Oracle Financial Services

Public APIs for Profitability and Balance Sheet Management Cloud Service

Release 24B

F83683-08

June 2024

ORACLE®

Oracle Financial Services Public APIs for Profitability and Balance Sheet Management Cloud Service, Release 24B

F83683-08

Contents

ORACLE

7   Generating PAR URL for File Operations

8   Deleting A File

9   Generate Report API

10   Get Report Generation Status and PAR URL Link API

11   External Scheduler Interface

1
Getting Started

Before executing the Rest APIs and performing various tasks, refer to the following topics to meet the necessary requirements.

- Prerequisites
- Support Methods
- Support Headers
- Status Codes

Prerequisites

The following are the set of prerequisites required for executing/invoking Rest APIs.

- Access to the Profitability and Balance Sheet Management Cloud Service.
- Appropriate User Privileges to access the services.
- Technical and Functional knowledge to understand and execute the REST APIs and configuration knowledge.
- Knowledge of REST Concepts, JSON, and browser-based REST Client.
- Knowledge of interactive and automatic tools to verify the APIs such as Postman and Command Line Interfaces (CLI).

Obtain Account Information

You can get the Identity Domain name from the account creation email sent by Oracle Support.

You can also contact your Service Administrator, to get the account information.

Supported Methods

- **GET**: Retrieve information about the service instance.
- **POST**: Create, scale, backup, start, and stop the service instance.

Media Types

The following media type is supported by the Accounting Foundation Cloud Service REST APIs:

- application/json

Supported Headers

The REST API supports headers that may be passed in the header section of an HTTP Request or Response.

Table 1-1   Table: Supported Headers

Headers	Description	Example
Content-Type	The media type of the body of the request. Required for POST and PUT requests, and the supported types vary with each endpoint.	Content-Type: application/json
Accept	The media type of the body of the response.	Accept: application/json

Status Code

When you call the Accounting Foundation Cloud Service REST APIs Resources, the Response Header returns one of the standard HTTP Status Codes.

Table 1-2   Status Code

HTTP Status Code	Description
200 OK	The request was successfully completed. A 200 status is returned for a successful GET or POST Method.
201 Created	The request has been fulfilled and resulted in a new resource being created. The response includes a Location Header containing the canonical URI for the newly created resource. A 201 status is returned from a synchronous resource creation or an asynchronous resource creation that was completed before the response was returned.
202 Accepted	The request has been accepted for processing, but the processing has not been completed. The request may or may not eventually be acted upon, as it may be disallowed at the time the processing takes place. When specifying an Asynchronous (__detached=true) Resource creation (for example, when deploying an application), or update (for example, when redeploying an application), a 202 is returned if the operation is still in progress. If __detached=false, a 202 may be returned if the underlying operation does not complete in a reasonable amount of time.
400 Bad Request	The request could not be processed because it contains missing or invalid information (such as a validation error on an input field, a missing required value, and so on).
401 Unauthorized	The request is not authorized. The Authentication Credentials included with this request are missing or invalid.
403 Forbidden	The user cannot be authenticated. The user does not have the authorization to perform this request.
404 Not Found	The request includes a resource URI that does not exist.
405 Method Not Allowed	The HTTP verb specified in the request (DELETE, GET, POST, PUT) is not supported for this request URI.

Table 1-2   (Cont.) Status Code

HTTP Status Code	Description
406 Not Acceptable	The resource identified by this request is not capable of generating a representation corresponding to one of the media types in the Accept Header of the request. For example, the client's Accept Header request XML be returned, but the resource can only return JSON.
409 Conflict	The client's ContentType Header is not correct (for example, the client attempts to send the request in XML, but the resource can only accept JSON).
415 Not Acceptable	The client's ContentType Header is not correct (for example, the client attempts to send the request in XML, but the resource can only accept JSON).
500 Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request.
503 Service Unavailable	The server is unable to handle the request due to temporary overloading or maintenance of the server. The REST Web Application is not currently running.

2
Authenticating for Token Generation

An Authentication token is required to invoke an API to generate the File Upload/Download PAR URL. The Authentication Process for token generation, utilizes cURL Commands in a CLI Tool to generate the access token and invoke REST APIs.

The Authentication Token is generated through the OAuth Client ID and Secret Credentials created in IAM during Provisioning. The Authentication Token does not require that you log in to the required Cloud Service to invoke the REST APIs from external applications.

Ensure that you have the appropriate log-in credentials to access the required Cloud Service and the appropriate roles to perform specific operations using the API Resources. Below is a list of authentication steps, with subsequent sections offering detailed information:

1. Download application certificate
2. Get the OAuth Client ID and Client Secret
3. Generate the access token
4. Generate refresh token
5. Invoke API using the access token

Download the Application Certificate

The Application Certificate is required for verification purposes when you use cURL commands.

You may choose not to download the certificate if you plan to turn off the cURL Certificate Verification and use an insecure connection (if you add the `--insecure` Flag to the cURL command).

To download the Application Certificate:

1. Log in to your Cloud service.
2. Click **View site information/Verified by** in the browser's Address bar.
3. Select **More information** to view the certificate.
4. Click **View Certificate** and then click **PEM(cert)** to download the certificate.

Get the OAuth Client ID and Client Secret

An OAuth Cliend ID and Client secret are required to generate an access token.

To get the OAuth Client ID and Client Secret:

1. Enter the **Oracle Identity and Access Management (IAM)** URL in the browser's Address bar to access the **Oracle Cloud Account Sign In** page.
2. Log in to **IAM** portal.
3. Click **Navigation** to view a list of available functions.
4. Select **Oracle Cloud Services**.

For more information, see Access Service Consoles from **Administering Oracle Identity Cloud Service**.

5. From the Oracle Cloud Services page, select the required Cloud Service Internal Application Service (in **<Cloud_service_name> <tenant-id> INTERNAL** format) from the list. For example, the cloud service name - **PBSMCS**

6. Click the **Configuration** tab.

   The Client ID and Client Secret Details are displayed in the General Information section.

7. Copy the `Client ID` and `Client Secret`.

8. Open a CLI Tool.

9. Proceed to generate the access token.

You can also get the OAuth client ID and client secret using Admin Console.

Generate the Access Token

Access token is required to invoke API and you can generate an access token using cURL commands.

To generate the Access Token, add the Client ID, Client Secret, User Name, and Password using cURL Commands in the CLI Tool. You can use an insecure connection (if you add the `--insecure` / `-k Flag` to the cURL command). The following is an example:

```
curl -k -i -H "Authorization: Basic  < Base64 Encoded
      Outh Cred >" -H "Content-Type: application/x-www-form-
urlencoded;charset=UTF-8"
      --request POST https://<iam_tenant>:443/oauth2/v1/token -d

"grant_type=password&scope=urn:opc:idm:__myscopes__+offline_access&username=<u
serid>&password=<Password>"
```

Sample Code

```
curl -k -i -H "Authorization: Basic
YWFpdGVzdGRldjEwMDEtcHJkX0FQUElEOjQyYjJlYWVlLTY1OGEtNDgzYi1hMWI2LTBlYzU0MzBmYW
QwNQ==" -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" --
request POST https://
iam-0cb0c2b3ba624afca67467fd5eb9db49.identity.c9dev2.oc9qadev.com:443/
oauth2/v1/token -d
"grant_type=password&scope=urn:opc:idm:__myscopes__+offline_access&username=cn
eadmin&password=Password@12345"
```

After generating the Access Token, invoke the API as shown in the following section.

> ✎ **Note:**
>
> The Access token expiry (in seconds) is configurable and can be set at the time of generating the access token. In the preceding example, it is set to 3600 seconds ~ 1 hour. By default, the expiry is set to 3600 seconds ~ 1 hour. You can configure this to a value of your choice up to a maximum value of 31536000 seconds ~ 1 year.

The token is sent as a response. Store the token in a secure location.

Sample Access Token (Truncated example)

```
{"access_token":"eyJ4NXQjUzI1NiI6IlF5azRtb3pIakhuQjJoQnVWdmZXZUpVeVZrNHhUdWd6a
WpHSC1pV21xb1EiLCJ4NXQiOiJDRFhHYVlWZXI3STVhQ1l
...
...
DB_be0Rtw1aMxFYg8Ft0VaK14wOVFGhgg1Cr6GiNvbgeYRG5uwgJGqw","token_type":"Bearer"
,"expires_in":3600,"refresh_token":"AgAgYjA1OGVlMjJiMWY2NGU3YWFkM2NjZWNlOTc2Mj
NiNDgIABBmZRHXpaHil2VBXkevFX-
iAAAAMMq9uQDo86eVVVisw3kYn80iX8qRJ2m7hMLmMAh1dY9Wgy-ESu8WYzdTBXOsnwHr7A=="}
```

Generate the Refresh Token

Refresh tokens are used to generate access tokens for invoking APIs.

To generate a Access token using Refresh token, use the following Curl command. You can use an insecure connection (if you add the `--insecure` / `-k` flag to the cURL command). The following is an example:

```
curl -k -i -H "Authorization: Basic <base64Encoded clientid:secret>" -H
"Content-Type:
        application/x-www-form-urlencoded;charset=UTF-8" --request POST
        https://<IdentityDomainURL>/oauth2/v1/token -d

"scope=urn:opc:idm:__myscopes__&grant_type=refresh_token&refresh_token=<refres
h_token>"
```

Sample Code

```
curl -k -i -H "Authorization: Basic

cWppMHBkLXByZF9BUFBJRDplZjFjMTVmZi1lZDBiLTQxNmItYTFmYy0wNjhlYzM5NmUxM2Y=" -H
        "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" --
request POST
        https://<IdentityDomainURL>/oauth2/v1/token -d

"scope=urn:opc:idm:__myscopes__&grant_type=refresh_token&refresh_token=AgAgYjA
1OGVlMjJiMWY2NGU3YWFkM2NjZWNlOTc2MjNiNDgIABBmZRHXpaHil2VBXkevFX-
iAAAAMMq9uQDo86eVVVisw3kYn80iX8qRJ2m7hMLmMAh1dY9Wgy-ESu8WYzdTBXOsnwHr7A=="
```

Sample Refresh Token (Truncated example)

```
{"access_token":"eyJ4NXQjUzI1NiI6IlF5azRtb3pIakhuQjJoQnVWdmZXZUpVeVZrNHhUdWd6a
WpHSC1pV21xb1EiLCJ4NXQiOiJDRFhHYVlWZXI3STVhQ1l...
...
...
token_type":"Bearer","expires_in":3600,"refresh_token":"AgAgYjA1OGVlMjJiMWY2NG
U3YWFkM2NjZWNlOTc2MjNiNDgIABA4t8V_dYVyc5lOuKezofTUAAAAMJrpmKRhDWf3-
ejCreU8_Po5Bb95srwUDDs5cV1gT-x26twbAfp_ffMCiEgjqGeDNw=="}
```

Invoke the API using the Access Token

After creating an access token using OAUth Client ID and Client secret, you can invoke the Specific API.

To invoke the API using the generated Access Token, refer to the following example executed using cURL Commands in the CLI Tool:

```
curl -iL -H "Authorization: Bearer <access token>" -H "Content-Type:
<content_type>" -d "<request_body>" --cacert <certificate(.pem)> -X
<http_verb> <api_url>

curl -iL -H "Authorization: Bearer <AUTH_TOKEN>"

-H "Content-Type: application/json" -d "{\"type\":\"files\",\"data\":
[{\"fileName\":\"testtoken\",\"mimeType\":\"text/plain\",\"fileSize\":

123}]}" --cacert outcert.pem -X POST https://<OCI-URL>/<TENANT><APP_ID>/dsa/
utils/getObjStoreParUrl
```

3

Invoking Export API

You can use Invoke Export API to export migration definitions that are created using the application interface.

For more information about Invoke Export API, refer to

- Endpoint details
- Response JSON Parameters - Invoke Export API

End Point Details

- **Method** – POST
- **REST Endpoint** -/public/migrate/export/`{code}`, where code is the migration ID.
- **Content-Type** - application/json

Request Headers Details

The following table lists the Request Headers:

- **ofs_tenant_id** - Tenant ID of the Application.
- **Locale** - Local language in the language code format. For example, `en-US`.
- **ofs_remote_user** - User ID of the user.
- **ofs_workspace_id** - Workspace ID of the Application. The default value is `WS001` and same should be passed each time.

Response JSON Parameters

This section provides the list of parameters in the JSON Response.

Sample Response

```
Success:200
{
"errorMessage": "Object Migration Triggered",
"statusCode": "TRIGGERED_SUCCESS",
"status": "SUCCESS"
}
Error: 400
{
"errorMessage": "Export already completed",
"statusCode": "EXPORT_ALREADY_COMPLETE",
"status": "FAILED"
}
{
"errorMessage":"Code does not exist",
"statusCode": "INVALID_INPUT",
"status": "FAILED"
```

```
}
{
"errorMessage":"Export in progress",
"statusCode": "EXPORT_INPROGRESS",
"status": "INPROGRESS"
}
Error :500
{
"errorMessage": "Internal Server Error",
"statusCode": "INTERNAL_ERROR",
"status": "FAILED"
}
```

4

Invoking Import API

You can use Invoke Import API to import migration definitions that are created using the application interface.

For more information about Invoke Import API, refer to

End Point Details

- **Method** – POST
- **REST Endpoint** -/public/migrate/import/ `{code}`, where code is the migration ID.
- **Content-Type** - application/json

Request Headers Details

The following table lists the Request Headers:

- **ofs_tenant_id** - Tenant ID of the Application.
- **Locale** - Local language in the language code format. For example, `en-US`.
- **ofs_remote_user** - User ID of the user.
- **ofs_workspace_id** - Workspace ID of the Application. The default value is `WS001` and same should be passed each time.

Response JSON Parameters

This section provides the list of parameters in the JSON Response.

Sample Response

```
Success:200
{
"errorMessage": "Object Migration Triggered",
"statusCode": "TRIGGERED_SUCCESS",
"status": "SUCCESS"
}
Error: 400
{
"errorMessage": "Import already completed",
"statusCode": "IMPORT_ALREADY_COMPLETE",
"status": "FAILED"
}
{
"errorMessage":"Code does not exist",
"statusCode": "INVALID_INPUT",
"status": "FAILED"
```

```
}
{
"errorMessage":"Import in progress",
"statusCode": "IMPORT_INPROGRESS",
"status": "INPROGRESS"
}
Error :500
{
"errorMessage": "Internal Server Error",
"statusCode": "INTERNAL_ERROR",
"status": "FAILED"
}
```

5

Get Export Status API

You can use Get Export Status API to view the export status of an ongoing migration .

For more information about the API, refer to

- Endpoint details
- Response JSON Parameters - Get Export Status API

End Point Details

- **Method** – GET
- **REST Endpoint** -/Public/status/export/`{code}`, where code is the migration ID.
- **Content-Type** - application/json

Request Headers Details

The following table lists the Request Headers:

- **ofs_tenant_id** - Tenant ID of the Application.
- **Locale** - Local language in the language code format. For example, `en-US`.
- **ofs_remote_user** - User ID of the user.
- **ofs_workspace_id** - Workspace ID of the Application. The default value is `WS001` and same should be passed each time.

Response JSON Parameters

This section provides the list of parameters in the JSON Response.

Sample Response

```
Success:200
{
"errorMessage": "Status fetched Successfully",
"statusCode": "SUCCESS",
"status": "SUCCESS"
}
Error: 400
{
"errorMessage": "Code does not exist",
"statusCode": "INVALID_INPUT",
"status": "FAILED"
}
{
"errorMessage":"Code does not exist",
"statusCode": "INVALID_INPUT",
"status": "FAILED"
}
```

```
Error :500
{
"errorMessage": "Internal Server Error",
"statusCode": "INTERNAL_ERROR",
"status": "FAILED"
}
```

6

Get Import Status API

You can use Get Import Status API to view the import status of an ongoing migration .

For more information about Get Import Status API, refer to

-
-

End Point Details

- **Method** – GET
- **REST Endpoint** -/public/status/import/ `{code}`, where code is the migration ID.
- **Content-Type** - application/json

Request Headers Details

The following table lists the Request Headers:

- **ofs_tenant_id** - Tenant ID of the Application.
- **Locale** - Local language in the language code format. For example, `en-US`.
- **ofs_remote_user** - User ID of the user.
- **ofs_workspace_id** - Workspace ID of the Application. The default value is `WS001` and same should be passed each time.

Response JSON Parameters

This section provides the list of parameters in the JSON Response.

Sample Response

```
Success:200
{
"errorMessage": "Status fetched Successfully",
"statusCode": "SUCCESS",
"status": "SUCCESS"
}
Error: 400
{
"errorMessage": "Code does not exist",
"statusCode": "INVALID_INPUT",
"status": "FAILED"
}
{
"errorMessage":"Code does not exist",
"statusCode": "INVALID_INPUT",
"status": "FAILED"
}
```

```
Error :500
{
"errorMessage": "Internal Server Error",
"statusCode": "INTERNAL_ERROR",
"status": "FAILED"
}
```

7

Generating PAR URL for File Operations

The PAR URL for File Operations API creates a PAR File that you can use to perform file operations in the Object Store for end-to-end integrations.

Generating PAR URL for File Upload

Generate PAR URL for File Upload

You can use this REST API to generate the PAR URL for File Upload. See the following sections for information on how to perform the POST operation.

- End Point Details
- Calling the API to Generate the URL

End Point Details

- **Method** – POST
- **URL** - `https://<HOST_NAME:PORT>/<TENANT>/utils-service/v1/file/uploadfile/parURL?prefix=<prefix>`
- **Content-Type** - Application/Json

Calling the API to Generate the URL

To call the API:

1. Open a relevant tool, such as via cURL command.
2. Prepare a cURL command with the authentication token and other details. For more information refer to the following code.

   Syntax

   ```
   curl -k --location --request POST 'https://<hostname>/<TENANT-ID>/utils-
   service/v1/file/uploadfile/parURL?prefix=' \
   --data-raw '{"fileName": "<remote filename>", "fileSize": <file size>,
   "mimeType": "<file type>"}' \
   --header 'ofs_remote_user: <USERID>' \
   --header 'locale: en-US' \
   --header 'ofs_tenant_id: <TENANT-ID>' \
   --header 'ofs_workspace_id: WS001' \
   --header 'content-type: application/json' \
   --header 'Authorization: Bearer <TOKEN>'
   ```

   Example (truncated)

   ```
   curl -k --location --request POST 'https://<hostname>/<TENANT-ID>/utils-
   service/v1/file/uploadfile/parURL?prefix=' \
   --data-raw '{"fileName": "idcs_log1.txt", "fileSize": 100, "mimeType":
   ```

```
"text/plain"}' \
--header 'ofs_remote_user: cneadmin' \
--header 'locale: en-US' \
--header 'ofs_tenant_id: aaitestdev1001-prd' \
--header 'ofs_workspace_id: WS001' \
--header 'content-type: application/json' \
--header 'Authorization: Bearer
eyJ4NXQjUzI1NiI6Ildia25rQUR5TUZIMlhlQ1pKcTY1c3o4VzdEVWhKa0s4MldYY0hadk4wWkk
iLCJ4
...
...
sQXj0iohsSIEmQXVwwjhhqnc4eJNnmCjx8Tb7TXjx1MIQLeOIcfrIj5gkzoMKX94_7USxHv-6Lh
Bzw'
```

Request JSON Parameters

This section provides the list of parameters in the JSON Request.

Table 7-1   Request JSON Parameters

Name	Type	Required	Description
fileName	STRING	Yes	The name of the file to be uploaded. The following are the conditions for to enter in this field: • Must start with an Alphanumeric Character • Allowed characters are alphabets, numbers, and special characters - hyphen(-), dot(.), and underscore(_) • Length of characters must not be greater than 255 characters
fileSize	INTEGER	Yes	The size of the file (in Bytes) to be uploaded. The size of the file should be greater than 1 Byte and should be less than 10 TB. It is recommended to use multipart upload for uploading files of size more than 100 MB. For more information about uploading large objects and multipart uploads, refer to Working with Pre-Authenticated Requests.
mimeType	STRING	Yes	The mime type to be uploaded. The following mime types are allowed: • Text/CSV • Text/plain • DAT

Request JSON Sample

```
 [{
"fileName": "File.csv",
"fileSize": 7654,
"mimeType": "text/csv"
}]
```

Response JSON Parameters

This section provides the list of parameters in the JSON Response.

Table 7-2   POST JSON Response

Name	Type	Description
fileName	STRING	The name of the file to be uploaded.
uploadURL	STRING	The generated pre-authenticated URL to upload a file.
fileId	INTEGER	The unique File Identifier.

Response JSON Sample

```
{
   "payload": {
       "uploadURL": "https://objectstorage.us-phoenix-1.oraclecloud.com/p/
bdSI-hzigiAoUU0lyEKnuk0YGs05L172gt_woZAgqNFYmUFQeexV3BDfTO97mhBI/n/
oraclegbudevcorp/b/fsgbu_pbsm_cndevcorp_ftpqa101231-prd_default/o/default/
2023-01-31/jfr/f9ce031f-4a42-471d-b4da-d0577f3eca15",
       "createUser": "user1",
       "stripeName": "default",
       "fileId": 5025,
       "createDate": "2023-01-31T09:14:16",
       "token": "",
       "status": "success"
   }
}
```

Viewing List of Uploaded Files

Run the following cURL command to generate and view all the files that are uploaded using PAR URL.

Syntax

```
curl -k --location --request GET 'https://<hostname>/<TENANT-ID>/utils-
service/v1/listfiles stripeName=default' \
--header 'locale: en-US' \
--header 'ofs_remote_user: <user id>' \
--header 'ofs_tenant_id: < TENANT-ID >' \
--header 'ofs_workspace_id: WS001' \
--header "Authorization: Bearer <TOKEN>'
```

Example

```
curl -k --location --request GET 'https://dc.pbsmcloud.us-phoenix-1.ocs.oc-
test.com/aaitestdev1001-prd/utils-service/v1/listfiles?stripeName=default' \
--header 'locale: en-US' \
--header 'ofs_remote_user: cneadmin' \
--header 'ofs_tenant_id: aaitestdev1001-prd' \
--header 'ofs_workspace_id: WS001' \
--header "Authorization: Bearer ${TOKEN}"
```

Response

```
    {"payload":
[{"v_file_name":"Idcs_log3.txt","n_file_id":9916,"d_upload_date":"31-JAN-23
    06:33:43 AM","v_stripe_name":"default"},
{"v_file_name":"Idcs_log4.txt","n_file_id":9917,"d_upload_date":"31-JAN-23
    06:40:25 AM","v_stripe_name":"default"}],"count":2}
```

Generating PAR URL For File Download

You can use this REST API to generate the PAR URL for File Download. See the following sections for information on how to perform the post operation.

- Calling the API to Generate PAR URL for File Download Using File Name
- Calling the API to Generate PAR URL for File Download Using File ID

Calling the API to Generate PAR URL for File Download Using File Name

To call the API:

1. Open a relevant tool, such as via the cURL command.

2. Prepare a cURL command with the authentication token and other details. For more information refer to the following code.

   Syntax

   ```
   curl -k --location --request GET < 'https://<hostname>/<TENANT-ID>/utils-
   service/v1/file/download?fileName=<file name>&stripeName=default&prefix=' \
    --header 'ofs_remote_user: <userid>' \
    --header 'locale: en-US' \
    --header 'ofs_tenant_id: <TENANT-ID>' \
    --header 'ofs_workspace_id: WS001' \
    --header "Authorization: Bearer <TOKEN>"
   ```

   Example

   ```
   curl -k --location --request GET 'https://<hostname>/<TENANT-ID>/utils-
   service/v1/file/download?fileName=test3GB.xml&stripeName=default&prefix=' \
    --header 'ofs_remote_user: cneadmin' \
    --header 'locale: en-US' \
    --header 'ofs_tenant_id: aaitestdev1001-prd' \
    --header 'ofs_workspace_id: WS001' \
    --header "Authorization: Bearer ${TOKEN}"
   ```

   Response

   ```
    {"payload":{"downloadURL":"https://objectstorage.us-
   phoenix-1.oraclecloud.com/p/8R68eVcQAxQjNjK__S04MZjS-
   v4BqEbWSILvu0w40kJNrzfKeCB8vWBwugW5XvsK/n/oraclegbudevcorp/b/
   fsgbu_pbsm_cndevcorp_aaitestdev1001-prd_default/o/default/2023-01-20/rnz/
   6c023e75-09e2-4265-815e-32cedcd2415e?
   httpResponseContentDisposition=ATTACHMENT%3B%20filename%3Dtest3GB.xml"}}
   ```

Calling the API to Generate PAR URL for File Download Using File ID

To call the API, follow these steps:

1. Open a relevant tool, such as via the cURL command.

2. Prepare a cURL command with the authentication token and other details. For more information, refer to the following code.

   Syntax

   ```
   curl -k --location --request GET ' 'https://<hostname>/<TENANT-ID> /utils-
   service/v1/file/downloadfile/<file id>' \
   --header 'ofs_remote_user: <userid>' \
   --header 'locale: en-US' \
   --header 'ofs_tenant_id: < TENANT-ID> ' \
   --header 'ofs_workspace_id: WS001' \
   --header "Authorization: Bearer <TOKEN>"
   ```

   Example

   ```
   curl -k --location --request GET 'https://<hostname>/<TENANT-ID>/utils-
   service/v1/file/downloadfile/9916' \
   --header 'ofs_remote_user: cneadmin' \
   --header 'locale: en-US' \
   --header 'ofs_tenant_id: aaitestdev1001-prd' \
   --header 'ofs_workspace_id: WS001' \
   --header "Authorization: Bearer ${TOKEN}"
   ```

   Response

   ```
   {"payload":{"downloadURL":"https://objectstorage.us-
   phoenix-1.oraclecloud.com/p/
   gTxxzhqLEea4Or2TRkBqTqHxt_JogVFa9G_0wtN8NYy_op0Zk4lvKGDxxeXGhLq7/n/
   oraclegbudevcorp/b/fsgbu_pbsm_cndevcorp_aaitestdev1001-prd_default/o/
   default/2023-01-31/fae/2d63d2fe-2090-4fb7-a4c8-9940d22987db?
   httpResponseContentDisposition=ATTACHMENT%3B%20filename%3DIdcs_log3.txt"}}
   ```

8

Deleting A File

Delete (DELETE) API helps to delete an uploaded file.

For more information about the Delete API, refer to Endpoint Details.

You can delete a file using one of the following methods:

- Using File ID
- Using File Name
- Delete multiple Files using File Names

Endpoint Details

Delete (DELETE) API helps to delete an uploaded file.

- **HTTP Method** - Delete
- **Header Parameters**
  - **ofs_remote_user** - User ID of the user mapped to `'BATCH_EXEC'` function.
  - **locale** - locale in `languageCode-countryCode` format. For example, `en-US`.
  - **ofs_tenant_id** - Tenant ID of the Application
  - **ofs_workspace_id** - Workspace ID of the Application. It is defaulted to `WS001` and same should be passed each time.
  - **Content-type** - The media type of the body of the request. The content-type details are required for POST and PUT requests, and the supported types vary with each endpoint. The value is `application/json`.
  - **Authorization** - Access token required to authenticate the API. If this token is not provided, `401 Unauthorized error` is generated. For more information about Bearer token, refer to Generate the Access Token.

Deleting a File Using the File ID

Delete a file from the object store, using the file ID as the reference.

To delete a file:

1. Open a relevant tool, such as via cURL command.
2. Prepare a cURL command with the authentication token and other details. For more information refer to the following code.

   Syntax

   ```
   curl -k --location --request POST 'https://<hostname>/<TENANT-ID>/utils-
   service/v1/file/deletefile/{fileId}
   --header 'ofs_remote_user: <USERID>' \
   --header 'locale: en-US' \
   ```

```
--header 'ofs_tenant_id: <TENANT-ID>' \
--header 'ofs_workspace_id: WS001' \
--header 'content-type: application/json' \
--header 'Authorization: Bearer <TOKEN>'
```

Example (truncated)

```
curl -k --location --request POST 'https://<hostname>/<TENANT-ID>/utils-
service/v1/file/deletefile/5
--header 'ofs_remote_user: cneadmin' \
--header 'locale: en-US' \
--header 'ofs_tenant_id: aaitestdev1001-prd' \
--header 'ofs_workspace_id: WS001' \
--header 'content-type: application/json' \
--header 'Authorization: Bearer
eyJ4NXQjUzI1NiI6Ildia25rQUR5TUZIMlhlQ1pKcTY1c3o4VzdEVWhKa0s4MldYY0hadk4wWkk
iLCJ4
...
...
sQXj0iohsSIEmQXVwwjhhqnc4eJNnmCjx8Tb7TXjx1MIQLeOIcfrIj5gkzoMKX94_7USxHv-6Lh
Bzw'
```

Response

```
{"payload":"File Deleted Successfully"}
```

Deleting a File Using Filename

Delete a file from the object store, using the file name as the reference.

To delete a file:

1. Open a relevant tool, such as via cURL command.

2. Prepare a cURL command with the authentication token and other details. For more information refer to the following code.

   Syntax

   ```
   curl -k --location --request POST 'https://<hostname>/<TENANT-ID>/utils-
   service/v1/file/deletefilename/{filename}?prefix=<foldername>
   --header 'ofs_remote_user: <USERID>' \
   --header 'locale: en-US' \
   --header 'ofs_tenant_id: <TENANT-ID>' \
   --header 'ofs_workspace_id: WS001' \
   --header 'content-type: application/json' \
   --header 'Authorization: Bearer <TOKEN>'
   ```

   > **✎ Note:**
   >
   > Prefix is an optional parameter.

ORACLE

Example (truncated)

```
curl -k --location --request POST 'https://<hostname>/<TENANT-ID>/utils-
service/v1/file/deletefilename/test.txt?prefix=folder1
--header 'ofs_remote_user: cneadmin' \
--header 'locale: en-US' \
--header 'ofs_tenant_id: aaitestdev1001-prd' \
--header 'ofs_workspace_id: WS001' \
--header 'content-type: application/json' \
--header 'Authorization: Bearer
eyJ4NXQjUzI1NiI6Ildia25rQUR5TUZIMlhlQ1pKcTY1c3o4VzdEVWhKa0s4MldYY0hadk4wWkk
iLCJ4
...
...
sQXj0iohsSIEmQXVwwjhhqnc4eJNnmCjx8Tb7TXjx1MIQLeOIcfrIj5gkzoMKX94_7USxHv-6Lh
Bzw'
```

Response

```
{"payload":"File Deleted Successfully"}
```

Deleting Multiple Files Using Filenames

Delete multiple files from the object store, using the file names as the reference.

To delete multiple files:

1. Open a relevant tool, such as via cURL command.

2. Prepare a cURL command with the authentication token and other details. For more
    information refer to the following code.

    Syntax

    ```
    curl -k --location --request POST 'https://<hostname>/<TENANT-ID>/utils-
    service/v1/file/deletefilenames/{filenames}
    --data-raw '[filenames]' \
    --header 'ofs_remote_user: <USERID>' \
    --header 'locale: en-US' \
    --header 'ofs_tenant_id: <TENANT-ID>' \
    --header 'ofs_workspace_id: WS001' \
    --header 'content-type: application/json' \
    --header 'Authorization: Bearer <TOKEN>'
    ```

    Example (truncated)

    ```
    curl -k --location --request POST 'https://<hostname>/<TENANT-ID>/utils-
    service/v1/file/deletefilenames
    --data-raw '["filename1.txt","filename2.txt","filename3.txt"]' \
    --header 'ofs_remote_user: cneadmin' \
    --header 'locale: en-US' \
    --header 'ofs_tenant_id: aaitestdev1001-prd' \
    --header 'ofs_workspace_id: WS001' \
    --header 'content-type: application/json' \
    --header 'Authorization: Bearer
    ```

eyJ4NXQjUzI1NiI6Ildia25rQUR5TUZIMlhlQ1pKcTY1c3o4VzdEVWhKa0s4MldYY0hadk4wWkk
iLCJ4
...
...
sQXj0iohsSIEmQXVwwjhhqnc4eJNnmCjx8Tb7TXjx1MIQLeOIcfrIj5gkzoMKX94_7USxHv-6Lh
Bzw'

Response

```
{"payload":"File Deleted Successfully"}
```

9

Generate Report API

Use Generate Report API, to generate report based on the given input data.

For more information about Generate Report API, refer to

- Endpoint Details
- Request JSON Parameters
- Response JSON Parameters

End Point Details

- **Method** – POST
- **REST Endpoint** - /aai-dmi-service/vw/v1/repgen
- **Content-type** - application/json

Request Headers Details

- **ofs_remote_user** - (String). User ID of the user.
- **ofs_service_id** - (String).The unique service ID associated with the report.
- **ofs_workspace_id** - (String). The Workspace ID of the Application. It is defaulted to "WS001" and same should be passed each time.
- **Locale** - (String). The local language in the language code format. For example, `en-US`.
- **reportcode** - (String). The auto-generated code for the specific report. You can get this code from the Report Details page.
- **reportType** - (String). The report file format.
- **appid** - (String). Respective application ID of the cloud service
- **reportDefType** - (String). The report definition type. By default this value is set to **V** (View).

Request JSON Parameters

Request JSON Parameters for Generate Report API

- **filterCriteria** - The filter conditions set to include a specific set of data in the report, using the defined expressions. You can add the filter criteria in the Filter Conditions page.

Request JSON Sample when there is no filter

```
{
"filterCriteria":""
"placeholderFilter": ""
}
```

Request JSON Sample when filter applied

```
{
 "filterCriteria": "FSI_ACCID_HEALTH_INS_CO_CD.CREATED_BY =
     'DMIUSER1'"
}
{
 "filterCriteria": "FSI_ACCID_HEALTH_INS_CO_CD.CREATED_BY = 'DMIUSER1' AND
     FSI_ACCID_HEALTH_INS_CO_CD.CO_ID IN ('100', '200') "
}
{  "filterCriteria": "FSI_ACCID_HEALTH_INS_CO_CD.CREATED_BY = 'DMIUSER1' AND
     FSI_ACCID_HEALTH_INS_CO_CD.CO_ID IN ('100', '200') ",
"placeholderFilter":
     "FSI_ACCID_HEALTH_INS_CO_CD.CURRENCY = PLACEHODER1 AND
FSI_ACCID_HEALTH_INS_CO_CD.COUNTRY =
     'IND' AND FSI_ACCID_HEALTH_INS_CO_CD.AS_OF_DATE = PLACEHOLDER2 "
}
```

Response JSON Parameters

Sample Response

```
{
  "status": "PENDING",
  "requestInstanceId": "100"
}
```

10
Get Report Generation Status and PAR URL Link API

View the report generation status and also get a pre-authenticated URL (PAR URL), to download the report in CSV format, using the Get Report Generation Status and PAR URL Link API.

The status response can contain the following values:

- **COMPLETED** - When the export API execution is completed successfully and response is generated.
- **PENDING** - denotes that the report export is yet to start and is in the queue.
- **RUNNING** - When the API execution is in progress.
- **FAILED** - When the Export API execution is not completed successfully.

For more information about this API,

- Endpoint Details
- Response JSON Parameters

End Point Details

- **Method** – POST
- **REST Endpoint** - aai-dmi-service/vw/v1/repgenstatus
- **Content-type** - application/json

Request Headers Details

- **ofs_remote_user** - (String). User ID of the user.
- **ofs_service_id** - (String).The unique service ID associated with the report.
- **ofs_workspace_id** - (String). The Workspace ID of the Application. It is defaulted to "WS001" and same should be passed each time.
- **Locale** - (String). The local language in the language code format. For example, `en-US`
- **reportcode** - (String). The auto-generated code for the specific report. You can get this code from the Report Details page.
- **reportInstanceID** - (Integer). The instance ID of the report which is being exported. You can get the report /request instance ID from Response JSON Parameters of Generate Report API.

Response JSON Parameters

This section provides the list of parameters in the JSON Response.

- **requestInstanceId** - (String). Request Instance ID of the export process
- **downloadlink** - (String). Click the link to download the document.

- **status** - (String). Status of the export

Sample Response

- The following sample response is generated, when the Download report request is in progress.

```
Report Request Accepted
{
  "status": "PENDING",
  "requestInstanceId": "100"
}
Report Request is being processed
{
 "status": "RUNNING",
 "requestInstanceId": "100"}
Report Request Generation Failed
{
 "status": "FAIL",
 "reportInstanceId": "100"}
Report Request Generation Completed  - for higher volume
{
 "status": "COMPLETE",
 "reportInstanceId": "100",
 "downloadLink": "<url>"
}
```

- The following sample response is generated , when the download request is processed and the PAR URL to download the report is included in the response. You can copy the PAR URL and paste it in a Web browser, to download the report in CSV format.

```
"requestInstanceId":"183",
"downloadLink": "<URL>",
"status":"COMPLETE"
```

- The following sample response is generated , when the download request is in pending status.

```
"{"requestInstanceId":"161","status":"PENDING"}
```

11
External Scheduler Interface

External scheduler interface helps you to execute tasks using any command-line utility, such as cURL commands. You won't require the application interface to execute the tasks.

You can also integrate batches with external schedulers using the external scheduler interface.

Related Topics

- Rest API Status Codes
  Refer to the following table for Rest API status codes and their descriptions.

- Execution API
  The Execution (POST) API triggers a batch or a batch group.

- Execution Status API
  The Execution Status (POST) API provides the current run status of batch/batch group execution.

- Interrupt API
  The Execution Status (POST) API Interrupts a batch/batch group execution.

- Restart API
  The Restart (POST) API restarts a batch/batch group execution.

- Rerun API
  The Rerun (POST) API helps to rerun a batch/batch group execution.

Rest API Status Codes

Refer to the following table for Rest API status codes and their descriptions.

Table 11-1   Status Codes

Status Code	Description
0	Success
-1	Failure
-2	Interrupted
1	Not Started
2	Ongoing
3	Aborted
4	Excluded
5	Held
-3	Object does not exist
-4	Invalid arguments passed in request/not enough params in Request body
-5	Invalid request headers/request headers missing
-6	No executable job is present
-7	Job is already interrupted
-8	Job is not ongoing/aborted

Execution API

The Execution (POST) API triggers a batch or a batch group.

- **HTTP Method** - POST
- **URL** - `/SchedulerService/rest-api/v1/external/trigger`
- **Header Parameters**
    - **ofs_tenant_id** - Tenant ID of the Application
    - **ofs_service_id** - Service ID of the Application
    - **ofs_workspace_id** - Workspace ID of the Application. It is defaulted to "WS001" and same should be passed each time.
    - **ofs_remote_user** - Used ID of the user. This parameter should be mapped to `'BATCH_EXEC'` function.
    - **locale** - locale in `languageCode-countryCode` format. For example, `en-US`.
    - **Authorization: Bearer <token>** - Access token required to authenticate the API. If this token is not provided, `401 Unauthorized error` is generated. For more information about Bearer token, refer to Generate the Access Token.
- **Sample cURL Command**

```
curl -i -H "ofs_service_id:<Service ID>" -H "ofs_remote_user:<User ID>" -H
    "ofs_tenant_id:<Tenant ID>" -H "ofs_workspace_id:WS001" -H
"locale:en-US" -H
    "Content-Type: application/json" -H "Authorization: Bearer
<BEARER_TOKEN>" -X POST
    <APPLICATION_BASE_PATH>/<URL> -d '<REQUEST_JSON>'
```

Batch Execution API

Use the Execution API to trigger a batch.

Attributes

- `batchName` - The unique batch code
- `batchType` - The object type. For Batch, the batch type should be set to `rest`.
- `dynamicParamList` - List of run time parameters which should be overridden over actual values for this trigger. This is an optional parameter.

Request Body

```
{
    "batchName": "<Batch_code>",
    "batchType": "rest",
    "dynamicParamList": {
        "batchParams": {
            "$FICMISDATE$": "<MISDate(yyyy-mm-dd)>"
    }
     }
}
```

Sample Response Body

The following Response body is a sample for `Success : 200 OK`. For more information about status code in the response body, refer to Rest API Status Codes.

```
{
    "severity": "info",
    "summary": "Object triggered successfully with Run Id:
B2001_2022-05-20_1653041947296_1",
    "batchRunId": "B2001_2022-05-20_1653041947296_1",
    "details": "Object triggered successfully.",
    "statusCode": "0",
    "status": "success"
}
```

Batch Group Execution API

Use the Execution API to trigger a batch group.

- **Attributes**
    - `batchName` - The unique batch code
    - `batchType` - The object type. For Batchgroup, the batch type should be set to `group`.
    - `dynamicParamList` - List of run time parameters which should be overridden over actual values for this trigger. This is an optional parameter.

Request Body

```
{
    "batchName": "<Batch_group_code>",
    "batchType": "group",
    "dynamicParamList": [
        {
            "batchName": "MIS_BATCH001",
            "batchParams": {
                "$FICMISDATE$": "2022-10-23"
            }
        },
        {
            "batchName": "MIS_BATCH002",
            "batchParams": {
                "$FICMISDATE$": "2022-10-28"
            }
        }
    ]
}
```

Sample Response Body

The following Response body is a sample for `Success : 200 OK`. For more information about status code in the response body, refer to Rest API Status Codes.

```
{
```

```
    "severity": "info",
    "summary": "Object triggered successfully with Run Id:
B2001_2022-05-20_1653041947296_1",
    "batchRunId": "B2001_2022-05-20_1653041947296_1",
    "details": "Object triggered successfully.",
    "statusCode": "0",
    "status": "success"
}
```

Execution Status API

The Execution Status (POST) API provides the current run status of batch/batch group execution.

- **HTTP Method** - POST

- **URL** - `/SchedulerService/rest-api/v1/external/status`

- **Header Parameters**

    – **ofs_tenant_id** - Tenant ID of the Application

    – **ofs_service_id** - Service ID of the Application

    – **ofs_workspace_id** - Workspace ID of the Application. It is defaulted to "WS001" and same should be passed each time.

    – **ofs_remote_user** - Used ID of the user. This parameter should be mapped to `'BATCH_EXEC'` function.

    – **locale** - locale in `languageCode-countryCode` format. For example, `en-US`.

    – **Authorization: Bearer <token>** - Access token required to authenticate the API. If this token is not provided, `401 Unauthorized error` is generated. For more information about Bearer token, refer to Generate the Access Token.

- **Sample cURL Command**

```
curl -i -H "ofs_service_id:<Service ID>" -H "ofs_remote_user:<User ID>" -H
    "ofs_tenant_id:<Tenant ID>" -H "ofs_workspace_id:WS001" -H
"locale:en-US" -H
    "Content-Type: application/json" -H "Authorization: Bearer
<BEARER_TOKEN>" -X POST
    <APPLICATION_BASE_PATH>/<URL> -d '<REQUEST_JSON>'
```

Batch Execution Status API

Use the Execution Status API to view the current run status of a batch execution.

Attributes

- `batchRunId` - Execution ID generated while triggering the object and can be obtained in the response of Execution API.

- `tasks` - List of Task Codes. This is an optional parameter.

Request Body

```
{
    "batchRunId": "<Batchrun_ID>",
    }
```

Sample Response Body

The following Response body is a sample for Success : 200 OK. For more information about
status code in the response body, refer to Rest API Status Codes.

```
{
    "severity": "info",
    "batchRunId": "BatchTEST1_2022-05-31_1653994545003_1",
    "taskStatusList": [
        {
            "taskCode": "t1",
            "taskStatus": "SUCCESSFUL",
            "statusCode": "0"
        },
        {
            "taskCode": "t5",
            "taskStatus": "FAILED",
            "statusCode": "-1"
        }
    ],
    "batchStatusCode": "-1",
    "batchList": [],
    "batchStatus": "FAILED",
    "status": "success",
    "statusCode": "0"
}
```

Batch Group Execution Status API

Use the Execution Status API to view the current run status of a batch group execution.

Attributes

• batchRunId - Execution ID generated while triggering the object and can be obtained in the
  response of Execution API.

• tasks - List of Task Codes. This is an optional parameter.

Request Body

```
{
    "batchRunId": "<Batchrun_ID>",
    }
```

Sample Response Body

The following Response body is a sample for `Success : 200 OK`. For more information about status code in the response body, refer to Rest API Status Codes.

```
{
    "severity": "info",
    "batchRunId": "AbTestBG001_2023-01-27_1674798339245_1",
    "batchStatusCode": "0",
    "batchList": [
        {
            "batchRunId": "AbTestBatch002_2023-01-27_1674798339462_1",
            "batchStatusCode": "0",
            "batchStatus": "SUCCESSFUL"
        },
        {
            "batchRunId": "AbTestBatch003_2023-01-27_1674798339556_1",
            "batchStatusCode": "0",
            "batchStatus": "SUCCESSFUL"
        }
    ],
    "batchStatus": "SUCCESSFUL",
    "status": "success",
    "statusCode": "0"
}
```

Interrupt API

The Execution Status (POST) API Interrupts a batch/batch group execution.

- **HTTP Method** - POST
- **URL** - `/SchedulerService/rest-api/v1/external/interrupt`
- **Header Parameters**
    - **ofs_tenant_id** - Tenant ID of the Application
    - **ofs_service_id** - Service ID of the Application
    - **ofs_workspace_id** - Workspace ID of the Application. It is defaulted to "WS001" and same should be passed each time.
    - **ofs_remote_user** - Used ID of the user. This parameter should be mapped to `'BATCH_EXEC'` function.
    - **locale** - locale in `languageCode-countryCode` format. For example, `en-US`.
    - **Authorization: Bearer <token>** - Access token required to authenticate the API. If this token is not provided, `401 Unauthorized error` is generated. For more information about Bearer token, refer to Generate the Access Token.
- **Sample cURL Command**

```
curl -i -H "ofs_service_id:<Service ID>" -H "ofs_remote_user:<User ID>" -H
     "ofs_tenant_id:<Tenant ID>" -H "ofs_workspace_id:WS001" -H
"locale:en-US" -H
     "Content-Type: application/json" -H "Authorization: Bearer
<BEARER_TOKEN>" -X POST
     <APPLICATION_BASE_PATH>/<URL> -d '<REQUEST_JSON>'
```

Batch Interrupt API

Use the Interrupt API to interrupt a batch execution.

Attributes

* `batchName` - The unique batch code

* `batchRunID` - Execution ID generated while triggering the object and can be obtained in the response of Execution API.

Request Body

```
{
    "batchName": "<Batch_code>",
    "batchRunId": "<Batchrun_ID>"
}
```

Sample Response Body

The following Response body is a sample for `Success : 200 OK`. For more information about status code in the response body, refer to Rest API Status Codes.

```
{
    "severity": "info",
    "batchRunId": "B2001_2022-05-30_1653233511394_1",
    "details": "Execution interrupted successfully.",
    "statusCode": "0",
    "status": "success"
}
```

Batch Group Interrupt API

Use the Interrupt API to interrupt a batch group execution.

Attributes

* `batchName` - The unique batch code

* `batchRunID` - Execution ID generated while triggering the object and can be obtained in the response of Execution API.

Request Body

```
{
    "batchName": "<Batchgroup_code>",
    "batchRunId": "<Batchrun_ID>"
}
```

Sample Response Body

The following Response body is a sample for `Success : 200 OK`. For more information about status code in the response body, refer to Rest API Status Codes.

```
{
    "severity": "info",
    "batchRunId": "B2001_2022-05-30_1653233511394_1",
    "details": "Execution interrupted successfully.",
    "statusCode": "0",
    "status": "success"
}
```

Rerun API

The Rerun (POST) API helps to rerun a batch/batch group execution.

- **HTTP Method** - POST
- **URL** - `/SchedulerService/rest-api/v1/external/rerun`
- **Header Parameters**
    - **ofs_tenant_id** - Tenant ID of the Application
    - **ofs_service_id** - Service ID of the Application
    - **ofs_workspace_id** - Workspace ID of the Application. It is defaulted to "WS001" and same should be passed each time.
    - **ofs_remote_user** - Used ID of the user. This parameter should be mapped to `'BATCH_EXEC'` function.
    - **locale** - locale in `languageCode-countryCode` format. For example, `en-US`.
    - **Authorization: Bearer <token>** - Access token required to authenticate the API. If this token is not provided, `401 Unauthorized error` is generated. For more information about Bearer token, refer to Generate the Access Token.
- **Sample cURL Command**

```
curl -i -H "ofs_service_id:<Service ID>" -H "ofs_remote_user:<User ID>" -H
    "ofs_tenant_id:<Tenant ID>" -H "ofs_workspace_id:WS001" -H
"locale:en-US" -H
    "Content-Type: application/json" -H "Authorization: Bearer
<BEARER_TOKEN>" -X POST
    <APPLICATION_BASE_PATH>/<URL> -d '<REQUEST_JSON>'
```

Batch Rerun API

Use the Rerun API to rerun an existing batch execution.

Attributes

- `batchName` - The unique batch code
- `batchRunID` - Execution ID generated while triggering the object and can be obtained in the response of Execution API.

Request Body

```
{
    "batchName": "<Batch_code>",
    "batchRunId": "<Batchrun_ID>"
}
```

Sample Response Body

The following Response body is a sample for `Success : 200 OK`. For more information about status code in the response body, refer to Rest API Status Codes.

```
Success Scenario: 200 OK
{
    "severity": "info",
    "summary": "Object triggered successfully for rerun with Run Id:
B2001_2022-05-30_1653223084727_1",
    "batchRunId": "B2001_2022-05-30_1653223084727_1",
    "details": "Object triggered successfully.",
    "statusCode": "0",
    "status": "success"
}
```

Batch Group Rerun API

Use the Rerun API to rerun an existing batch group execution.

Attributes

- `batchName` - The unique batch code

- `batchRunID` - Execution ID generated while triggering the object and can be obtained in the response of Execution API.

Request Body

```
{
    "batchName": "<Batchgroup_code>",
    "batchRunId": "<Batchrun_ID>"
}
```

Sample Response Body

The following Response body is a sample for `Success : 200 OK`. For more information about status code in the response body, refer to Rest API Status Codes.

```
Success Scenario: 200 OK
{
    "severity": "info",
    "summary": "Object triggered successfully for rerun with Run Id:
B2001_2022-05-30_1653223084727_1",
    "batchRunId": "B2001_2022-05-30_1653223084727_1",
    "details": "Object triggered successfully.",
    "statusCode": "0",
```

```
            "status": "success"
        }
```

Restart API

The Restart (POST) API restarts a batch/batch group execution.

- **HTTP Method** - POST

- **URL** - `/SchedulerService/rest-api/v1/external/restart`

- **Header Parameters**

    – **ofs_tenant_id** - Tenant ID of the Application

    – **ofs_service_id** - Service ID of the Application

    – **ofs_workspace_id** - Workspace ID of the Application. It is defaulted to "WS001" and same should be passed each time.

    – **ofs_remote_user** - Used ID of the user. This parameter should be mapped to `'BATCH_EXEC'` function.

    – **locale** - locale in `languageCode-countryCode` format. For example, `en-US`.

    – **Authorization: Bearer <token>** - Access token required to authenticate the API. If this token is not provided, `401 Unauthorized error` is generated. For more information about Bearer token, refer to Generate the Access Token.

- **Sample cURL Command**

```
curl -i -H "ofs_service_id:<Service ID>" -H "ofs_remote_user:<User ID>" -H
      "ofs_tenant_id:<Tenant ID>" -H "ofs_workspace_id:WS001" -H
"locale:en-US" -H
      "Content-Type: application/json" -H "Authorization: Bearer
<BEARER_TOKEN>" -X POST
      <APPLICATION_BASE_PATH>/<URL> -d '<REQUEST_JSON>'
```

Batch Restart API

Use the Restart API to restart a batch execution.

Attributes

- `batchName` - The unique batch code

- `batchRunID` - Execution ID generated while triggering the object and can be obtained in the response of Execution API.

Request Body

```
{
    "batchName": "<Batch_code>",
    "batchRunId": "<Batchrun_ID>"
}
```

Sample Response Body

The following Response body is a sample for `Success : 200 OK`. For more information about status code in the response body, refer to Rest API Status Codes.

```
{
    "severity": "info",
    "summary": "Object triggered successfully for restart with Run Id:
B0001_2022-04-30_1651731208588_1",
    "batchRunId": "B0001_2022-04-30_1651731208588_1",
    "details": "Object triggered successfully.",
    "statusCode": "0",
    "status": "success"
}
```

Batch Group Restart API

Use the Restart API to restart a batch group execution.

Attributes

* `batchName` - The unique batch code

* `batchRunID` - Execution ID generated while triggering the object and can be obtained in the response of Execution API.

Request Body

```
{
    "batchName": "<Batchgroup_code>",
    "batchRunId": "<Batchrun_ID>"
}
```

Sample Response Body

The following Response body is a sample for `Success : 200 OK`. For more information about status code in the response body, refer to Rest API Status Codes.

```
{
    "severity": "info",
    "summary": "Object triggered successfully for restart with Run Id:
B0001_2022-04-30_1651731208588_1",
    "batchRunId": "B0001_2022-04-30_1651731208588_1",
    "details": "Object triggered successfully.",
    "statusCode": "0",
    "status": "success"
}
```