_ERROR_THRESHOLD",
        "sysValue": "0",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "Is the split size to run multiple map tasks against ↵
↵different parts of a single input file; similar to Hadoop's mapred.max.split.size.",
        "id": 75,
        "required": 1,
        "sysKey": "DATA_INVENTORY_FILE_CHUNK_SIZE",
        "sysValue": "2097152",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "Date format for displaying dates in ZDP. All formats ↵
↵available in https://docs.angularjs.org/api/ng/filter/date are applicable.",

```

```

        "id": 84,
        "required": 1,
        "sysKey": "DATE_DISPLAY_FORMAT",
        "sysValue": "medium",
        "systemConfigTagList": [{
            "id": 1,
            "tag": "GENERAL"
        }],
        "type": "text"
    }, {
        "description": "The default batch size associated to a distribution_
↪list",
        "id": 107,
        "required": 1,
        "sysKey": "DEFAULT_BATCH_SIZE_DISTRIBUTION_LIST",
        "sysValue": "500",
        "systemConfigTagList": [{
            "id": 11,
            "tag": "NOTIFICATION"
        }],
        "type": "text"
    }, {
        "description": "The default batch size of entities in a data_
↪profiling action.",
        "id": 111,
        "required": 1,
        "sysKey": "DP_DEFAULT_BATCH_SIZE",
        "sysValue": "1",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "The default number of mappers to be used in a data_
↪profiling action. Set 0 for unlimited.",
        "id": 112,
        "required": 1,
        "sysKey": "DP_DEFAULT_MAPPERS_COUNT",
        "sysValue": "0",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "The default number of reducers to be used in a data_
↪profiling action. Set 0 for unlimited.",
        "id": 113,
        "required": 1,
        "sysKey": "DP_DEFAULT_REDUCERS_COUNT",
        "sysValue": "0",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {

```

Chapter 1. REST API Guide

```

        "id": 3,
        "tag": "WORKFLOW"
    }, {
        "id": 2,
        "tag": "METADATA"
    }],
    "type": "text"
}, {
    "description": "The default number of records to be displayed for_
→most common value in data profiling.",
    "id": 92,
    "required": 1,
    "sysKey": "DP_NUMBER_MOST_COMMON_VALUES",
    "sysValue": "20",
    "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
    }, {
        "id": 2,
        "tag": "METADATA"
    }],
    "type": "text"
}, {
    "description": "HDFS repository for data profiling report files",
    "id": 7,
    "required": 1,
    "sysKey": "DP_REPORT_PATH",
    "sysValue": "/tmp/dp/report",
    "systemConfigTagList": [{
        "id": 2,
        "tag": "METADATA"
    }],
    "type": "text"
}, {
    "description": "HDFS temp working folder for data profiling workflows
→",
    "id": 8,
    "required": 1,
    "sysKey": "DP_TEMP_PATH",
    "sysValue": "/tmp/dp",
    "systemConfigTagList": [{
        "id": 2,
        "tag": "METADATA"
    }],
    "type": "text"
}, {
    "description": "Is a positive integer or a date in yyyy-MM-dd format;_
→defaults to 0",
    "id": 46,
    "required": 1,
    "sysKey": "DQ_DASHBOARD_DURATION",
    "sysValue": "0",
    "systemConfigTagList": [{
        "id": 9,
        "tag": "DATA QUALITY"
    }, {
        "id": 6,
        "tag": "REPORTING"
    }],

```

```

    }, {
      "type": "text"
    }, {
      "description": "Is the message text to represent time duration in DQ_
↪Dashboard tiles footer; defaults to today. If either of DQ_DASHBOARD_DURATION or DQ_
↪DASHBOARD_TILE_MSG variables are missing/invalid, default values will be considered_
↪for both to maintain consistency.",
      "id": 47,
      "required": 1,
      "sysKey": "DQ_DASHBOARD_TILE_MSG",
      "sysValue": "today",
      "systemConfigTagList": [{
        "id": 6,
        "tag": "REPORTING"
      }], {
        "id": 9,
        "tag": "DATA QUALITY"
      }],
      "type": "text"
    }, {
      "description": "The Data Quality Engine to use to run Data Quality_
↪Action. Valid value is HIVESQL",
      "id": 102,
      "required": 1,
      "sysKey": "DQ_ENGINE",
      "sysValue": "HIVESQL",
      "systemConfigTagList": [{
        "id": 9,
        "tag": "DATA QUALITY"
      }],
      "type": "text"
    }, {
      "description": "Delimiter for the Data Quality error file data",
      "id": 55,
      "required": 1,
      "sysKey": "DQ_ERRORFILE_DELIMITER",
      "sysValue": ",",
      "systemConfigTagList": [{
        "id": 9,
        "tag": "DATA QUALITY"
      }],
      "type": "text"
    }, {
      "description": "Represents the delimiter for the Data Quality reports
↪",
      "id": 54,
      "required": 1,
      "sysKey": "DQ_REPORT_DELIMITER",
      "sysValue": ",",
      "systemConfigTagList": [{
        "id": 9,
        "tag": "DATA QUALITY"
      }],
      "type": "text"
    }, {
      "description": "Is the HDFS output location of the Data Quality_
↪action; MapReduce job is collected here temporarily",
      "id": 9,

```

```

        "required": 1,
        "sysKey": "DQ_TEMP_PATH",
        "sysValue": "/tmp/dq",
        "systemConfigTagList": [{
            "id": 9,
            "tag": "DATA QUALITY"
        }],
        "type": "text"
    }, {
        "description": "Is the email server URL in the form: <ip_of_email_
↪server>:<port>",
        "id": 20,
        "required": 1,
        "sysKey": "emailserver",
        "sysValue": "",
        "systemConfigTagList": [{
            "id": 1,
            "tag": "GENERAL"
        }],
        "type": "text"
    }, {
        "description": "Set it TRUE to enable various features that are
↪deprecated in ZDP. The default value is FALSE",
        "id": 82,
        "required": 1,
        "sysKey": "ENABLE_DEPRECATED_FEATURES",
        "sysValue": "false",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }, {
            "id": 1,
            "tag": "GENERAL"
        }],
        "type": "flag"
    }, {
        "description": "Set it TRUE to enable inline scripting in Shell and
↪Python action in a workflow",
        "id": 101,
        "required": 1,
        "sysKey": "ENABLE_INLINE_SCRIPT",
        "sysValue": "true",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "flag"
    }, {
        "description": "Define bulk size for concurrent execution to ES.
↪Default size is 1000",
        "id": 103,
        "required": 1,
        "sysKey": "ES_BULK_SIZE",
        "sysValue": "1000",
        "systemConfigTagList": [{
            "id": 9,
            "tag": "DATA QUALITY"
        }],

```

```

        "type": "text"
      }, {
        "description": "Is the index in the elastic search that is used for_
↪Data Quality reporting",
        "id": 44,
        "required": 1,
        "sysKey": "ES_DQ_INDEX",
        "sysValue": "dqparentreport",
        "systemConfigTagList": [{
          "id": 1,
          "tag": "GENERAL"
        }, {
          "id": 9,
          "tag": "DATA QUALITY"
        }, {
          "id": 3,
          "tag": "WORKFLOW"
        }, {
          "id": 6,
          "tag": "REPORTING"
        }
      ],
        "type": "text"
      }, {
        "description": "The mapping type in elastic search for data quality.",
        "id": 45,
        "required": 1,
        "sysKey": "ES_DQ_TYPE",
        "sysValue": "reportfields",
        "systemConfigTagList": [{
          "id": 6,
          "tag": "REPORTING"
        }, {
          "id": 1,
          "tag": "GENERAL"
        }, {
          "id": 3,
          "tag": "WORKFLOW"
        }, {
          "id": 9,
          "tag": "DATA QUALITY"
        }
      ],
        "type": "text"
      }, {
        "description": "Specify Number of ES Bulks to be executed_
↪concurrently.",
        "id": 104,
        "required": 1,
        "sysKey": "ES_MAX_CONCURRENT_BULK_REQUESTS ",
        "sysValue": "2",
        "systemConfigTagList": [{
          "id": 9,
          "tag": "DATA QUALITY"
        }
      ],
        "type": "text"
      }, {
        "description": "The location of uploaded files",
        "id": 4,
        "required": 1,

```



```

        "sysKey": "FILE_UPLOAD_LOCATION",
        "sysValue": "/projects/zaloni/BEDROCK_APP_HOME/441_29Oct2017_56/
↪bedrock/bedrock-app/uploaded-files",
        "systemConfigTagList": [{
            "id": 1,
            "tag": "GENERAL"
        }],
        "type": "text"
    }, {
        "description": "Is the Home directory of the flume binaries; the
↪binaries are installed to BEDROCK_HOME/flume directory by the ZDP installer",
        "id": 3,
        "required": 1,
        "sysKey": "FLUME_HOME",
        "sysValue": "/projects/zaloni/BEDROCK_APP_HOME/441_29Oct2017_56/
↪bedrock/flume",
        "systemConfigTagList": [{
            "id": 8,
            "tag": "INGESTION"
        }],
        "type": "text"
    }, {
        "description": "The default Hive schema to which the Hive tables
↪detected by ADI will be added",
        "id": 77,
        "required": 1,
        "sysKey": "HCATALOG_TARGET_SCHEMA",
        "sysValue": "bedrock",
        "systemConfigTagList": [{
            "id": 2,
            "tag": "METADATA"
        }],
        "type": "text"
    }, {
        "description": "The time (in seconds) after which the write lock,
↪acquired by a workflow on an HDFS location, is released. Setting the value to -1
↪will mean no timeout.",
        "id": 51,
        "required": 1,
        "sysKey": "HDFS_LOCK_EXPIRY_SECONDS",
        "sysValue": "-1",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "Is the maximum number of attempts before reaching a
↪Change Data Capture (CDC) lock out",
        "id": 49,
        "required": 1,
        "sysKey": "HDFS_LOCK_MAX_RETRIES",
        "sysValue": "-1",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }

```

```

    }, {
      "description": "Is the interval between two retries used by CDC_
↪action (in seconds)",
      "id": 48,
      "required": 1,
      "sysKey": "HDFS_LOCK_WAIT_INTERVAL_SECONDS",
      "sysValue": "60",
      "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
      }],
      "type": "text"
    }, {
      "description": "Is the timeout for a CDC lock out (in seconds)",
      "id": 50,
      "required": 1,
      "sysKey": "HDFS_LOCK_WAIT_TIMEOUT_SECONDS",
      "sysValue": "-1",
      "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
      }],
      "type": "text"
    }, {
      "description": "The path to the HDFS directory that is used for_
↪temporary files. In a multi-cluster environment, the same directory is expected to_
↪be present in each cluster.",
      "id": 19,
      "required": 1,
      "sysKey": "HDFS_TEMP_DIR",
      "sysValue": "/user/bedrock/tmp",
      "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
      }],
      {
        "id": 1,
        "tag": "GENERAL"
      }],
      "type": "text"
    }, {
      "description": "Whether to skip adding the ZDP jars while executing_
↪Hive action. If set to true then the JARs must be located in the 'auxiliary jar path
↪' of Hive or else lineage will not be available for the Hive action.",
      "id": 98,
      "required": 0,
      "sysKey": "HIVE_ACTION_SKIP_ADD_JARS",
      "sysValue": "false",
      "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
      }],
      "type": "flag"
    }, {
      "description": "The version of Hive in use. Note that all clusters_
↪that ZDP connect to are expected to have the same version of Hive.",
      "id": 5,
      "required": 1,
      "sysKey": "HIVE_VERSION",

```

```

        "sysValue": "1.2",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "The User Identity Service to implement the dual_
↪identity management. Expected value for Centrify environment is CENTRIFY-SCRIPT,
↪leaving the value blank will implement the default implementation",
        "id": 109,
        "required": 0,
        "sysKey": "IDENTITY_MANAGED_SERVICE",
        "sysValue": "CENTRIFY-SCRIPT",
        "systemConfigTagList": [{
            "id": 1,
            "tag": "GENERAL"
        }],
        "type": "text"
    }, {
        "description": "The boolean to indicate if the impersonation is_
↪enabled for ZDP.",
        "id": 91,
        "required": 1,
        "sysKey": "IMPERSONATION_ENABLED",
        "sysValue": "true",
        "systemConfigTagList": [{
            "id": 5,
            "tag": "SCHEDULE"
        }, {
            "id": 4,
            "tag": "TRANSFORMATION"
        }, {
            "id": 3,
            "tag": "WORKFLOW"
        }, {
            "id": 2,
            "tag": "METADATA"
        }, {
            "id": 7,
            "tag": "LINEAGE"
        }],
        "type": "flag"
    }, {
        "description": "The default input format class used for delimited_
↪entities.",
        "id": 95,
        "required": 1,
        "sysKey": "INPUT_FORMAT_CLASS",
        "sysValue": "org.apache.hadoop.mapred.TextInputFormat",
        "systemConfigTagList": [{
            "id": 2,
            "tag": "METADATA"
        }, {
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }

```

```

    }, {
      "description": "The directory where JDBC jars will be put for DB_
↪Import action. For remote workflow executors this location needs to be a shared_
↪location.",
      "id": 105,
      "required": 0,
      "sysKey": "JDBC_DRIVER_JARS_LOCATION",
      "sysValue": "",
      "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
      }],
      "type": "text"
    }, {
      "description": "The time in milliseconds for which the workflow_
↪executor keeps a DB connection open. A smaller number will mean more connections_
↪opened.",
      "id": 78,
      "required": 1,
      "sysKey": "KEEP_ALIVE_TIME",
      "sysValue": "60000",
      "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
      }], {
        "id": 1,
        "tag": "GENERAL"
      }],
      "type": "text"
    }, {
      "description": "Indicates the maximum number of instances that should_
↪be displayed in the lineage window; the default value being 10.",
      "id": 80,
      "required": 1,
      "sysKey": "LINEAGE_MAX_VIEW_INSTANCE",
      "sysValue": "10",
      "systemConfigTagList": [{
        "id": 7,
        "tag": "LINEAGE"
      }],
      "type": "text"
    }, {
      "description": "Indicates the maximum number of instances that are_
↪displayed in RL (default) Graph direction pre-set as Auto mode; exceeding this_
↪count will automatically shift the orientation of the lineage graph from RL (Right-
↪Left) to UD (Up-Down). By default, the value of this count is set to 5 (five).",
      "id": 81,
      "required": 1,
      "sysKey": "LINEAGE_ORIENTATION_CHANGE_INSTANCE_COUNT",
      "sysValue": "5",
      "systemConfigTagList": [{
        "id": 7,
        "tag": "LINEAGE"
      }],
      "type": "text"
    }, {
      "description": "Is the path where the all configuration, properties_
↪and necessary files for lineage tracker are stored",

```

```

        "id": 10,
        "required": 1,
        "sysKey": "LINEAGE_TRACKER_HOME",
        "sysValue": "/projects/zaloni/BEDROCK_APP_HOME/441_29Oct2017_56/
↪bedrock/data-lineage-trackers",
        "systemConfigTagList": [{
            "id": 7,
            "tag": "LINEAGE"
        }],
        "type": "text"
    }, {
        "description": "The path to the local (shared between bedrock-app and
↪the workflow executors) directory that is used for temporary files.",
        "id": 71,
        "required": 1,
        "sysKey": "LOCAL_TEMP_DIR",
        "sysValue": "/tmp/bedrock",
        "systemConfigTagList": [{
            "id": 1,
            "tag": "GENERAL"
        }, {
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "Is the ID of the entity type that will be used for
↪storing the information of the fields for the Lookup workflow action",
        "id": 22,
        "required": 1,
        "sysKey": "LOOK_UP_ENTITY_TYPE_LIST_ID",
        "sysValue": "",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "Represents the maximum number downloadable records;
↪enter a count value to limit maximum downloadable records",
        "id": 56,
        "required": 1,
        "sysKey": "MAX_DOWNLOADABLE_RECORDS",
        "sysValue": "1000",
        "systemConfigTagList": [{
            "id": 1,
            "tag": "GENERAL"
        }],
        "type": "text"
    }, {
        "description": "Is the maximum allowed file size of uploaded files,
↪in bytes",
        "id": 23,
        "required": 1,
        "sysKey": "MAX_FILE_UPLOAD_SIZE",
        "sysValue": "1048576",
        "systemConfigTagList": [{
            "id": 1,

```

```

        "tag": "GENERAL"
    }],
    "type": "text"
}, {
    "description": "The location where the micro-service tar is present.",
    "id": 6,
    "required": 1,
    "sysKey": "MICRO_SERVICE_SOURCE_DIR",
    "sysValue": "/projects/zaloni/BEDROCK_APP_HOME/441_29Oct2017_56/
↪bedrock/microservice",
    "systemConfigTagList": [{
        "id": 1,
        "tag": "GENERAL"
    }],
    "type": "text"
}, {
    "description": "The identifier for namespace variables that need
↪masking for containing sensitive data",
    "id": 120,
    "required": 1,
    "sysKey": "NAMESPACE_MASK_IDENTIFIER",
    "sysValue": "MSK_",
    "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
    }],
    "type": "text"
}, {
    "description": "The Java date format for worklflow notifications.
↪\n\t\t\t\t\tAll formats available in\n\t\t\t\t\thttps://docs.oracle.com/javase/8/docs/
↪api/java/text/SimpleDateFormat.html\n\t\t\t\t\tare applicable",
    "id": 121,
    "required": 1,
    "sysKey": "NOTIFICATION_DATE_FORMAT",
    "sysValue": "MM-dd-yyyy HH:mm:ss",
    "systemConfigTagList": [{
        "id": 11,
        "tag": "NOTIFICATION"
    }],
    "type": "text"
}, {
    "description": "The default output format class used for delimited
↪entities.",
    "id": 96,
    "required": 1,
    "sysKey": "OUTPUT_FORMAT_CLASS",
    "sysValue": "org.apache.hadoop.hive ql.io.
↪HiveIgnoreKeyTextOutputFormat",
    "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
    }], {
        "id": 2,
        "tag": "METADATA"
    }],
    "type": "text"
}, {
    "description": "Is the maximum of number of rows/records that are to
↪be displayed in reports; this is required to prevent too many records being
↪displayed in a report, which can potentially slow down the application; the default
↪value is 100",

```

```

        "id": 24,
        "required": 1,
        "sysKey": "REPORT_MAX_ROWS",
        "sysValue": "100",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }, {
            "id": 6,
            "tag": "REPORTING"
        }],
        "type": "text"
    }, {
        "description": "The default serde class used for delimited entities.",
        "id": 97,
        "required": 1,
        "sysKey": "SERDE_CLASS",
        "sysValue": "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
        "systemConfigTagList": [{
            "id": 2,
            "tag": "METADATA"
        }, {
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "A string of extra JVM options to pass to the driver.
↪For instance, GC settings or other logging.",
        "id": 118,
        "required": 1,
        "sysKey": "spark.driver.extraJavaOptions",
        "sysValue": "'-Dhdp.version=2.6.0.2.2.4.2-2'",
        "systemConfigTagList": [{
            "id": 4,
            "tag": "TRANSFORMATION"
        }],
        "type": "text"
    }, {
        "description": "Whether dynamic allocation is enabled in Spark. Note
↪that this only indicates whether dynamic allocation is enabled in Spark and does
↪not enable or disable dynamic allocation.",
        "id": 87,
        "required": 1,
        "sysKey": "spark.dynamicAllocation.enabled",
        "sysValue": "false",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "flag"
    }, {
        "description": "The idle time for executors in case dynamic
↪allocation is enabled.",
        "id": 90,
        "required": 1,
        "sysKey": "spark.dynamicAllocation.executorIdleTimeout",
        "sysValue": "60",

```

```

        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "The maximum number of executors in case dynamic_
↪allocation is enabled.",
        "id": 89,
        "required": 1,
        "sysKey": "spark.dynamicAllocation.maxExecutors",
        "sysValue": "4",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "The minimum number of executors in case dynamic_
↪allocation is enabled.",
        "id": 88,
        "required": 1,
        "sysKey": "spark.dynamicAllocation.minExecutors",
        "sysValue": "1",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "Whether the Spark shuffle service is on. Note that_
↪this only indicates whether the service is enabled and does not enable or disable_
↪the service.",
        "id": 86,
        "required": 1,
        "sysKey": "spark.shuffle.service.enabled",
        "sysValue": "false",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "flag"
    }, {
        "description": "A string of extra JVM options to pass to the YARN_
↪Application Master in client mode. In cluster mode, use spark.driver_
↪extraJavaOptions instead.",
        "id": 117,
        "required": 1,
        "sysKey": "spark.yarn.am.extraJavaOptions",
        "sysValue": "-Dhdp.version=2.6.0.2.2.4.2-2",
        "systemConfigTagList": [{
            "id": 4,
            "tag": "TRANSFORMATION"
        }],
        "type": "text"
    }, {
        "description": "Allows selecting whether Spark should be available as_
↪a backend in execution engine; should be ?True? or ?False?. True will enable_
↪Sparkified execution mode.",

```



```

        "id": 39,
        "required": 1,
        "sysKey": "SPARK_ENABLED",
        "sysValue": "false",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "flag"
    }, {
        "description": "Is the timeout (in days) for a workflow step; if a
↪workflow step runs longer than this, it will be marked as failed",
        "id": 21,
        "required": 1,
        "sysKey": "StepTimeOut",
        "sysValue": "",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "Is the number of hours of inactivity after which the
↪authentication token expires",
        "id": 52,
        "required": 1,
        "sysKey": "TOKEN_EXPIRY_IN_HOURS",
        "sysValue": "24",
        "systemConfigTagList": [{
            "id": 1,
            "tag": "GENERAL"
        }, {
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": "text"
    }, {
        "description": "Comma separated list of additional jars (full
↪absolute paths to the jars) that could be needed for Spark jobs. Additional jars
↪might be needed for Spark jobs in some environments.",
        "id": 99,
        "required": 0,
        "sysKey": "TR_EXTRA_JARS",
        "sysValue": "",
        "systemConfigTagList": [{
            "id": 3,
            "tag": "WORKFLOW"
        }],
        "type": " "
    }, {
        "description": "Is the shared location in the server to store all JAR
↪files used for defining and executing UDF in transformation",
        "id": 12,
        "required": 1,
        "sysKey": "TR_SPARK_JARS_DIR",
        "sysValue": "/projects/zaloni/BEDROCK_APP_HOME/441_29Oct2017_56/
↪bedrock/bedrock-workflow-executor/transformation-spark",
        "systemConfigTagList": [{

```

```

        "id": 3,
        "tag": "WORKFLOW"
    }],
    "type": "text"
}, {
    "description": "This is the maximum allowed login attempts with_
↪invalid credential.",
    "id": 119,
    "required": 1,
    "sysKey": "USER_MAX_FAILED_LOGIN",
    "sysValue": "0",
    "systemConfigTagList": [{
        "id": 1,
        "tag": "GENERAL"
    }],
    "type": "int"
}, {
    "description": "The location where the VSAM schema parser jar files_
↪will be created. The location will need to be a shared location accessible by both_
↪the bedrock-app and the workflow-executors",
    "id": 106,
    "required": 1,
    "sysKey": "VSAM_COPYBOOK_PARSER_JAR_LOCATION",
    "sysValue": "",
    "systemConfigTagList": [{
        "id": 3,
        "tag": "WORKFLOW"
    }, {
        "id": 2,
        "tag": "METADATA"
    }],
    "type": "text"
}, {
    "description": "Untar set of XSD files that are zipped in a location_
↪to start processing of Metadata creation.",
    "id": 110,
    "required": 1,
    "sysKey": "XSD_FILE_SET_UNTAR_LOCATION",
    "sysValue": "",
    "systemConfigTagList": [{
        "id": 2,
        "tag": "METADATA"
    }],
    "type": ""
}],
    "status": 1
}

```

1.11.2 Global Search API

This API allows you to search the artifacts in ZDP.

URL:

```
http://<IP>:port/bedrock-app/services/rest/artifacts
```

Method:

POST

Request Payload:

```
{
  "searchTerm": "<This is the term by which you want to perform the search operation.>"
  ↪ ",
  "fields":{
    "<artifact_type: Valid values are METADATA, DQRULE, DQRULESET, WORKFLOW_GS, ↪
    ↪ TRANSFORMATION, NAMESPACE, DMEMODELS>": "<Fields that need to be displayed in the ↪
    ↪ search results for a specific artifact type. For example, you can specify ↪
    ↪ businessName, ownerProjectId, entityTypeId, zone, tableName, etc to be displayed ↪
    ↪ for the METADATA artifact. The fields parameter is optional and all the fields for ↪
    ↪ all the artifacts will be returned (if the parameter is not present in the request ↪
    ↪ payload).>"
    ↪ },
  "artifacts":
    [
      {
        "artifactType": "<This is the artifact type. For example, METADATA, WORKFLOW, ↪
        ↪ DQRULE, DQRULESET, TRANSFORMATION, NAMESPACE_VARIABLE, ALL.>",
        "facetFilter":
          {
            "facetName": "<This is property name.>",
            "facetValue": "<This is the facet value.>"
            ↪ },
        "page":
          {
            "currentPage": "<This is the current page. If not specified, the default page ↪
            ↪ number is 1 - i.e first page.>",
            "chunkSize": "<This is the chunk size. If not specified, the default page size ↪
            ↪ is 20.>"
            ↪ }
          ↪ ]
    }
  ↪ }
```

Example:

```
{
  "searchTerm": "",
  "fields":{
    "METADATA": ["shared", "businessName", "ownerProjectId", "entityTypeId",
    ↪ "version", "zone", "projectIds", "tableName", "entityClusterLocation", "modifiedTime",
    ↪ "modifiedBy", "technicalName", "sourceSchema", "sourcePlatform", "latestVersion",
    ↪ "dataFileFormatId", "dataFileFormat", "dataStoreName"], "DQRULE": ["ruleName",
    ↪ "modifiedDate", "modifiedBy", "ruleId", "dataType", "ruleDesc", "ruleExpression",
    ↪ "invertFunction"], "DQRULESET": ["ruleSetName", "modifiedDate", "modifiedBy", "ruleSetId
    ↪ ", "dataType"], "WORKFLOW_GS": ["shared", "wfName", "ownerProjectId", "modifiedDate",
    ↪ "modifiedBy", "wfId", "projectIds"], "TRANSFORMATION": ["shared", "trMappingId",
    ↪ "trMappingName", "trMappingId", "ownerProjectId", "trModifiedDateString", "trModifiedBy",
    ↪ "projectIds", "isEditable", "trWfId", "trWfName"], "NAMESPACE": ["key", "value", "id"],
    ↪ "DMEMODELS": ["dmModelName", "description", "brEntityName", "status", "id", "dmModelId"]},
  "artifacts": [
    { "artifactType": "METADATA", "facetFilter": [], "page": { "currentPage": 1, "chunkSize": 20 } }
    ↪ ,
    { "artifactType": "DQRULE", "facetFilter": [], "page": { "currentPage": 1, "chunkSize": 20 } },
    { "artifactType": "DQRULESET", "facetFilter": [], "page": { "currentPage": 1, "chunkSize": 20 } }
    ↪ ],
  ↪ }
```

```
{
  "artifactType": "WORKFLOW_GS", "facetFilter": [], "page": {"currentPage": 1, "chunkSize": 20},
  "artifactType": "TRANSFORMATION", "facetFilter": [], "page": {"currentPage": 1, "chunkSize": 20},
  "artifactType": "NAMESPACE", "facetFilter": [], "page": {"currentPage": 1, "chunkSize": 20},
  "artifactType": "DMEMODELS", "facetFilter": [], "page": {"currentPage": 1, "chunkSize": 20}
}
```

Note: For all the facets except *projectIds* and *ruleStatusId*, you must use the *.raw* suffix. For example, if the *facetName* parameter is set to *tableType* and the value is *INTERNAL*, the facet filter in the request is:

```
{
  "facetFilter": {
    "facetName": ["tableType .raw"];
    "facetValue": ["INTERNAL"]
  }
}
```

Example Request:

- Scenario 1:

If you want to search for all the artifacts:

```
{
  "searchTerm": "*",
  "artifactType": "ALL",
  "facetFilter": [],
  "page": {
    "currentPage": 1,
    "chunkSize": 20
  }
}
```

- Scenario 2:

If the initial search is performed without any search term:

```
{
  "searchTerm": "*",
  "artifactType": "ALL",
  "facetFilter": [],
  "page": {
    "currentPage": 1,
    "chunkSize": 20
  }
}
```

- Scenario 3:

If the values for the parameters: *searchTerm* and *facetName* are *Data* and *businessName* respectively:

```
{
  "searchTerm": "Data",
  "artifactType": "METADATA",
  "facetFilter": {
    "facetName": "businessName";
    "facetValue": ["MarketData"]
  },
  "page": {
    "currentPage": 1,
    "chunkSize": 20
  }
}
```

- Scenario 4:

If you want to specify the fields to be displayed for specific artifacts in the searched results:

```
{
  "searchTerm": "",
  "artifactType": "METADATA",
  "page": {
    "currentPage": 1,
    "chunkSize": 20
  },
  "fields": {
    "METADATA": [
      "shared",
      "businessName",
      "ownerProjectId",
      "entityTypeId",
      "version",
      "zone",
      "projectIds",
      "tableName",
      "entityClusterLocation",
      "modifiedTime",
      "modifiedBy"
    ],
    "DQRULE": [
      "ruleName",
      "modifiedDate",
      "modifiedBy",
      "ruleId"
    ],
    "DQRULESET": [
      "ruleSetName",
      "modifiedDate",
      "modifiedBy",
      "ruleSetId"
    ],
    "WORKFLOW_GS": [
      "shared",
      "wfName",
      "ownerProjectId",
      "modifiedDate",
      "modifiedBy",
      "wfId",

```

```
"projectIds"
],
"TRANSFORMATION": [
  "shared",
  "trMappingName",
  "ownerProjectId",
  "trModifiedDateString",
  "trModifiedBy",
  "projectIds"
],
"NAMESPACE": [
  "key",
  "value",
  "id"
],
"DMEMODELS": [
  "dmModelName",
  "description",
  "brEntityName",
  "status",
  "id"
]
},
"facetFilter": [
]
}
```

Example Response:

```
{
  "responseMessage": "",
  "restUri": "/artifacts/search",
  "result": {
    "metadata": {
      "currentPage": 1,
      "totalRecords": 21,
      "list": {
        "search document object"
      }
    },
    "workflow": {
      "currentPage": 1,
      "totalRecords": 21,
      "list": {
        "search document object"
      }
    },
    "transformation": {
      "currentPage": 1,
      "totalRecords": 21,
      "list": {
        "search document object"
      }
    },
    "dqrule": {
      "currentPage": 1,
      "totalRecords": 21,

```

```
"list": {
  "search document object"
},
"dqruleset": {
  "currentPage": 1,
  "totalRecords": 21,
  "list": {
    "search document object"
  },
  "filepattern": {
    "currentPage": 1,
    "totalRecords": 21,
    "list": [{
      "search document object"
    }],
    "namespace": {
      "currentPage": 1,
      "totalRecords": 21,
      "list": {
        "search document object"
      }
    }
  }
}
```

1.11.3 Services Monitor

1.11.3.1 Fetch services

This API allows you to fetch the details/status about the following services in ZDP:

- *ZDP-EXECUTOR*
- *ZDP-INGESTION-WARDEN*
- *ZDP-LINEAGE*
- *ZDP-SPARK*
- *ZDP_KIBANA*
- *ZDP-HIVE*
- *ZDP-MR*
- *ZDP-GATEWAY*

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/servicemanagement/services/  
↪healthcheck?refresh={true_or_false}
```

Note: You must pass the value for the *refresh* parameter in the API URL. Set the value to *true* to fetch the real-time status of all the services. Alternatively, set the value to *false* to fetch the status of the services based on the last refresh.

Method:

GET

Example Response:

```
{
  "responseMessage": "SUCCESS",
  "restUri": "/servicemanagement/services/healthcheck?refresh={refresh}",
  "result": [
    {
      "id": 776488398,
      "name": "ZDP-GATEWAY",
      "technicalName": "ZDP-GATEWAY",
      "type": "INTERNAL",
      "status": {
        "totalInstances": 1,
        "activeInstances": 1
      },
      "lastStatusUpdateTime": "08/24/2018 17:10:29",
      "processOrder": 0
    },
    {
      "id": -865160315,
      "name": "ZDP-INGESTION-WARDEN",
      "technicalName": "ZDP-INGESTION-WARDEN",
      "type": "INTERNAL",
      "status": {
        "totalInstances": 1,
        "activeInstances": 1
      },
      "lastStatusUpdateTime": "08/24/2018 17:10:29",
      "processOrder": 0
    },
    {
      "id": -570536730,
      "name": "ZDP-HIVE",
      "technicalName": "ZDP-HIVE",
      "type": "INTERNAL",
      "status": {
        "totalInstances": 1,
        "activeInstances": 1
      },
      "lastStatusUpdateTime": "08/24/2018 17:10:29",
      "processOrder": 0
    },
    {
      "id": -1088001273,
      "name": "ZDP-MR",
      "technicalName": "ZDP-MR",
      "type": "INTERNAL",
      "status": {
        "totalInstances": 1,
        "activeInstances": 1
      },
      "lastStatusUpdateTime": "08/24/2018 17:10:29",
      "processOrder": 0
    }
  ]
}
```



```

      "id": 1696097206,
      "name": "ZDP_KIBANA",
      "technicalName": "ZDP_KIBANA",
      "type": "INTERNAL",
      "status": {
        "totalInstances": 1,
        "activeInstances": 1
      },
      "lastStatusUpdateTime": "08/24/2018 17:10:29",
      "processOrder": 0
    },
    {
      "id": 1479068665,
      "name": "ZDP-SPARK",
      "technicalName": "ZDP-SPARK",
      "type": "INTERNAL",
      "status": {
        "totalInstances": 1,
        "activeInstances": 1
      },
      "lastStatusUpdateTime": "08/24/2018 17:10:29",
      "processOrder": 0
    },
    {
      "id": 1784176108,
      "name": "ZDP-EXECUTOR",
      "technicalName": "ZDP-EXECUTOR",
      "type": "INTERNAL",
      "status": {
        "totalInstances": 1,
        "activeInstances": 1
      },
      "lastStatusUpdateTime": "08/24/2018 17:10:29",
      "processOrder": 0
    },
    {
      "id": 1630586505,
      "name": "ZDP-LINEAGE",
      "technicalName": "ZDP-LINEAGE",
      "type": "INTERNAL",
      "status": {
        "totalInstances": 1,
        "activeInstances": 1
      },
      "lastStatusUpdateTime": "08/24/2018 17:10:29",
      "processOrder": 0
    }
  ],
  "page": null
}

```

1.11.3.2 Fetch instance details per service

This API allows you to fetch the number of instances and their details for a specific service in ZDP.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/servicemanagement/services/  
→{service_name}/instances/healthcheck?refresh={true_or_false}
```

Note:

- You can set the value of the *service_name* parameter in the API URL to one of the following:
 - ZDP-EXECUTOR
 - ZDP-INGESTION-WARDEN
 - ZDP-LINEAGE
 - ZDP-SPARK
 - ZDP_KIBANA
 - ZDP-HIVE
 - ZDP-MR
 - ZDP-GATEWAY
- You must pass the value for the *refresh* parameter in the API URL. Set the value to *true* to fetch the real-time status of all the services. Alternatively, set the value to *false* to fetch the status of the services based on the last refresh.

Method:

```
POST
```

Request Payload:

```
{  
  "currentPage": "<An integer value indicating to fetch the 'nth' page based on_  
→the 'chunkSize' size set.>",  
  "chunkSize": "<An integer value that indicates the number of records per page.  
→ When set as '0' (or unspecified), all results are fetched.>",  
  "sortBy": "<Set the attribute value by which the results must be sorted. For_  
→example, 'instanceId'.>",  
  "sortOrder": "<This is the sorting order of the results. probable value can_  
→be 'ASC' or 'DESC'.>"  
}
```

Example Request:

```
{  
  "currentPage": 1,  
  "chunkSize": 10,  
  "sortBy": "instanceId",  
  "sortOrder": "DESC"  
}
```

Example Response:

```
{  
  "responseMessage": "SUCCESS",  
  "restUri": "/servicemanagement/services/{serviceName}/instances/healthcheck?  
→refresh=<true/false>",  
}
```

```
"result": [{
  "instanceId": "zdp-gateway:2534f9038c7c8b63e4d4a782523c2327",
  "status": "Active",
  "fields": {
    "host": "10.11.13.34",
    "zone": "primary",
    "version": "5.0.2-SNAPSHOT",
    "port": "8080",
    "profile": "prod,gateway"
  }
}],
"page": {
  "chunkSize": 20,
  "currentPage": 1,
  "sortBy": "instanceId",
  "sortOrder": "DESC",
  "totalRecords": 0
}
```

Response in Case of Failure

If the *service_name* parameter provided in the *URL* is invalid (or does not exist):

```
{
  "responseMessage": "Invalid serviceName :WORKFLOW_EXECUTOR provided",
  "restUri": "/bedrock-app/services/rest/servicemanagement/services/WORKFLOW_EXECUTOR/instances/healthcheck",
  "result": null,
  "page": null
}
```

1.11.3.3 Fetch details about ActiveMQ, Elasticsearch, and Logstash services

Important: This is a PREVIEW feature.

This API allows you to fetch the details about the following ZDP services:

- *zdp-activemq*
- *zdp-es*
- *zdp-logstash*

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/servicemanagement/services/ZDP-
↳ GATEWAY/instances/{instance_Id}
```

Note: You must specify the value for the *instance_Id* parameter in the API URL. The value for that parameter can be retrieved in the response from the *Fetch instance details per service* API (when the *service_name* parameter is set to *ZDP-GATEWAY*). Example value: *zdp-gateway:add7649b6b17657ac3f811a5df3621b0*.

Method:

GET

Example Response:

```
{
  "status": "UP",
  "zdpServices": {
    "status": "UP",
    "ACTIVE_MQ": {
      "status": "UP",
      "activemq.broker.url": "failover:(tcp://10.11.13.34:61616,tcp://10.11.13.13:61616,
→tcp://10.11.13.31:61616)?jms.prefetchPolicy.all=1"
    },
    "LOGSTASH": {
      "status": "UP",
      "logstash.host": "10.11.13.31",
      "logstash.http.port": 19600
    },
    "ELASTIC_SEARCH": {
      "status": "UP",
      "clusterStatus": "GREEN",
      "clusterName": "zdp_rtp",
      "numberOfNodes": 3,
      "numberOfDataNodes": 3,
      "activePrimaryShards": 56,
      "activeShards": 112,
      "relocatingShards": 0,
      "initializingShards": 0,
      "unassignedShards": 0
    },
    "MAIL_SERVER": {
      "status": "UP",
      "mail.smtp.host": "184.106.200.146",
      "mail.smtp.port": 25
    }
  },
  "discoveryComposite": {
    "description": "Spring Cloud Eureka Discovery Client",
    "status": "UP",
    "discoveryClient": {
      "description": "Spring Cloud Eureka Discovery Client",
      "status": "UP",
      "services": [
        "zdp-dme",
        "zdp-ingestion-warden",
        "zdp-gateway",
        "zdp-spark",
        "zdp-hive",
        "zdp-mr",
        "zdp-executor",
        "zdp_kibana",
        "zdp-lineage"
      ]
    }
  },
  "eureka": {
    "description": "Remote status from Eureka server",
    "status": "UP",
    "applications": {
```

```

    "ZDP-EXECUTOR": 1,
    "ZDP-DME": 1,
    "ZDP-INGESTION-WARDEN": 2,
    "ZDP-LINEAGE": 1,
    "ZDP-SPARK": 1,
    "ZDP_KIBANA": 1,
    "ZDP-HIVE": 1,
    "ZDP-GATEWAY": 2,
    "ZDP-MR": 2
  }
},
"diskSpace": {
  "status": "UP",
  "total": 209822412800,
  "free": 174674984960,
  "threshold": 10485760
},
"db": {
  "status": "UP",
  "database": "MySQL",
  "hello": 1
},
"refreshScope": {
  "status": "UP"
},
"configServer": {
  "status": "UP",
  "propertySources": [
    "file://etc/zdp-registry/zdp-gateway-prod.yml",
    "file://etc/zdp-registry/application.yml"
  ]
},
"hystrix": {
  "status": "UP"
}
}

```

Response in Case of Failure

If you specify an incorrect value for the *instance_Id* parameter:

```

{
  "responseMessage": "No value present",
  "restUri": "/bedrock-app/services/rest/servicemanagement/services/ZDP-GATEWAY/
→instances/zdp-gateway:add7649b6b17657ac3f811a5df3621b1",
  "result": null,
  "page": null
}

```

1.11.4 Capacity Scheduler

1.11.4.1 Configure capacity-scheduler

This API allows you to fetch the capacity-scheduler configuration and determine whether ACL is enabled for queues.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/capacityschedule/  
↪getCapacityConfiguration
```

Method:

```
GET
```

Example Response:

- Scenario 1:

If ACL is enabled for queues:

```
{  
  "result": {  
    "id": 66,  
    "enabled": true,  
    "aclEnabled": true,  
    "capacitySchedulerConfigurationFile": "/projects/bedrock/dev/  
↪capacity-scheduler.xml",  
    "autoRefreshEnabled": true,  
    "autoRefreshInterval": 5,  
    "autoRefreshIntervalUnits": "Days"  
  },  
  "status": {  
    "responseCode": 200,  
    "result": null,  
    "responseMessage": "Success",  
    "responseType": "info"  
  }  
}
```

- Scenario 2:

If ACL is disabled:

```
{  
  "result": {  
    "autoRefreshInterval": 0,  
    "autoRefreshIntervalUnits": null,  
    "capacitySchedulerConfigurationFile": null,  
    "autoRefreshEnabled": false,  
    "aclEnabled": false,  
    "enabled": false,  
    "id": 0  
  },  
  "status": {  
    "responseType": "info",  
    "responseMessage": "Success",  
    "result": null,  
    "responseCode": 200  
  }  
}
```

1.11.4.2 Fetch capacity-scheduler queues

This API allows you to fetch the capacity-scheduler queues assigned to a user.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/capacityschedule/getCapacityQueue
```

Method:

```
GET
```

Example Response:

- Scenario 1:

If Capacity-Queues are configured:

```
{
  "result": [
    {
      "queueName": "q1",
      "queueId": 172,
      "queueCapacity": 30,
      "parentQueueId": 0,
      "queueElasticity": 0
    },
    {
      "queueName": "q2",
      "queueId": 173,
      "queueCapacity": 50,
      "parentQueueId": 0,
      "queueElasticity": 0
    },
    {
      "queueName": "default",
      "queueId": 174,
      "queueCapacity": 20,
      "parentQueueId": 0,
      "queueElasticity": 0
    },
    {
      "queueName": "q1.q3",
      "queueId": 175,
      "queueCapacity": 30,
      "parentQueueId": 172,
      "queueElasticity": 0
    },
    {
      "queueName": "q1.q4",
      "queueId": 176,
      "queueCapacity": 70,
      "parentQueueId": 172,
      "queueElasticity": 0
    }
  ],
  "status": {
    "responseCode": 200,
    "result": null,
    "responseMessage": "Success",
    "responseType": "info"
  }
}
```

- Scenario 2:

If no Capacity-Queues are configured:

```
{
  "result": [ ],
  "status": {
    "responseType": "info",
    "responseMessage": "Success",
    "result": null,
    "responseCode": 200
  }
}
```

1.11.5 Export-Import

ZDP Export-Import APIs serve to export (or migrate) resources from a ZDP server instance (or source) and import those into a target ZDP instance. Export-Import identifies ZDP resources (that need to be migrated) based on the module name defined in the payload. The following module names are standardised for ZDP Export-Import:

- *dataQuality*
- *metadata*
- *transformation*
- *dataInventory*
- *workflow*
- *filePattern*
- *lzDirectory*
- *flumeAgent*
- *notificationTemplate*
- *users*
- *distributionList*
- *bedrockNamespace*
- *systemConfiguration*

When exporting ZDP resources, if the flag: *dependencyRequired* is set to *true*, all dependencies (across ZDP modules) associated with the selected resource are exported automatically. Additionally, if the *externalResourceRequired* flag is set to *true*, the system exports the related external files associated with the selected resource(s).

For instance, if a workflow is exported and these flags are set to *true*, all dependencies across the ZDP modules, such as metadata entity(s), data quality (rules, rule sets), transformation, distribution list(s), ZDP user(s); along with the related external files, such as Shell script, Jar file, etc are bundled and exported.

1.11.5.1 Export ZDP resources

This API allows you to generate an export (.tar) file that bundles the ZDP resources from the source instance. The tar file stores the definition of the resources that needs to be exported and the export filename follows the format:

```
EI-<export-import-Id>:<moduleName>-<exportfileId>.tar
```


URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/export/projects/{source_project_
↪Id}/{module_name}
```

Method:

```
POST
```

Example Request:

- Scenario 1:

To export ZDP system variable(s) by using *moduleName* - *systemConfiguration*:

```
{
  "systemConfigList": [
    {
      "key": "ADMIN_EMAIL_ADDRESS"
    },
    {
      "key": "APP_LOGS"
    }
  ]
}
```

- Scenario 2:

To export ZDP users by using *moduleName* - *users*.

```
{
  "userList": [
    {
      "userId": "user1"
    },
    {
      "userId": "user2"
    }
  ]
}
```

- Scenario 3:

To export ZDP notification template(s) by using *moduleName* - *notificationTemplate*:

```
{
  "notificationTemplateList": [
    {
      "templateName": "Notification template1"
    },
    {
      "templateName": "Notification template2"
    }
  ],
  "dependencyRequired": "true"
}
```

- Scenario 4:

To export ZDP distribution list(s) by using *moduleName* - *distributionList*:

```
{
  "distributionList": [
    {
      "distributionListName": "distributionList1"
    },
    {
      "distributionListName": "distributionList2"
    }
  ],
  "dependencyRequired": "true"
}
```

- Scenario 5:

To export file patterns by using *moduleName* - *filePattern*:

```
{
  "filePatternList": [
    {
      "patternPrefix": "sid_[0-9]+",
      "patternSuffix": ".dat",
      "fileSystem": "HDFS"
    },
    {
      "patternPrefix": "[0-9]*",
      "patternSuffix": ".txt",
      "fileSystem": "LOCAL"
    }
  ],
  "dependencyRequired": "true",
  "externalResourceRequired": "true"
}
```

- Scenario 6:

To export data inventory profiles by using *moduleName* - *dataInventory*:

```
{
  "dataInventoryProfileList": [
    {
      "profileName": "profile1"
    },
    {
      "profileName": "profile2"
    }
  ]
}
```

- Scenario 7:

To export entity types by using *moduleName* - *metadata*:

```
{
  "entityList": [
    {
      "context": [
        {
          "partName": "sourcePlatform",
          "partValue": "b_employee_text"
        }
      ]
    }
  ]
}
```

```

        },
        {
            "partName": "sourceSchema",
            "partValue": "b_employee_text"
        }
    ],
    "technicalName": "b_employee_text",
    "dataStore": "HDFS"
},
{
    "context": [
        {
            "partName": "sourcePlatform",
            "partValue": "Sasp_api"
        },
        {
            "partName": "sourceSchema",
            "partValue": "Sass_api"
        }
    ],
    "technicalName": "Saas_api",
    "dataStore": "HDFS"
}
],
"dependencyRequired": "true",
"externalResourceRequired": "true"
}

```

- Scenario 8:

To export multiple ZDP namespace variables by using *moduleName* - *bedrockNamespace*:

```

{
    "bedrockNamespaceList": [
        {
            "key": "Namespace1"
        },
        {
            "key": "Namespace2"
        }
    ]
}

```

- Scenario 9:

To export multiple workflows by using *moduleName* - *workflow*:

```

{
    "workflowList": [
        {
            "workflowName": "workflowName1"
        },
        {
            "workflowName": "workflowName2"
        }
    ],
    "dependencyRequired": "true",
    "externalResourceRequired": "true"
}

```

- Scenario 10:

To export transformations by using *moduleName - transformation*:

```
{
  "transformationList": [
    {
      "transformationName": "transformationName1"
    },
    {
      "transformationName": "transformationName2"
    }
  ],
  "dependencyRequired": "true",
  "externalResourceRequired": "true"
}
```

- Scenario 11:

To export Flume agents by using *moduleName - flumeAgent*:

```
{
  "flumeAgentList": [
    {
      "agentName": "agent1"
    },
    {
      "agentName": "agent2"
    }
  ],
  "dependencyRequired": "true",
  "externalResourceRequired": "true"
}
```

- Scenario 12:

To export landing zone or source directories by using *moduleName - lzDirectory*:

```
{
  "lzDirectoryList": {
    "serverIp": "10.11.12.117",
    "connectionInstanceName": "Example Connection_HDFS",
    "directoryPathList": [
      {
        "directoryPath": "/home/bedrock/dir1"
      },
      {
        "directoryPath": "/home/bedrock/dir2"
      }
    ]
  },
  "dependencyRequired": "true",
  "externalResourceRequired": "false"
}
```

- Scenario 13:

To export data quality rules/rule sets by using *moduleName - dataQuality*:

```
{
  "ruleList": [
    {
      "type": "rule",
      "name": "not_null"
    },
    {
      "type": "rule set",
      "name": "pattern_check"
    }
  ]
}
```

Example Response:

```
{
  "responseMessage": "Data exported successfully",
  "restUri": null,
  "result": "EI-20151208181132871:<exportModuleName>-20151208181132872.tar"
}
```

1.11.5.2 Download export file

This API allows you to download the export (*tar*) file into the local system. The exported filename is the one as displayed against the *result* attribute under the **Example Response** of the *Export ZDP resources* API.

URL:

```
http://<bedrock-host>:<port>/bedrock-app/services/rest/export/bundle/{export_filename}
```

Example URL:

```
http://10.11.12.117:9090/bedrock-app/services/rest/export/bundle/EI-
↪ 20151208134534997:systemConfiguration-20151208134534997.tar
```

The export file is required later to import resources into the target ZDP instance by the use of the *Import ZDP resources* API.

1.11.5.3 Import ZDP resources

This API allows you to import resources by using the bundled file (*.tar*) downloaded earlier to a project that exists within the same or different ZDP instance.

URL:

```
http://<bedrock-host>:port/bedrock-app/services/rest/projects/{target_project_Id}/
↪ import
```

Method:

```
POST
```

Parameter Description:

The parameter type must be of form-data with the key-values as below:

Key	Value
<i>file</i>	Choose the export (.tar) file from the local system.
<i>updateAllConflicts</i>	Indicate either <i>True</i> or <i>False</i> to update or ignore all the conflicting records.
<i>createNewRevision</i>	Indicate if you want to create a new revision for an entity. This parameter applies only to metadata. The default value is <i>true</i> , indicating a new revision of the entity is created. If set to <i>false</i> , the latest version of the entity is updated.
<i>serverIp</i>	Indicate the server IP address. This parameter is required only when you want to import landing zone directory and Flume agent. You can import only one server's source directory or Flume agent in one request.

Example Request:

Key	Value
<i>file</i>	metadata-20160209223451693.tar
<i>updateAllConflicts</i>	true
<i>createNewRevision</i>	true
<i>serverIp</i>	-

Example Response:

```
{
  "responseMessage": "Import process completed",
  "restUri": null,
  "result": [
    "File metadata-20160209223451693.tar contains : 3 metadata",
    "Number of existing entity(s): 2",
    "Successfully added: 1 entity(s)",
    "Successfully updated with new revision: 1 entity(s)",
    "Number of records ignored as no change detected : 1 entity(s)",
    "Log file created at: <APP_LOGS path>/IMPORT_LOG/metadata-
    ↪20160209223451693_20160210120839587.txt",
    "Log file name: metadata-20160209223451693_20160210120839587.txt"
  ]
}
```

1.11.5.4 Download import log file

This API allows you to download the import logs into the local system, after importing resources into the target ZDP instance.

URL:

```
http://<bedrock-host-target>:<port>/bedrock-app/services/rest/import/log/{import_log_
    ↪file}
```

Example URL:

```
http://10.12.13.116:6060/bedrock-app/services/rest/import/log/metadata-
    ↪20160209035739329_20160209173248647.txt
```

1.11.6 Data Lifecycle Management

1.11.6.1 Add DLM policy

This API allows you to add a DLM policy.

URL:

```
http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/policies
```

Method:

```
POST
```

Request Payload:

```
{
  "createdBy": "<This is the user who is creating the policy.>",
  "createdTime": "<This is the time when you are creating the policy.>",
  "modifiedBy": "<This is the user who is modifying the policy.>",
  "modifiedTime": "<This is the time when you are modifying the policy.>",
  "policyId": "<This is the Id of the policy>",
  * "name": "<This is the name of the policy.>",
  "description": "<This is the description of the policy.>",
  "defaultPolicy": "<This indicates if the policy is the default policy. Set it to_
  ↳ true or false.>",
  * "policyRules": [
    {
      "ruleId": "<This is the Id of the rule.>",
      "order": "<This is the order of the rule.>",
      "state": "<This is the state of the rule, such as Hot.>",
      "storageState": "<This is the state of the storage.>",
      "stateId": "<This is the Id of the state.>",
      "age": "<This is the number of days that a file remains in that state.>"
    }
  ],
  "vaultConnectionInstanceId": "<This is the connection instance Id for the vault.>
  ↳ ",
  "vaultConnectionInstanceName": "<This is the connection name for the vault.>"
}
```

Example Request:

```
{
  "name": "p4",
  "description": "",
  "defaultPolicy": false,
  "vaultConnectionInstanceId": 1,
  "policyId": 12,
  "policyRules": [
    {
      "ruleId": 0,
      "order": 1,
      "state": "All_SSD",
      "age": 30
    },
    {
      "ruleId": 0,
```

```
    "order": 2,
    "state": "Hot",
    "age": 60
  },
  {
    "ruleId": 0,
    "order": 3,
    "state": "S3_Vault",
    "age": null
  }
]
```

Example Response:

```
{
  "responseMessage": "Policy saved successfully",
  "restUri": null,
  "result": {
    "id": 1,
    "status": "SUCCESS"
  },
  "page": null
}
```

Response in Case of Failure:

```
{
  "responseMessage": "Could not read document: Unrecognized token 'dccc': was expecting ('true', 'false' or 'null')\n at [Source: java.io.PushbackInputStream@57515ce1; line: 2, column: 21]; nested exception is com.fasterxml.jackson.core.JsonParseException: Unrecognized token 'dccc': was expecting ('true', 'false' or 'null')\n at [Source: java.io.PushbackInputStream@57515ce1; line: 2, column: 21]",
  "restUri": "/bedrock-app/services/rest/govern/dlm/policies",
  "result": null,
  "page": null
}
```

1.11.6.2 Fetch DLM policies

This API allows you to fetch DLM policies that are associated with a policy Id.

URL:

```
http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/policies/
↳ {policy_Id}
```

Note: You must pass the *policy_Id* parameter in the API URL.

Method:

GET

Example Response:


```
{
  "responseMessage": null,
  "restUri": null,
  "result": {
    "createdBy": "admin",
    "createdTime": "06/17/2016 08:21:32",
    "modifiedBy": "admin",
    "modifiedTime": "06/17/2016 08:21:32",
    "policyId": 1,
    "name": "GLOBAL POLICY",
    "description": "Default",
    "defaultPolicy": true,
    "policyRules": [
      {
        "ruleId": 1,
        "order": 1,
        "state": "Hot",
        "storageState": "HOT",
        "stateId": 1,
        "age": 30
      },
      {
        "ruleId": 2,
        "order": 2,
        "state": "Warm",
        "storageState": "WARM",
        "stateId": 2,
        "age": 90
      },
      {
        "ruleId": 3,
        "order": 3,
        "state": "Cold",
        "storageState": "COLD",
        "stateId": 3,
        "age": null
      }
    ],
    "vaultConnectionInstanceId": null,
    "vaultConnectionInstanceName": null
  },
  "page": null
}
```

Response in Case of Failure:

```
{
  "responseMessage": "Failed to fetch policy definition for policyID :123",
  "restUri": "/bedrock-app/services/rest/govern/dlm/policies/123",
  "result": null,
  "page": null
}
```

1.11.6.3 Modify DLM policy

This API allows you to modify a DLM policy associated with a policy Id.

URL:

```
http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/policies/  
→{policy_Id}
```

Note: You must pass the *policy_Id* parameter in the API URL.

Method:

PUT

Request Payload:

```
{  
  {  
    "createdBy": "<This is the user who has created the policy.>",  
    "createdTime": "<This is the time when the policy was created.>",  
    "modifiedBy": "<This is the user who is modifying the policy.>",  
    "modifiedTime": "<This is the time when you are modifying the policy.>",  
    * "policyId": "<This is the Id of the policy>",  
    "name": "<This is the name of the policy.>",  
    "description": "<This is the description of the policy.>",  
    "defaultPolicy": "<This indicates if the policy is the default policy. Set it to_  
→true or false.>",  
    "policyRules": [  
      {  
        "ruleId": "<This is the Id of the rule.>",  
        "order": "<This is the order of the rule.>",  
        "state": "<This is the state of the rule, such as Hot.>",  
        "storageState": "<This is the state of the storage.>",  
        "stateId": "<This is the Id of the state.>",  
        "age": "<This is the number of days that a file remains in that state.>"  
      }  
    ],  
    "vaultConnectionInstanceId": "<This is the connection instance Id for the vault.>",  
    "vaultConnectionInstanceName": "<This is the connection name for the vault.>"  
  }  
}
```

Example Payload:

```
{  
  "policyId": 1,  
  "name": "Test Policy Name",  
  "description": "Test Policy Description",  
  "defaultPolicy": true,  
  "policyRules": [  
    {  
      "ruleId": 1,  
      "order": 1,  
      "state": "Hot",  
      "age": 30  
    },  
    {  
      "ruleId": 2,  
      "order": 2,  
      "state": "Warm",  
      "age": 30  
    }  
  ]  
}
```

```

        "age": 60
      },
      {
        "ruleId": 3,
        "order": 3,
        "state": "Cold",
        "age": 0
      }
    ]
  }
}

```

Example Response:

```

{
  "responseMessage": "Policy updated successfully",
  "restUri": null,
  "result": {
    "id": 1,
    "status": "SUCCESS"
  },
  "page": null
}

```

Response in Case of Failure:

```

{
  "responseMessage": "Could not read document: Failed to parse Date value '2016-06-20T13:41:48.685Z' (format: \"MM/dd/yyyy HH:mm:ss\"): Unparseable date: \"2016-06-20T13:41:48.685Z\" (through reference chain: com.zaloni.bedrock.dlm.vo.PolicyDefinitionVO[\"createdTime\"]); nested exception is com.fasterxml.jackson.databind.JsonMappingException: Failed to parse Date value '2016-06-20T13:41:48.685Z' (format: \"MM/dd/yyyy HH:mm:ss\"): Unparseable date: \"2016-06-20T13:41:48.685Z\" (through reference chain: com.zaloni.bedrock.dlm.vo.PolicyDefinitionVO[\"createdTime\"])",
  "restUri": "/bedrock-app/services/rest/govern/dlm/policies/%7BpolicyId%7D",
  "result": null,
  "page": null
}

```

1.11.6.4 Delete DLM policy

This API allows you to delete a DLM policy associated with a policy Id.

URL:

```

http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/policies/
↳ {policy_Id}

```

Note: You must pass the *policy_Id* parameter in the API URL.

Method:

```
DELETE
```

Example Response:

```
{
  "responseMessage": "Policy deleted successfully",
  "restUri": null,
  "result": null,
  "page": null
}
```

Response in Case of Failure:

```
{
  "responseMessage": "Cannot delete the default policy",
  "restUri": "/bedrock-app/services/rest/govern/dlm/policies/1",
  "result": null,
  "page": null
}
```

1.11.6.5 Fetch DLM policy states

This API allows you to fetch the DLM policy states.

URL:

```
http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/policies/
↪states
```

Method:

```
GET
```

Example Response:

```
{
  "responseMessage": null,
  "restUri": null,
  "result": [
    {
      "stateId": 1,
      "name": "Hot",
      "mappedDLMState": "HOT",
      "stateOrder": 3,
      "stateType": "STORAGE"
    },
    {
      "stateId": 2,
      "name": "Warm",
      "mappedDLMState": "WARM",
      "stateOrder": 4,
      "stateType": "STORAGE"
    },
    {
      "stateId": 3,
      "name": "Cold",
      "mappedDLMState": "COLD",
      "stateOrder": 5,
      "stateType": "STORAGE"
    },
    {

```

```

    "stateId": 4,
    "name": "All_SSD",
    "mappedDLMState": "ALL_SSD",
    "stateOrder": 1,
    "stateType": "STORAGE"
  },
  {
    "stateId": 5,
    "name": "One_SSD",
    "mappedDLMState": "ONE_SSD",
    "stateOrder": 2,
    "stateType": "STORAGE"
  },
  {
    "stateId": 6,
    "name": "Lazy_Persist",
    "mappedDLMState": "LAZY_PERSIST",
    "stateOrder": 3,
    "stateType": "STORAGE"
  },
  {
    "stateId": 7,
    "name": "S3_Vault",
    "mappedDLMState": "S3_VAULT",
    "stateOrder": 7,
    "stateType": "ARCHIVE"
  }
],
"page": null
}

```

Response in Case of Failure:

```

{
  "responseMessage": "DLM is required to be enabled",
  "restUri": "/bedrock-app/services/rest/govern/dlm/policies/states",
  "result": null,
  "page": null
}

```

1.11.6.6 Fetch DLM policy age evaluators

This API allows you to fetch the DLM policy age evaluators.

URL:

```

http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/policies/
↪evaluators

```

Method:

```
GET
```

Example Response:

```

{
  "responseMessage": null,

```

```

"restUri": null,
"result": [
  {
    "pluginId": "DLM_c0297ae6-8157-40b3-b9f8-2d1d880be993",
    "pluginName": "file_creation_date",
    "pluginType": null,
    "pluginClass": "com.zaloni.bedrock.evaluator.FileCreationDateBasedEvaluator
↪",
    "pluginFile": "bedrock-dlm-default-evaluator-4.2.1-SNAPSHOT.jar"
  }
],
"page": null
}

```

1.11.6.7 Fetch list of policy definitions

This API allows you to fetch the list of policy definitions.

URL:

```

http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/policies/
↪search

```

Method:

POST

Request Payload:

```

{
  "chunkSize": "<Indicates the pagination size. When set as '0' (or unspecified), all_
↪results are fetched.>",
  "currentPage": "<This is the page number of the current policy.>",
  "orderBy": "<This is the criterion by which the list of policies are ordered.>",
  "sortBy": "<This is the criterion by which the list of policies are sorted.>",
  "searchCriteria": [
    {
      "joinAttribute": "<This is the join attribute.>",
      "searchAttribute": "<This is the search attribute.>",
      "searchValue": "<This is the search value.>",
      "dataType": "<This is tha data type.>",
      "searchMode": "<This is the search mode.>",
      "rangeStart": "<This is the start date.>",
      "rangeEnd": "<This is the end date.>"
    }
  ],
  "artifactIds": [
    0
  ]
}

```

Example Request:

```

{
  "chunkSize": 20,
  "currentPage": 1,
  "orderBy": null,

```

```

    "sortBy": null,
    "searchCriteria": [
      {
        "joinAttribute": null,
        "searchAttribute": "name",
        "searchValue": "test1",
        "dataType": null,
        "searchMode": null,
        "rangeStart": null,
        "rangeEnd": null
      }
    ]
  }
}

```

Example Response:

```

{
  "responseMessage": null,
  "restUri": null,
  "result": {
    "resultList": [
      {
        "createdBy": "admin",
        "createdTime": "06/17/2016 08:49:16",
        "modifiedBy": "admin",
        "modifiedTime": "06/17/2016 08:49:16",
        "policyId": 2,
        "name": "Test",
        "description": "",
        "defaultPolicy": false,
        "policyRules": [],
        "vaultConnectionInstanceId": null,
        "vaultConnectionInstanceName": null
      },
      {
        "createdBy": "admin",
        "createdTime": "06/17/2016 08:21:32",
        "modifiedBy": "admin",
        "modifiedTime": "06/17/2016 08:21:32",
        "policyId": 1,
        "name": "GLOBAL POLICY",
        "description": "Default",
        "defaultPolicy": true,
        "policyRules": [
          {
            "ruleId": 1,
            "order": 1,
            "state": "Hot",
            "storageState": "HOT",
            "stateId": 1,
            "age": 30
          },
          {
            "ruleId": 2,
            "order": 2,
            "state": "Warm",
            "storageState": "WARM",
            "stateId": 2,

```

```

        "age": 90
      },
      {
        "ruleId": 3,
        "order": 3,
        "state": "Cold",
        "storageState": "COLD",
        "stateId": 3,
        "age": null
      }
    ],
    "vaultConnectionInstanceId": null,
    "vaultConnectionInstanceName": null
  }
],
"totalRecords": 2,
"currentPage": 0,
"chunkSize": 0
},
"page": null
}

```

Response in Case of Failure:

```

{
  "responseMessage": "No Policy Definition records found",
  "restUri": null,
  "result": null,
  "page": null
}

```

1.11.6.8 Execute DLM process

This API allows you to execute the DLM process on demand.

URL:

```
http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/enforcement
```

Method:

```
GET
```

Example Response:

```

{
  "responseMessage": "DLM Execution request submitted successfully",
  "restUri": null,
  "result": "SUCCESS",
  "page": null
}

```

Response in Case of Failure:

```

{
  "responseMessage": "DLM enforcement process failed.Required microservices not_
↪registered",

```



```
{
  "restUri": "/bedrock-app/services/rest/govern/dlm/enforcement",
  "result": null,
  "page": null
}
```

1.11.6.9 Fetch list of DLM execution instances

This API allows you to fetch the list of DLM execution instances.

URL:

```
http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/enforcement/
↪history
```

Method:

```
POST
```

Request Payload:

```
{
  "chunkSize": "<This is the number of results that must be fetched in each page.↪
↪When set as '0' (or unspecified), all results are fetched.>",
  "currentPage": "<This is the page number of the current policy.>",
  "orderBy": "<This is the criterion by which the list of policies are ordered.>",
  "sortBy": "<This is the criterion by which the list of policies are sorted.>",
  "searchCriteria": [
    {
      "joinAttribute": "<This is the join attribute.>",
      "searchAttribute": "<This is the search attribute.>",
      "searchValue": "<This is the search value.>",
      "dataType": "<This is the data type.>",
      "searchMode": "<This is the search mode.>",
      "rangeStart": "<This is the start date.>",
      "rangeEnd": "<This is the end date.>"
    }
  ],
  "artifactIds": [
    0
  ]
}
```

Example Request:

```
{
  "chunkSize": 0,
  "currentPage": 0,
  "orderBy": "string",
  "sortBy": "string",
  "searchCriteria": [
    {
      "joinAttribute": "string",
      "searchAttribute": "string",
      "searchValue": "string",
      "dataType": "string",
      "searchMode": "string",
      "rangeStart": "string",

```

```

        "rangeEnd": "string"
      }
    ],
    "artifactIds": [
      0
    ]
  }
}

```

Example Response:

```

{
  "responseMessage": null,
  "restUri": null,
  "result": {
    "resultList": [
      {
        "executionInstanceId": 1,
        "executionStartTime": "06/20/2016 07:37:02",
        "executedBy": "admin",
        "executionEndTime": "06/20/2016 07:37:21",
        "status": "FAILURE",
        "noOfFilesProcessed": 0
      }
    ],
    "totalRecords": 1,
    "currentPage": 0,
    "chunkSize": 0
  },
  "page": null
}

```

Response in Case of Failure:

```

{
  "responseMessage": "could not resolve property: string of: com.zaloni.bedrock.dlm.
↪entity.DLMExecutionLog",
  "restUri": "/bedrock-app/services/rest/govern/dlm/enforcement/history",
  "result": null,
  "page": null
}

```

1.11.6.10 Fetch list of DLM policy association

This API allows you to fetch the list of DLM policy associations.

URL:

```
http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/associations
```

Method:

```
GET
```

Example Response:

```

{
  "responseMessage": null,

```

```

"restUri": null,
"result": [
  {
    "associationId": 1,
    "policyId": 1,
    "policyName": "GLOBAL POLICY",
    "associatedResource": "ENTITY_TYPE",
    "evaluatorName": "file_creation_date",
    "evaluatorClass": "com.zaloni.bedrock.evaluator.
↪FileCreationDateBasedEvaluator",
    "evaluatorFile": "bedrock-dlm-default-evaluator-4.2.1-SNAPSHOT.jar",
    "entityTypeId": 3,
    "entityTypeVersion": 1,
    "associationType": "FILE",
    "dataLocation": null
  },
  {
    "associationId": 3,
    "policyId": 1,
    "policyName": "GLOBAL POLICY",
    "associatedResource": "ENTITY_TYPE",
    "evaluatorName": "file_creation_date",
    "evaluatorClass": "com.zaloni.bedrock.evaluator.
↪FileCreationDateBasedEvaluator",
    "evaluatorFile": "bedrock-dlm-default-evaluator-4.2.1-SNAPSHOT.jar",
    "entityTypeId": 5,
    "entityTypeVersion": 1,
    "associationType": "FILE",
    "dataLocation": null
  }
],
"page": null
}

```

Response in Case of Failure:

```

{
  "responseMessage": "DLM is required to be enabled",
  "restUri": "/bedrock-app/services/rest/govern/dlm/associations",
  "result": null,
  "page": null
}

```

1.11.6.11 Add a DLM policy association

This API allows you to add a DLM policy associations.

URL:

```
http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/associations
```

Method:

```
POST
```

Request Payload:

```
{
  "associationId": "<This is the Id of the association.>",
  "policyId": "<This is the Id of the policy.>",
  "policyName": "<This is the name of the policy.>",
  "associatedResource": "<This is the associated resource.>",
  "evaluatorName": "<This is the name of the evaluator.>",
  "evaluatorClass": "<This is the class of the evaluator.>",
  "evaluatorFile": "<This is the evaluator file.>",
  "entityTypeId": "<This is the Id of the entity type.>",
  "entityTypeVersion": "<This is the version of the entity type.>",
  "associationType": "<This is the type of association.>",
  "dataLocation": "<This is the location of the data.>"
}
```

Example Request:

```
{
  "associationId": 0,
  "policyId": 1,
  "policyName": "string",
  "associatedResource": "string",
  "evaluatorName": "string",
  "evaluatorClass": "string",
  "evaluatorFile": "string",
  "entityTypeId": 12,
  "entityTypeVersion": 1,
  "associationType": "FILE",
  "dataLocation": "string"
}
```

Example Response:

```
{
  "responseMessage": "Policy associated successfully",
  "restUri": null,
  "result": null,
  "page": null
}
```

Response in Case of Failure:

```
{
  "responseMessage": "Entity type not found for id :5 and version :2",
  "restUri": "/bedrock-app/services/rest/govern/dlm/associations",
  "result": null,
  "page": null
}
```

1.11.6.12 Fetch DLM policy association

This API allows you to fetch the list of DLM policy association by using the details about the associated entity, such as the Id and the version of the entity.

URL:

```
http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/associations/
↳ {entity_Id}/{entity_version}
```

Note: You must pass the *entity_Id* and *entity_version* parameters in the API URL.

Method:

GET

Example Response:

```
{
  "responseMessage": null,
  "restUri": null,
  "result": {
    "associationId": 1,
    "policyId": 1,
    "policyName": "GLOBAL POLICY",
    "associatedResource": "ENTITY_TYPE",
    "evaluatorName": null,
    "evaluatorClass": null,
    "evaluatorFile": null,
    "entityTypeId": 53,
    "entityTypeVersion": 1,
    "associationType": "File",
    "dataLocation": null
  },
  "page": null
}
```

Response in Case of Failure:

```
{
  "responseMessage": "Entity is not associated with any policy",
  "restUri": "/bedrock-app/services/rest/govern/dlm/associations/1/1",
  "result": null,
  "page": null
}
```

1.11.6.13 Update DLM policy association

This API allows you to update the list of DLM policy association.

URL:

```
http://<bedrock-host-target>:<port>/bedrock-app/services/rest/govern/dlm/associations/
↪{association_Id}
```

Note: You must pass the *association_Id* parameter in the API URL.

Method:

PUT

Request Payload:

```
{
  "associationId": "<This is the Id of the association.>",
  "policyId": "<This is the Id of the policy.>",
  "policyName": "<This is the name of the policy.>",
  "associatedResource": "<This is the associated resource.>",
  "evaluatorName": "<This is the name of the evaluator.>",
  "evaluatorClass": "<This is the class of the evaluator.>",
  "evaluatorFile": "<This is the evaluator file.>",
  "entityTypeId": "<This is the Id of the entity type.>",
  "entityTypeVersion": "<This is the version of the entity type.>",
  "associationType": "<This is the type of association.>",
  "dataLocation": "<This is the location of the data.>"
}
```

Example Request:

```
{
  "associationId": 0,
  "policyId": 0,
  "policyName": "string",
  "associatedResource": "string",
  "evaluatorName": "string",
  "evaluatorClass": "string",
  "evaluatorFile": "string",
  "entityTypeId": 0,
  "entityTypeVersion": 0,
  "associationType": "string",
  "dataLocation": "string"
}
```

Example Response:

```
{
  "responseMessage": null,
  "restUri": null,
  "result": {
    "resultList": [
      {
        "executionInstanceId": 1,
        "executionStartTime": "06/20/2016 07:37:02",
        "executedBy": "admin",
        "executionEndTime": "06/20/2016 07:37:21",
        "status": "FAILURE",
        "noOfFilesProcessed": 0
      }
    ],
    "totalRecords": 1,
    "currentPage": 0,
    "chunkSize": 0
  },
  "page": null
}
```

Response in Case of Failure:

```
{
  "responseMessage": "Failed to convert value of type 'java.lang.String' to required_
  ↪type 'java.lang.Integer'; nested exception is java.lang.NumberFormatException: For_
  ↪input string: \"{associationId}\"",
}
```

```
"restUri": "/bedrock-app/services/rest/govern/dlm/associations/%7BassociationId%7D",
"result": null,
"page": null
}
```

1.12 Notification API

By using the notification API, emails in the form of a predefined template can be sent to email addresses predefined in the distribution lists. The following must be ensured for the notification API to work successfully:

- Distribution lists must be configured in the ZDP platform. For more information on mailing lists, refer to the `notification_view` section.
- An email server must be configured in the ZDP platform. For more information on configuring servers, refer to the `system_view` section.

The following API can be invoked for saving and sending a notification email to a pre-configured distribution list in a predefined template with dynamic variable substitution. This registers a notification with the ZDP platform and sends the notification email instantaneously.

URL:

```
http://<bedrock-host:port>/bedrock-app/services/rest/notification/
↪saveNotificationDetails
```

Method:

```
POST
```

Request Payload:

```
{
  "severity": "<This is the name of distribution list defined in the ZDP application.↪
↪(This variable has been deprecated).>",
  *"distributionListName": "<This is the name of distribution list defined in the ZDP↪
↪application.>",
  *"subMsg": "<This is the email subject line. (This variable has been deprecated).> ",
  *"bodyMsg": "<This is the body of the email. (This variable has been deprecated).> ",
  "templateName": "<This is the name of the email template designed in the ZDP↪
↪application. (If this variable is used, the values used in subMsg and bodyMsg are↪
↪overridden with the template subject and body respectively).>",
  "templateVariableMap": {
    "<This is the variable key used in the template.>": "<This is the variable value↪
↪which needs to be replaced for the key.>"
  }
}
```

Example Request:

```
{
  "severity": "test distribution list name",
  "subMsg": "test subject",
  "bodyMsg": "test body",
}
```

OR

```
{
  "distributionListName": "test distribution list name",
  "subMsg": "test subject",
  "bodyMsg": "test body",
  "templateName": "my_template" (Note : values in subMsg and bodyMsg will be
  ↳ overridden by the subject and body respectively defined in my_template),
  "templateVariableMap": {
    "dummyText": "mail for testing",
    "ip": "127.0.0.1",
    "user": "admin"
  }
}
```

Maximum Length of Fields:

```
subMsg: 400 characters
bodyMsg: 65,535 characters
```

Example Response:

```
{
  "result": "Notification mail request submitted successfully",
  "status": {
    "responseCode": 200,
    "result": null,
    "responseMessage": "Success",
    "responseType": "info"
  }
}
```

Response in Case of Failure:

- Scenario 1:

If invalid distribution list name is provided:

```
{
  "result": "",
  "status": {
    "responseCode": 500,
    "result": null,
    "responseMessage": "Invalid distribution list name 'invalidDistributionList'",
    "responseType": "ERROR"
  }
}
```

- Scenario 2:

If distribution list name is not provided or it does not contain any email Ids:

```
{
  "result": "",
  "status": {
    "responseCode": 500,
    "result": null,
    "responseMessage": "Distribution list empty or does not contain any email ids",
    "responseType": "ERROR"
  }
}
```


- Scenario 3:

If invalid template name is provided:

```
{
  "result": "",
  "status": {
    "responseCode": 500,
    "result": null,
    "responseMessage": "Invalid mail template name 'invalid_template'",
    "responseType": "ERROR"
  }
}
```

1.13 Deprecated API

The following APIs are deprecated in ZDP. While these deprecated APIs are available for usage, it is possible these may be dropped in subsequent releases of ZDP.

Therefore, the recommendation is to leverage the new APIs that correspond to each deprecated service (listed below along with the Bedrock/ZDP version from which they are deprecated).

```
http://<bedrock-host>:port/bedrock-app/services/rest/admin/getUserRole (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/getWorkFlows (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/instances/{instance_Id}/
↳{instance_Version} (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/deleteWorkflows (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/executeWorkflow (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/publish/
↳getWorkflowExecutionHistory (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/fetchWorkflowStatus/
↳{instance_Id} (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/publish/
↳restartWorkflowInstance (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/changeWorkflowStatus (4.
↳3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/
↳changeWorkflowStepStatus (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedule/getSchedules (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedule/addWorkFlowSchedule (4.
↳3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedule/{schedule_instance_Id}/
↳{action} (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/schedule/workflowUnschedule (4.3.
↳0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/exportWorkflowJson (4.3.
↳0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/importWorkflows (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/transformation/
↳searchMappingsOverview (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/publish/
↳saveFilePattern (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/publish/getFileIndex_
↳ (4.4.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/publish/
↳searchFilePatterns (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/ingestion/publish/getFilePattern/
↳{File_pattern_Id} (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/flume/publish/saveFlumeAgent (4.
↳3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/publish/
↳saveEntityTypeDetails (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/publish/
↳searchEntityTypeDetails (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/publish/
↳fetchEntityTypeDetails/{entity_Id}/{entity_version} (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/entities/{entity_id}/versions/
↳{version_id}/details?projectId={project_id} (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/entities/{entity_Id} (4.
↳3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/publish/
↳compareSchemaWithEntityType (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/metadata/entities/{Entity_Id}/
↳versions/{Entity_Version}/sample-data?projectId={Project_Id} (4.4.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/export/{module_name} (4.3.0)
```

```
http://<bedrock-host-target>:port/bedrock-app/services/rest/import (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/clusters/search (4.5.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/cluster/{Cloud_Cluster_
↳Information_Id} (4.5.0)
```

```
http://<bedrock-host>:<port>/bedrock-app/services/rest/clusters/{cluster_Id} (5.0.0)
```

```
http://<bedrock-host>:<port>/bedrock-app/services/rest/clusters (5.0.0)
```

```
http://<bedrock-host>:<port>/bedrock-app/services/rest/clusters/{cluster_Id}/keypair_
↳ (5.0.0)
```

```
http://<bedrock-host>:<port>/bedrock-app/services/rest/clusters/{cluster_Id}/
↳attributes/search (5.0.0)
```

```
http://<bedrock-host>:<port>/bedrock-app/services/rest/clusters/{cluster_Id}?
↳operation={operation_type} (5.0.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflow/modifyWorkflow (4.3.0)
```

```
http://<bedrock-host>:port/bedrock-app/services/rest/workflows/{workflow_Id}?
↳projectIds={project_Id} (4.4.0)
```

For more information on these APIs, refer to the REST API guide for the specific ZDP release, available in the support.zaloni.com.

1.14 API Response Codes

Each REST call in ZDP returns a response code that indicates the status of the request made by the client. The most common response codes, their descriptions, and the probable causes (if any) are tabulated as follows:

Response Code	Response Message	Description	Probable Cause
200	OK	Indicates that the request is successfully received.	-
201	Created	Indicates that the request is successfully fulfilled.	-
400	Bad Request	Indicates that the request payload cannot be interpreted by the server.	<ul style="list-style-type: none"> The request payload has incorrect parameter(s). The request payload (or structure) is invalid.
401	Unauthorized	Indicates that the user authentication has failed.	<ul style="list-style-type: none"> The <i>x-auth-token</i> is not passed with the request. The token has expired.
403	Forbidden	Indicates that the server cannot process the request as the user does not have access rights.	-
404	Not Found	Indicates that the request made by the client is unrecognized.	<ul style="list-style-type: none"> The request url is incorrect. The url expects request parameters to be passed.
405	Method Not Allowed	Indicates that the method for the request is not supported.	The request method is incorrect.
500	Internal Server Error	Indicates that the server encounters an unexpected condition and restricts the request made by the client (Web browser).	-

1.15 References

1.15.1 Add or Update DQ Rules by using CSV file

To add or update rules in bulk by using the CSV format, follow the below description:

The column names that must be specified in the CSV uploader file are displayed below:

```
Rule Name, Rule Tag, Rule Description, Expression, Failure Message, Error Code, ↵
↵Effective Date, Termination Date, Rule Status, Invert Function, Date Format
```

An example CSV file content is shown below. The example rule: *SB_API_CSV* (in the file content below) during execution can perform a date-range check over records from current date to 5 years back.

```
Rule Name, Rule Tag, Rule Description, Expression, Failure Message, Error Code,
↵Effective Date, Termination Date, Rule Status, Invert Function, Date Format
SB_API_CSV,CSV,Rule Uploaded from CSV,"RANGE_CHECK_DATE (DATE,NOW(),MINUS_DATE (NOW(),
↵YEARS (5) ) )",CSV Uploaded Rule,404,2016-04-26 16:00:00, 2016-05-26 16:00:00,Active,
↵False,"yyyy-MM-dd"
Simple_Rule,CSV,Rule Uploaded from CSV,"GREATER_THAN_NUMBER (NUMBER,1)",CSV Uploaded ↵
↵Rule,404,2016-04-26 16:00:00, 2016-05-26 16:00:00,Active,False,
```

There are a few considerations that must be taken care while using CSV format to add or update rules. These are:

- Only simple rules can be added/updated by using the CSV format.
- While defining fields values to add/update rules, the *Expression* field must comply with the below format:

```
"<Function_Name (<Function_Data_Type>, <user_input_values/operators>)>"
Example - "RANGE_CHECK_DATE (DATE, NOW () , MINUS_DATE (NOW () , YEARS (5) ) ) "
```

- The rule data type is governed by the function data type specified upon the data type. For example, the data type of the rule using the above expression will be of *DATE* data type.
- The *Expression* or *Date Format* (if any) field-value must be included in double quotes (as displayed in the above examples).
- The **Date Format** (if any) must comply with the ones supported in Bedrock Data Quality module. For more information on date formats supported by data quality rules, refer to the *define_rule* section.
- The field (or column) values must not start or end with any white spaces.

1.15.2 Usage of Correct Parameters (Keys) in JSON

Usage of wrong parameters (keys), such as *endDate* as *END DATE*, *startDate* as *START DATE* may result in error response. The keys (such as *executedBy*, *wfId*, *wfName*, *instanceId*, *startDate*, *endDate*) must not be confused as they appear in the ZDP UI.

Observe the keys shown below for a response JSON.

Response:

JSON data:

```
{
  "version": 1,
  "status": "SUCCESS",
  "executedBy": "admin",
  "wfDescription": "Test",
  "scheduledBy": "",
  "firstInstanceStartDate": "",
  "queueName": "",
  "endDate": "05/07/201518: 30:06",
    "startDate": "05/07/201518: 29:00",
  "wfId": 21,
  "wfName": "DIB_CDC_DEL",
  "csvvalues": "250,1,05/07/201518:29:00,05/07/201518:30:06,admin,GENERAL,SUCCESS,null,
  ↪",
  "wfType": "GENERAL",
  "customStatus": null,
  "instanceId": "250"
}
```

1.15.3 Bulk Importing of Metadata by using REST API

Entity fields can be populated in large numbers for an entity type by using the upload CSV feature under the **FIELDS** tab. However, by using this feature, you can only create entity fields in bulk and not set the related features, such as create file patterns or associate entities to workflows.

However, you can use the *Bulk Import of Entities, Ingestion Processes, and Post Ingestion Workflows* API to bulk import new entities and perform ingestion/post-ingestion workflows when adding new datasets to a Hadoop data lake. By using the API, you can save time when compared to performing the same tasks by using the ZDP UI.

By using this service, you can:

- Define entities.
- Register landing zones and configure file patterns.
- Associate post-ingestion workflows.
- Associate masking patterns or tokenization algorithms.

Important: The *Bulk Import of Entities, Ingestion Processes, and Post Ingestion Workflows* API allows you to import only commonly used attributes. For advanced options, use the other APIs.

Bulk Import Pre-requisites

Before using the *Bulk Import of Entities, Ingestion Processes, and Post Ingestion Workflows* API, note the following points:

Note:

- This API can register file pattern with HDFS destination only. Thus, you can ingest files to HDFS locations only.
 - This API does not accept any filesystem connection parameters as input and always uses the first HDFS connection available under the **Connection List** page to associate with the file pattern.
 - The target ZDP instance must have at least one HDFS connection already defined.
 - This API can create source directories based on the value set for the **File System** property set in the **Driver File**. The value can be any of the ZDP supported file systems, such as *LOCAL*, *HDFS*, *AMAZON S3*, or *AZURE BLOB STORAGE*.
 - Before using this service, ensure that at least one connection exists for the file system as defined for the field: **File System Key** in the Driver file.
 - If your ZDP instance has multiple connections for the file system defined in the Driver file, this API always uses the first connection available under the Connection List page.
 - This API always uses the first available connection from the ZDP instance based on the *Connection Id*.
-

To use the bulk import API, certain meta files must be created. There are two types of files required to make this API work. Optionally, a third file is required for tokenization/masking:

- Driver file (required):
 - Driver file must be a CSV delimited file.
 - This file contains a list of the entities that you need to create along with the associated information required to set up the source directory (or landing zone) configuration for each new entity.
 - Each row represents a new entity that needs to be created in the ZDP instance.
 - Each entity (a row in the Driver file) has a one-to-one relationship with one meta file.
- Meta file (required):
 - This is a pipe delimited file.

- Each file defines one entity type.
- Includes all the field level information for the entity type.
- File extension (suffix): *.meta*.
- PII file or sensitivity file (optional):
 - This is specified once and applies to all the entities within the Driver file.
 - Identifies the sensitivity level of fields, masking pattern, and tokenization algorithm.
 - File extension (suffix): *.csv*.

1.15.3.1 Driver file fields

Each row in this file has information about the entity, ingestion, and associated post-ingestion workflow.

Assume that your bulk import driver file tries to create a landing zone directory (or source directory) with a file pattern that already exists. In such a case, this API uses the existing file pattern and updates the entity Id, version as defined in the meta file.

The fields of a driver file are as follows:

Field Name	Description
Meta File Name	Specify the name of the meta file with suffix (if any).
Data File Format	Specify the format in which you want to store the data.
Integrate With HCatalog	Set the value to <i>TRUE</i> as all the <i>HDFS</i> managed entities in ZDP must be <i>HCatalog</i> integrated.
Table Delimiter	Specify the separator (within quotes) if the data format is delimited.
Target Schema	Specify the Hive db under which the HCatalog tables need to be created.
External Table	Set the value to <i>TRUE</i> if you want the Hive table to load data from an external location.
External Data Path	Specify the path where your data is located. The Hive table points to this location for loading data. This value is applicable only if the <i>External Table</i> property is set as <i>TRUE</i> .
Stored As	Set the format in which data is stored in Hive, such as <i>Text File</i> , <i>RC File</i> , <i>Sequence File</i> , or <i>ORC File</i> .
Table Properties	Set the custom table properties (if any) as a key-value pair. The property value(s) must be provided within quotes.
Change Data Capture Field	Set the Change Data Capture (CDC) field.
Change Data Capture Inactive Table	Set whether CDC Inactive Table must be created or not.
File System	Set the file system under which the Landing Zone (or Source Directory) needs to be created. The probable value for this field can be <i>LOCAL</i> , <i>HDFS</i> , <i>AMAZON S3</i> , or <i>AZURE BLOB STORAGE</i> .
File Pattern Prefix	Set the ZDP landing zone data file name prefix.
File Pattern Suffix	Set the ZDP landing zone data file name suffix.
Trigger file Suffix	Set the ZDP landing zone trigger file suffix.
Control File Suffix	Set the suffix of the control file associated with the data file.
Meta File Suffix	Set the suffix of the meta file that will be ingested with the data file.
Delimiter	Set the delimiter used in the data file to be ingested.
Workflow Name	Set the workflow that needs to be triggered as part of the post ingestion process.

Continued on next page

Table 1.1 – continued from previous page

Field Name	Description
Global Parameter	Set the parameters to pass to the post ingestion workflow when it runs. This property is deprecated and thus recommended to leave blank.
Workflow Level Parameter	Set workflow level parameters (if any) to pass automatically when the post ingestion workflow triggers. <ul style="list-style-type: none"> The parameter(s) is applicable only if the <i>Workflow Name</i> property has a value set. Multiple values can be separated using comma (,).
Additional Files extensions	Set the suffix of any additional file that needs to be ingested (optional). Multiple file types can be separated by comma (,).
Destination Path	Set the location in HDFS where this file should be ingested to.
Notification Distribution List	Set the distribution list to be notified as part of ZDP notifications.
Landing Zone Directory	Set the ZDP landing zone directory.
Script Location	Set the location of the pre-ingestion script (if any).
Script Timeout	Set the timeout, in seconds, of the pre-ingestion script. This field value is applicable only if the <i>Script Location</i> is set.
Queue Name	Set the Hadoop capacity queue name (if any).
Bypass Ingestion	Set the value to <i>TRUE</i> to bypass ingestion (if required). Recommended value <i>FALSE</i> .
RuleSet Name	Set the rule set that needs to be associated for the Entity Level Data Quality Action .
Partition File Name	Set the name of the file (if any) containing the partition definitions.

A sample snippet below displays a Driver file content to populate two entity types. Blue snippet represents the Driver file header, whereas the brown snip is the metadata details.

```
Meta File Name,Data File Format,Integrate With HCatalog,Table Delimiter,Target
Schema,External Table,External Data Path,Stored As,Table Properties,Change Data
Capture Field,Change Data Capture Inactive Table,File System,File| Pattern
Prefix,File Pattern Suffix,Trigger file Suffix,Additional Files extensions,Meta
File Suffix,Control File Suffix,Delimiter,Workflow Name,Global
Parameter,Workflow Level Parameter,Workflow Namespace Parameter,Landing Zone
Directory,Destination Path,Notification Distribution List,Script Location,Script
TimeOut,Queue Name,ByPass Ingestion,RuleSet Name,Partition File Name

DH_HSR_HSR_BRND_001.meta,DELIMITED,TRUE,"",zappak,TRUE,/user/bedrock/entity/met
asystem,TextFile,"com.zaloni:abc,org.apache:jar,name:john",,FALSE,LOCAL,BULK_D_P
ATCH_[0-
9]+,.csv,.trg,.done,.meta,.ctl","",DH_Shell_out,"DH_Shell_out,,,,,,",key1:value
1,"MYPATH:/home/bedrock/debajit/BulkImport32,key3:value3",/home/bedrock/debajit/
mylz,/user/bedrock/debajit/mylz,DQ Notification
List,/home/bedrock/debajit/shellpro/preIngestion.sh,20,,FALSE,Standard Entity DQ
Ruleset,partition_sample.csv

HSR_HSR_SYS_SET_APPL_FULL.meta,FIX
LENGTH,TRUE,"",zappak,TRUE,/user/bedrock/debajit/metaugh,TextFile,"com.zaloni:a
bc,org.apache:jar,name:john",,FALSE,LOCAL,BULK_SYS_PATCH_[0-
9]+,.csv,.trg,.done,.meta,.ctl","",DH_Shell_out,"DH_Shell_out,,,,,,",key1:value
1,"MYPATH:/home/bedrock/debajit/BulkImport32,key3:value3",/home/bedrock/debajit/
mylz,/user/bedrock/debajit/mylz,DQ Notification
List,/home/bedrock/debajit/shellpro/preIngestion.sh,20,,FALSE,Standard Entity DQ
Ruleset,partition_sample.csv
```


Note:

- Driver file properties with multiple values must be separated by comma (,) and enclosed within quotes (“”).
- While using the *Bulk Import of Entities, Ingestion Processes, and Post Ingestion Workflows* API, ensure that the Driver file headers and properties use , (comma) as the separator (as shown in the sample snip above).

1.15.3.2 Meta file fields

Meta file stores the field and a few catalog definitions for an entity. A typical meta file in ZDP must have the following headers (or column name).

Field Name	Description
Source_Name	Indicates the name of the source platform of an entity. This value must not have special character.
Schema_Name	Indicates the source schema of an entity. Ensure that the value does not have special characters.
TABLE_NAME	Indicates the business/technical name of the entity. This value must not have special characters.
COLUMN_NAME	Indicates the technical name of the entity field.
DATA_TYPE	Indicates the data type of this field.
DATA_LENGTH	Indicates the size (number of characters) of this field.
DATA_SCALE	Indicates the data scale for the <i>DECIMAL</i> type fields. This field indicates the number of decimal places to round off.
Format	Indicates the display format for this field.
Primary_Keys	Indicates if a field is a <i>Primary Key</i> . The valid values are <i>Y</i> or <i>N</i> .
COLUMN_ID	Indicates the position the entity field. <ul style="list-style-type: none"> • In the ZDP UI, this field is labeled as Order. • Multiple rows within a meta file cannot have the same <i>COLUMN_ID</i> value.
SENSITIVITY	Indicates the sensitivity of this field and can be set based on the requirement.
DELIMITER	Indicates the delimiter to be used.

Note:

- The *HCatalog* table name of the entity is automatically created using the combination of *Source_Name*, *Schema_Name*, and *TABLE_NAME* as one string following the convention:

```
<Source_Name>_<Schema_Name>_<TABLE_NAME>_<Entity_Version#>
```

- The value set for the *Target Schema* property in the *Driver file fields* determines the *Hive* db under which the *HCatalog* table is created.
- For more information on meta file properties, refer to the *configure_fields_settings* section.

A sample meta file content is displayed below. Each row corresponds to an entity field with pipe (|) as the separator for the field properties.

```
Source_Name|Schema_Name|TABLE_NAME|COLUMN_NAME|DATA_TYPE|DATA_LENGTH|DATA_SCALE|Format|Primary_Keys|COLUMN_ID|SENSITIVITY|DELIMITER
source1530647531863|schema1530647531863|table1530647531863|EmpID|INT|8||Y|1|LOW|,
source1530647531863|schema1530647531863|table1530647531863|Name|STRING|10||N|2|HIGH|,
source1530647531863|schema1530647531863|table1530647531863|Designations|STRING|10||N|3|MEDIUM|,|
```

Note: While using the *Bulk Import of Entities, Ingestion Processes, and Post Ingestion Workflows* API, ensure that the meta file headers as well as the property values uses | (pipe) as the separator (as shown in the sample snip above).

1.15.3.3 PII/Sensitivity file fields

Each row in the PII file starts with a field name. This field name is matched to the field names in the meta files and the masking and tokenization rules are applied accordingly. The format of PII files includes the following fields:

Field Name	Description
Name	Indicates the name of the field in an entity to which the sensitivity setting applies. It must match exactly and is case sensitive.
Visible	This field is currently not used in this ZDP version.
Mask	Indicates whether this field is masked or not. The valid values are Y or N.
Masking Pattern	Indicates the masking pattern that applies to this field.
Full Token	Allows tokenization. The valid values are Y or N.
Partial Token	This field is currently not used in this ZDP version.
Tokenization Algorithm Name	Indicates the tokenization algorithm that applies to this field.

The following image displays the content of a PII/Sensitivity file with headers and entity field details.

```
Name,Visible,Mask,Masking Pattern,Full Token,Partial Token,Tokenization Algorithm Name
Entity_Field_1,Y,Y,1-3:##,Y,Y,SHA1
Entity_Field_2,N,Y,1-3:##,Y,Y,SHA1
Entity_Field_3,N,Y,1-2:$;,N,N,NA
```

1.15.4 Reindexing Artifacts

This API allows you to reindex the artifacts: *globalsearch*, *workflow*, *metadata*, and *schedule* available in the ZDP application.

Important:

- To avoid mismatch between data in Elasticsearch with that in the database, you must not perform any create/update/delete operations during execution of the API.
- To reindex the artifacts (*globalsearch*, *workflow*, *metadata*, and *schedule*), you must have the global permission: *Re-Index Artifacts*. By default, this permission is associated with the role: *Global Administrator*.

URL:

```
/bedrock-app/services/rest/artifacts/reindex/es?indexAliases=<comma-separated_index_
↪aliases>
```

Note:

- The parameter: *indexAliases* is optional and re-indexes all the supported indices (if you do not specify any value for this parameter). Valid values for this parameter are: *globalsearch*, *zdp-artifacts*, *metadata*, *schedule*, and *workflow*. This API is asynchronous.

- The following Elasticsearch types are not present in database (you cannot perform reindexing):

Index	Type
globalsearch	dmemodels
workflow	jobdetails

Method:

POST

Example Response:

```
{
  "responseMessage": "Reindexing Triggered.",
  "restUri": "/bedrock-app/services/rest/artifacts/reindex/es?indexAliases=schedule,
↪metadata,workflow,globalsearch",
  "result": "Reindexing has been triggered for aliases: schedule,metadata,workflow,
↪globalsearch. It is an asynchronous process and will be completed soon. Refer_
↪gateway logs for details.",
  "page": null
}
```