Oracle® Retail Dynamic Data Service
Guide

Release 24.0.101.0
F90471-01
January 2024

Oracle Retail Dynamic Data Service Guide, Release 24.0.101.0

F90471-01

Copyright © 2024, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Send Us Your Comments

Preface

Audience vii
Customer Support vii
Review Patch Documentation vii
Improved Process for Oracle Retail Documentation Corrections vii
Oracle Help Center (docs.oracle.com) viii
Conventions viii

1 Introduction

What is DDS? 1-1
Need for DDS 1-1
Functionality Offered 1-1
Accessibility 1-2

2 RESTful Services

Discover 2-1
Get Data 2-1
Get Data from a Table 2-1
Insert Data 2-2
Update Data 2-2
Delete Data 2-2
Get Schemas 2-3
Get Tables 2-3
Get Columns for a Table 2-3
Get System Options 2-3
Create System Options 2-3
Update System Options 2-3
Delete System Options 2-4

Reset Cache 2-4

3 Security

Table Level Security 3-2
IDCS Security Setup 3-2
Create OAuth2 Client Application in IDCS 3-2

4 User Interface

Admin Tab Guide 4-1
Initial Permissions Setup 4-1
Manage Access Levels 4-1
Manage Security Groups 4-2
Manage Security Permissions 4-3
Designer Tab Guide 4-5
Table Definition 4-5
Query Builder 4-6

5 Advanced Backend Features

LIKE Statement 5-1
IN Statement 5-2
Comparative Operators 5-2
Filtering DATE Column 5-3
Combinations of OR, AND & BETWEEN Statements 5-4
Viewing Clob Data 5-5
Viewing Query Response as JSON Data 5-5
JOINing Two or More Tables 5-6
Count of Records 5-7

List of Tables

3-1 Functions Performed by Roles and Groups 3-1

Send Us Your Comments

Oracle Retail Dynamic Data Service Guide, Release 23.1.401.0

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

> **Note:**
>
> Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

The *Dynamic Data Service (DDS) Guide* describes the integration and configuration information for Oracle Retail Dynamic Data Service.

Audience

This guide is for:

- Systems administration and operations personnel
- Systems analysts
- Integrators and implementers
- Business analysts who need information about Product processes and interfaces

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 19.0.000) or a later patch release (for example, 19.0.001). If you are installing the base release and additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be

attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

Oracle Retail product documentation is available on the following web site:

https://docs.oracle.com/en/industries/retail/index.html

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

Oracle Help Center (docs.oracle.com)

Oracle Retail product documentation is available on the following web site:

https://docs.oracle.com/en/industries/retail/index.html

(Data Model documents can be obtained through My Oracle Support.)

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

This chapter gives a brief introduction to the Dynamic Data Service Web Application.

What is DDS?

Dynamic Data Service (DDS) is a cloud based web application with functionality to remotely interact with databases. It has the ability to interact with many schemas at once in a database. The functionality also has the ability to interact with different databases as well.

Data can be viewed and modified through the application. It also has security built into it to restrict data access and modification.

Dynamic Data Service provides access to data in a database through RESTful services. Users can perform CRUD operations on the data using RESTful services. From a security perspective, access to data can be restricted to users at the table, column and row level.

Dynamic Data Service allows users to access data from any configured databases the time of install. Dynamic Data Service UI provides functionality to setup the security and perform CRUD operations on the data.

Need for DDS

DDS is useful for exposing data in a schema with data that is not directly visible to the user. Good examples for this would be a back-end schema of other Oracle applications. Any data that is not exposed by these applications can be viewed using DDS because users cannot connect to database schemas, from a cloud environment, using standard database tools without proper permissions.

Functionality Offered

DDS offers various functionality regarding databases. It can connect to various database connections. The application then discovers the schema inside the databases and allows the user to work and interact with the tables and data within the tables. The high level list of functionalities offered:

- Schema Discovery
- Viewing Table Definition
- Querying table data
- Creation, modification and deletion of records
- Bulk Updates and Deletion of data in tables
- Security setup for limiting data access and modification

Accessibility

Oracle JET components have built-in accessibility support that conforms with the Web Content Accessibility Guidelines version 2.0 at the AA level (WCAG 2.0 AA), developed by the World Wide Web Consortium (W3C).

Accessibility involves making your application usable for persons with disabilities such as low vision or blindness, deafness, or other physical limitations. This means, for example, creating applications that can be:

- Used without a mouse (keyboard only).
- Used with assistive technologies such as screen readers and screen magnifiers.
- Used without reliance on sound, color, animation, or timing.

DDS provides the ability to support the above accessibility in the applications.

Users should be able to navigate to all parts and functions of the application using the Tab and arrow keys, without using any keyboard shortcuts. In addition to that, keyboard shortcuts merely provide an additional way to access a function quickly.

Keyboard shortcuts provide an alternative to pointing devices for navigating the page. There are five types of keyboard shortcuts that can be provided in OJET applications:

- Tab traversal, using Tab and Shift+Tab keys: Moves the focus through UI elements on a screen.
- Accelerator keys (hot keys): bypasses menu and page navigation, and performs an action directly, for example, Ctrl+C for Copy.
- Access keys: Moves the focus to a specific UI element, for example, Alt+F for the File menu.
- Default cursor/focus placement: Puts the initial focus on a component so that keyboard users can start interacting with the page without excessive navigation.
- Enter key: Triggers an action when the cursor is in certain fields or when the focus is on a link or button.

2
RESTful Services

This chapter provides information about the RESTful Services used by DDS.

Discover

This end point returns a list of Dynamic Data Service end points.

HTTP Operation: GET

Path: **/resources/discover**

Get Data

This end point returns data from the provided schema and table(s). The query parameters "fromTables" allows single or multiple tables.

The query parameter "rowFilter" can be used to provide a join between tables.

Tables and columns need to be qualified if multiple tables are provided in "fromTables". The data is returned in JSON/XML format and one page of data is returned. The response contains a link to the next page if there are additional pages.

HTTP Operation: GET

Path: **/resources/dds/{schemaName}/data?fromTables=<tables>**

The following query parameters can be provided to filter the data.

- columnFilter - Columns to be included in the response
- rowFilter - Predicate can be provided in the rowFilter to filter the data
- sortBy - Valid values are ascending(asc) or descending(desc)
- Page - Page number
- pageLimit - Number of records to be included in the page. The default page size is 25
- format - Valid values are short or long. Short format provides data without column names. Long format provides data with column names.

Get Data from a Table

This end point returns data from a table using a primary key.

HTTP Operation: GET

Path: **/resources/dds/{schemaName}/{table}?idFilter=<Primary Key>**

Insert Data

This end point inserts data in a table. Input can be a single record or multiple records provided in JSON format. Users that belong to the admin or operator group can only perform this operation. Users also need to have DataCreatePermission to insert data.

HTTP Operation: PUT

Path: **/resources/dds/{schemaName}/data/{table}**

Sample input for inserting data in the BDI_RECEIVER_OPTIONS table

/dds/bdi_rms_schema/data/BDI_RECEIVER_OPTIONS

```
{
    "items":
 [
        {
            "ID":1,
            "BASE_FOLDER":"base",
            "FOLDER_TEMPLATE":"baseTemplate",
            "INTERFACE_MODULE":"interfaceModule",
            "INTERFACE_SHORT_NAME":"interfaceShortName",
            "MERGE_STRATEGY":"mergeStrategy"
        }
    ]
}
```

Update Data

This end point updates data in a table using the rowFilter query parameter and input is provided in JSON format. Users that belong to the admin or operator group can only perform this operation. Users also need to have DataUpdatePermission to update data.

HTTP Operation: POST

Path: **/resources/dds/{schemaName}/data/{table}?rowFilter=<predicate>**

Sample input for updating data in the BDI_SYSTEM_OPTIONS table

/dds/bdi_rms_schema/data/BDI_SYSTEM_OPTIONS?rowFilter=VARIABLE_NAME='LOADJOBDEF'

```
{
    "VARIABLE_VALUE":"TRUE"
}
```

Delete Data

This end point deletes data from a table based on the rowFilter query parameter. Users that belong to the admin or operator group can only perform this operation. Users also need to have DataDeletePermission to delete data.

HTTP Operation: DELETE

Path: **/resources/dds/{schemaName}/data/{table}?rowFilter=<predicate>**

Sample End Point

/dds/bdi_rms_schema/data/BDI_SYSTEM_OPTIONS?rowFilter=VARIABLE_NAME='LOADJOBDEF'

Get Schemas

This end point returns all configured database schemas.

HTTP Operation: GET

Path: **/resources/dds/schemas**

Get Tables

This end point returns all tables from a database schema.

HTTP Operation: GET

Path: **/resources/dds/{schemaName}/tables**

Get Columns for a Table

This end point returns columns for a table.

HTTP Operation: GET

Path: **/resources/dds/{schemaName}/{table}**

Get System Options

This end point returns all system options from the DDS_SYSTEM_OPTIONS table.

HTTP Operation: GET

Path: **/resources/system-setting/system-options**

Create System Options

This end point creates a system option in the DDS_SYSTEM_OPTIONS table.

HTTP Operation: PUT

Path: **/resources/system-setting/system-options**

Sample Input

```
{
    "key":"testKey",
    "value":"testValue"
}
```

Update System Options

This end point updates a system option in the DDS_SYSTEM_OPTIONS table.

HTTP Operation: POST

Path: **/resources/system-setting/system-options**

Sample Input

```
{
    "key":"testKey",
    "value":"testValue1"
}
```

Delete System Options

This end point deletes a system option in the DDS_SYSTEM_OPTIONS table.

HTTP Operation: DELETE

Path: **/resources/system-setting/system-options/{key}**

Reset Cache

This end point resets the system option cache.

HTTP Operation: POST

Path: **/resources/system-setting/reset-cache**

3
Security

Dynamic Data Service provides table level, row level, and column level security. All the end points are protected with basic authentication and role based authorization.

There are three security groups that provide role based authorization.

There are three Roles and three Groups.

Roles:

- **AdminRole** - Users with this role have access to all the functions of the DDS app. They can also setup the security permissions for other users.
- **OperatorRole** - Users with this role have the ability to read, write and modify content in the schemas and tables. However they will not have access to the admin functions and cannot setup security permissions.
- **MonitorRole** - Users with this role can only read the data from schemas and tables. They also will not have access to security setup functions.

Groups:

- **RicsAdminGroup** - Users that belong to this group can perform all operations
- **RicsOperatorGroup** - Users that belong to this group can perform all operations except security setup
- **RicsMonitorGroup** - Users that belong to this group can only perform read only operations

The following table lists all the functions which can be performed by the roles and groups mentioned above.

Table 3-1 Functions Performed by Roles and Groups

Role Name	Admin Role	Operator Role	Monitor Role
Group Name	**RicsAdminGroup**	**RicsOperatorGroup**	**RicsMonitorGroup**
Create Access Level	Yes	No	No
Delete Access Level	Yes	No	No
Create Security Group	Yes	No	No
Delete Security Group	Yes	No	No
Create Table Level Security	Yes	No	No
Delete Table Level Security	Yes	No	No
View Table Definition	Yes	Yes	Yes
Create Queries	Yes	Yes	Yes
Run Queries	Yes	Yes	Yes

Table 3-1 (Cont.) Functions Performed by Roles and Groups

Role Name	Admin Role	Operator Role	Monitor Role
Group Name	**RicsAdminGroup**	**RicsOperatorGroup**	**RicsMonitorGroup**
View Table Data	Yes	Yes	Yes
Modify Table data	Yes	Yes	No
Delete Table Data	Yes	Yes	No
Bulk Update Table Data	Yes	Yes	No

Table Level Security

Table level security defines who can access a table with set of allowed permissions. A table cannot be accessed if table level security is not setup.

Table level security is associated with the following information.

- Schema Name
- Table Name
- Security Group - Provides who can access the table
- Access Level - Provides permissions

Table level securities are set up for all tables when default security is setup using the end point. Every table is setup to be accessed by users in all security groups with default permissions.

Example

Table - DdsAdminGroup, DdsAdminAccessLevel

Table - DdsOperatorGroup, DdsOperatorAccessLevel

Table - DdsMonitorGroup, DdsMonitorAccessLevel

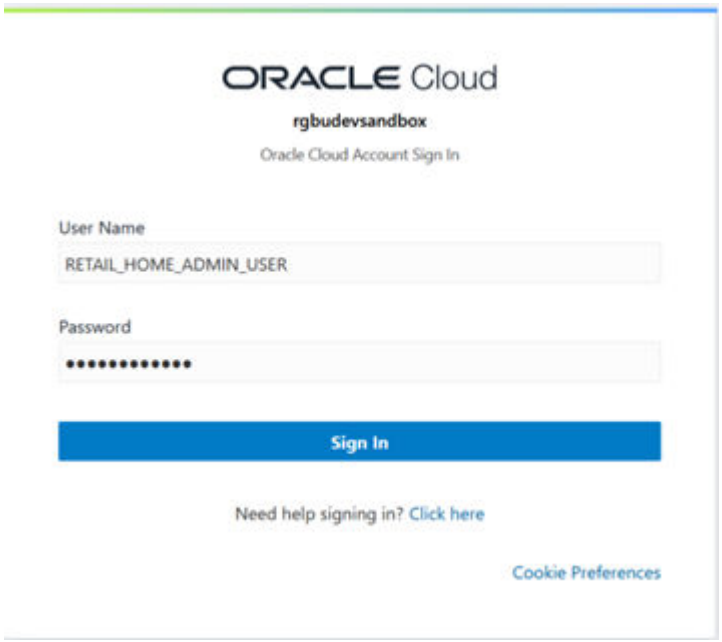
IDCS Security Setup

For getting access to RICS DDS REST services you need to create a client app in IDCS. IDCS app generates an access token that will be used for making publishing service calls. Follow steps for creating the client app in IDCS.

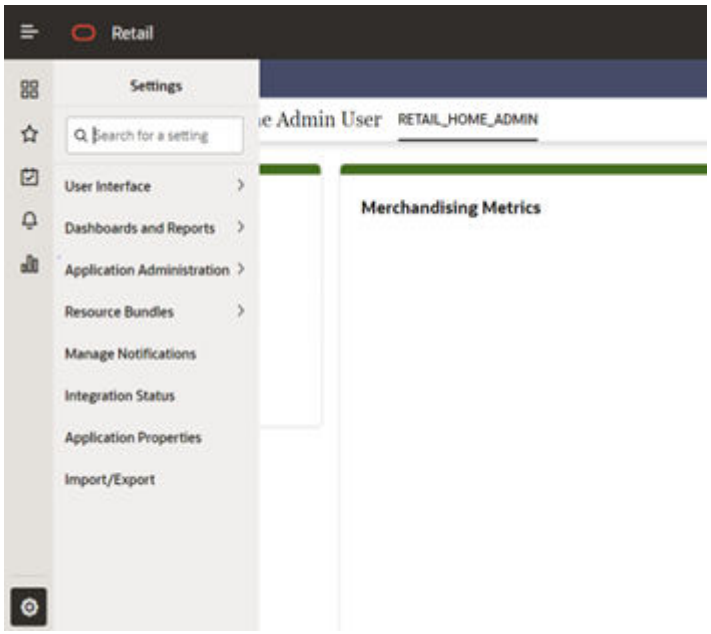
Create OAuth2 Client Application in IDCS

Use Retail Home for creating the client app in IDCS. Once app is created you will get cli-ent id and client secret both of them necessary to get access token. Follow the instructions below for generating the access token and making service call using OAuth2 token.

1. Login into retail home as retail home administrator.

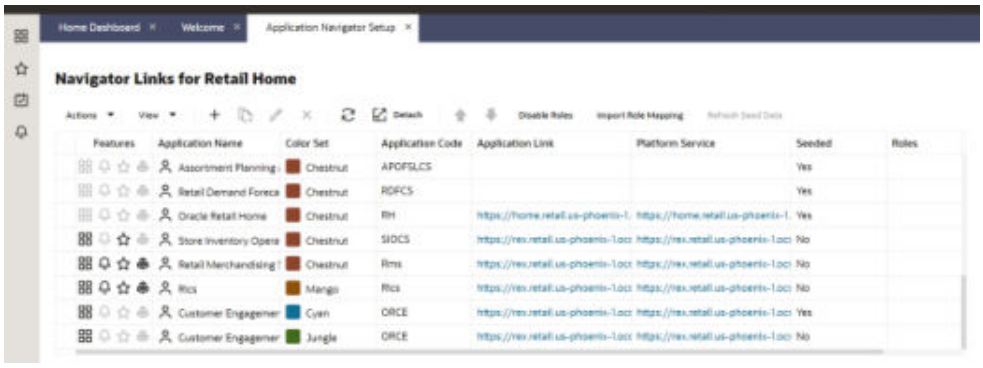


2. In retail home screen click on Settings menu icon on the left and then click on Application Administration.



3. On the Application Administration menu click on Application Navigator Setup. Notice all the hosted applications are listed here with their application and platform service url.

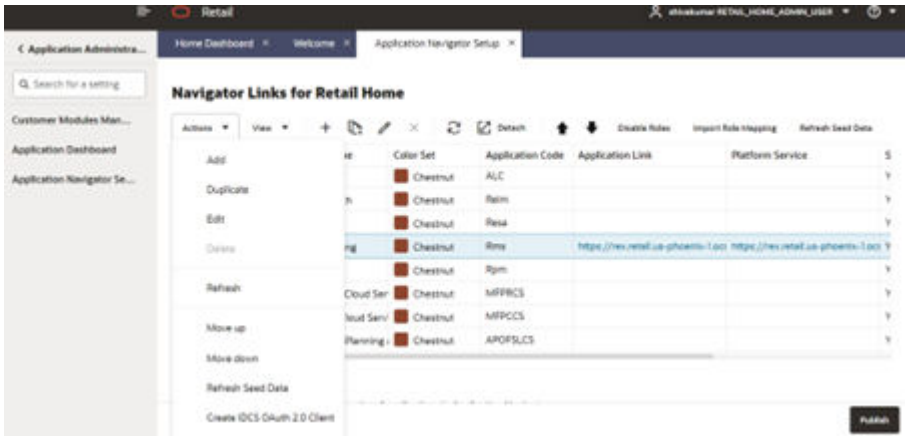
   Settings -> Application Administration->Application Navigator Settings



4. Look for application with name RICS. If you are not seeing RICS, application try refreshing seed. Steps
   a. Select the row with the application code as Rms.
   b. Click the Refresh Seed Data button on top right corner of the menu.
   c. Wait for some time and refresh the screen.
   d. RICS should reflect now.

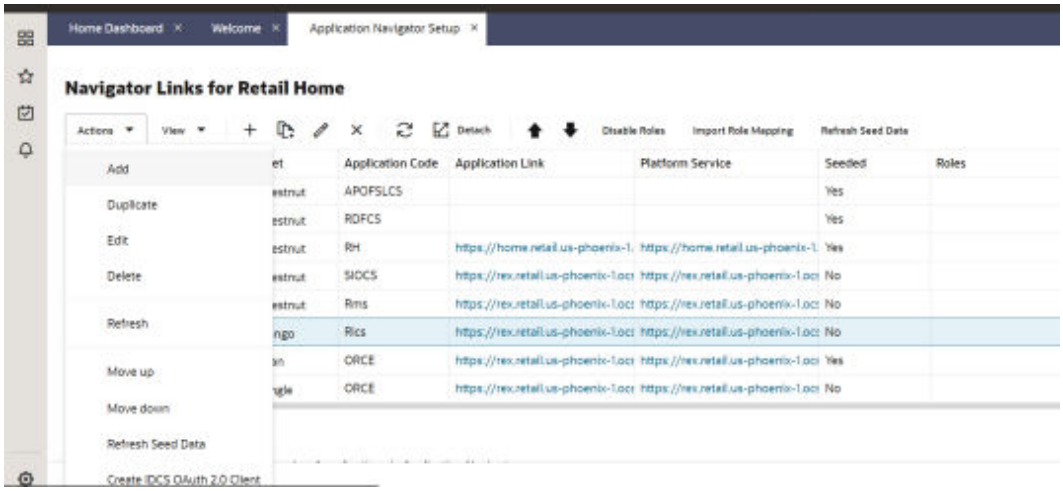
5. If RICS application is not reflecting even after following step 4. Select the row with the application code as RMS and click on the Actions menu on top left. Select Create IDCS OAuth 2.0 Client. A dialog will open for entering oauth2 client details.

> **Note:**
>
> Create IDCS OAuth 2.0 Client option is available only for applications those have platform service URL mentioned. RICS is making use of Merch platform service as both the apps are sharing same IDCS tenancy.



6. Skip this step if RICS application is not showing up. One of either Step 5 or Step 6 needs to be followed.

   Select a row with application code as RICS. Click on the Actions menu on top left and select Create IDCS OAuth 2.0 Client. A dialog will open for entering oauth2 client details.



7. This dialog takes the following values:

   - **App Name** is 2-100 characters and will be used as the name in IDCS. Provide unique application name.
   - **Description** is a detailed description of the application.
   - **Scope:** <Custom environment-specific scope>

     The scope pattern that is used in the RICS IDCS app creation template is rgbu:rics:<SERVICETYPE>-<ENVIRONMENT> where SERVICETYPE is RICS and EN-VIRONMENT is the environment type (STG, PRD, UAT, DEV1, DEV2, and so on).

     For example:

     ```
     "scope": "rgbu:rics:RICS-PRD""scope": "rgbu:rics:RICS-STG"
     ```

8. When the application is created, another dialog will open to show the client ID and client secret of the new application. These values should be copied down to a safe location, as they will only be shown once. Retail Home cannot retrieve the credentials again after the dialog is closed.

9. The Client ID and Client Secret from previous step will be used for generating access token.

   Sample code for generating Access Token:

```
clientId=RICS_TEST_APPID
clientSecret=998e1e1d-f146-45a5-a9a1-99785e3ebf43
idcsUrl=https://
idcs-234e8f7334564936aa0ed93f2c39e9ca.identity.pint.oc9qadev.com
scope=rgbu:rics:RICS-STG99
ec=$(echo -n "$clientId:$clientSecret" | base64 -w 0)

AccessToken=$(curl -iv \
-H "Authorization: Basic $ec" \
-H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" \
--request POST $idcsUrl/oauth2/v1/token \
-d "grant_type=client_credentials&scope=$scope" | grep -o -P '(?
<=access_token":").*(?=","token_type)')

echo $AccessToken
```

10. Now a service call can be made by passing along the access token generated in previous step.

    Here is sample curl command with Bearer token and the DDS get schemas endpoint:

```
ddsGetSchema-sUrl=https://rex.retail.us-phoenix-1.ocs.oc-test.com:443/rgbu-rex-eit-
stg99-rics/dynamic-data-service-web/resources/dds/schemas
curl -ivkL --noproxy '*' -H "Authorization: Bearer $AccessToken"  -H "Con-tent-
Type: application/xml" -X GET $ddsGetSchemasUrl
```

Sample Response:

```
{
    "schemas": [
        "DDS_RIBEH",
        "RFI_TEST"
    ],
    "userInRoles": [
        "AdminRole"
    ]
}
```

4
User Interface

The DDS UI is divided into two sections; Admin tab, and Designer tab. The Admin tab lets admin users control access and perform security operations. The Designer tab lets admin, operators and monitor users perform authorized functions.

This chapter explains the main UI functionality available to the user.

Admin Tab Guide

This tab has all the functions for configuring security related permissions and user access. Only the admin user has access to this tab. The admin can enable read, write and modify access permissions and also limit functionality of a user.

The Admin Tab has three main sections:

- Manage Access Levels
- Manage Security Groups
- Manage Security permissions

Initial Permissions Setup

In order to access the data on tables being viewed in the Designer Tab, first the Table level securities have to be setup in the Admin tab of the DDS Application. The Admin can select the desired database schema from the associated database schema dropdown.

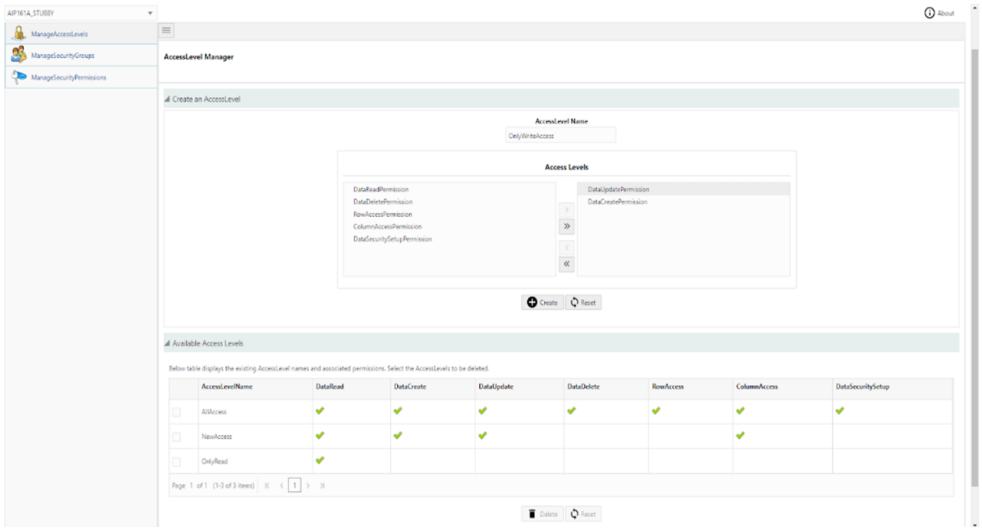
Access to this tab is only allowed to the Admin User.

Manage Access Levels

This tab allows the creation and deletion of Access Levels for data and functionality. These can be assigned to groups for particular tables in a schema.

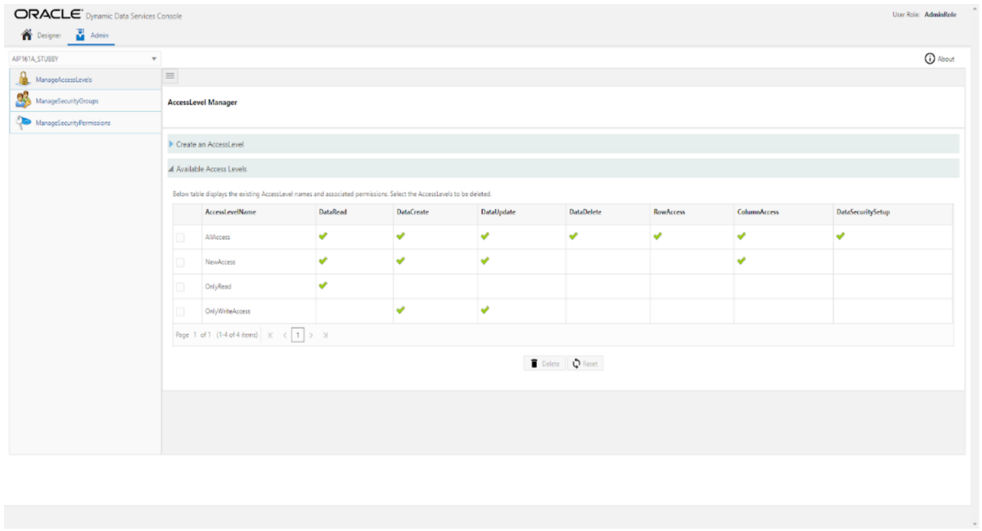
To create an access level

1. Input a name in the Name text box.
2. Select the necessary access permissions.
3. Click **Create**.



To delete an access level

1. Select the Access Level to be deleted from the table which displays the available access levels.
2. Click **Delete**.

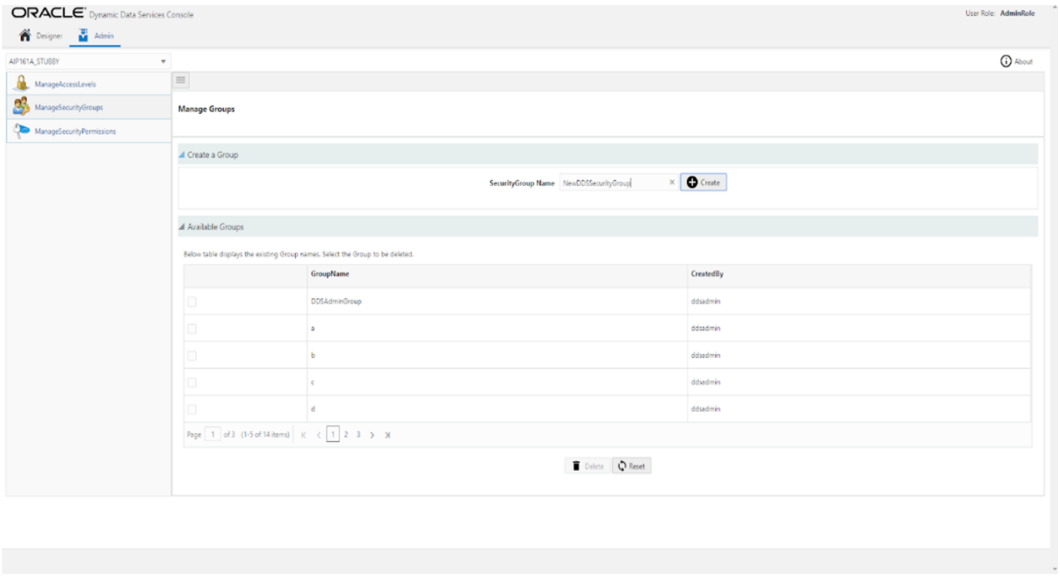


Manage Security Groups

This tab includes functionality for the creation and deletion of Security Groups. Security Groups are groups of users that have a certain level of data access. These, along with one or more access levels, can be assigned to a table to limit access to data and functionality in the Designer tab.

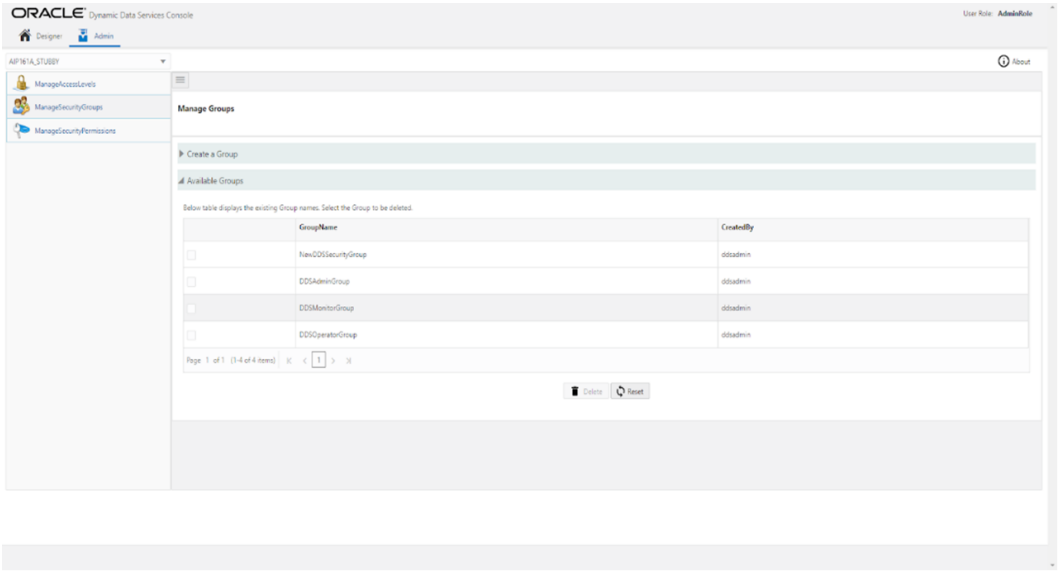
To create a Security Group

1. Input a name in the Name text box.
2. Click **Create**.



To delete a Security Group

1. Select the Security Group to be deleted from the table that displays the available Security Groups.
2. Click **Delete**.



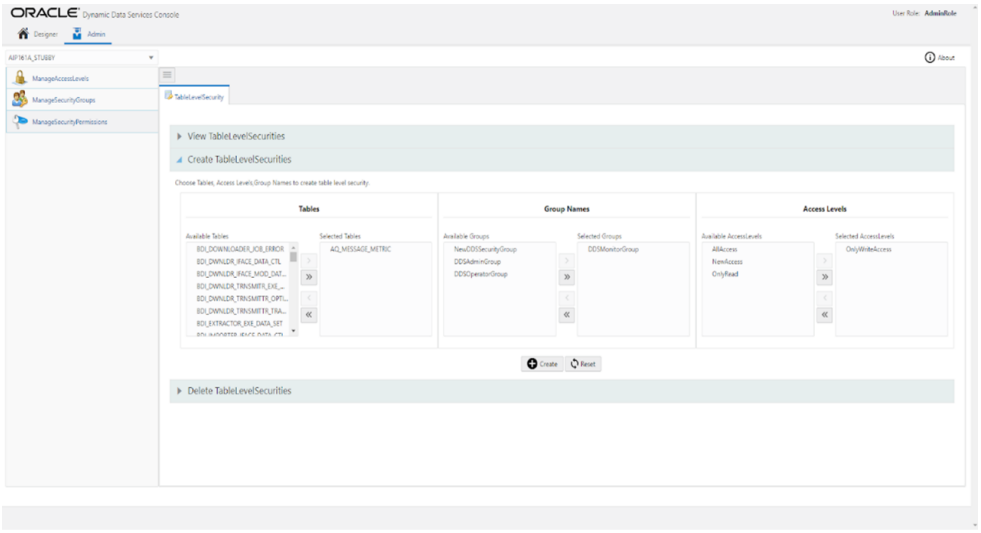
Manage Security Permissions

This tab allows the user to create and delete Security Permissions. Security Permissions are a combination of one or more Access levels in a Security Group mapped to a table to restrict data access and also limit functionality, like removing the ability to create, modify or delete data.

To create a Security Permission for a table

1. In the Create TableLevelSecurities collapsible pane, select the desired table.

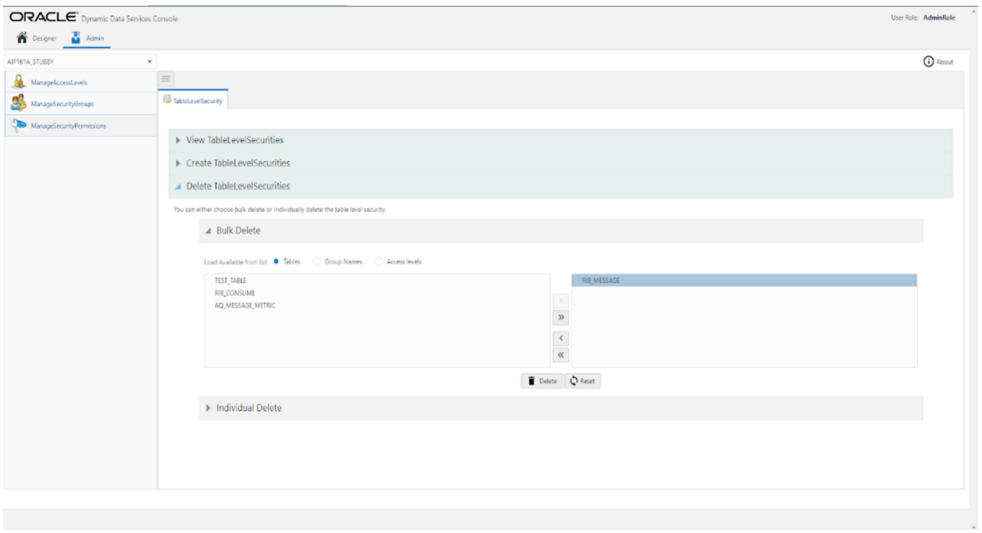
2. Then select the required security group.
3. Finally select the required access level(s).
4. Click **Create**.



To delete Security Permission(s)

Functionality is provided to either perform bulk delete or individual deletion of security.

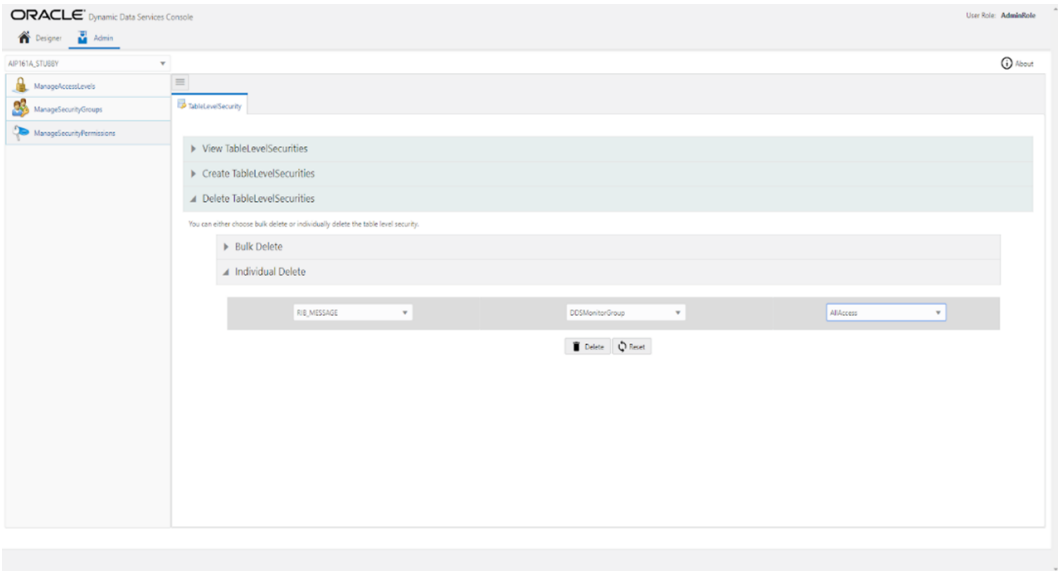
1. To bulk delete security permissions:
   a. Select the required table name(s) or security group(s) or access level(s).
   b. Click **Delete**.



2. To individually delete security permissions:
   a. Select the required table name.
   b. Select the particular security group associated.

c. Select the associated access level.

d. Click **Delete**.



Designer Tab Guide

This tab contains all the CRUD operations to be performed by user on tables in the database schema. The user can select the desired database schema available in the database schema dropdown. It has two sections:

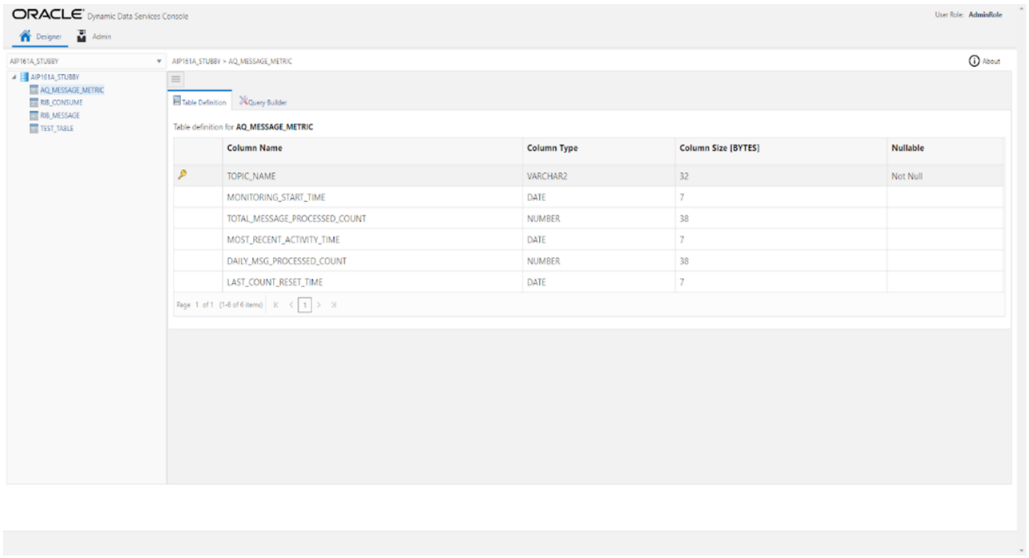
- Table Definition
- Query Builder

Make sure the user has sufficient permissions before performing any action in the designer tab.

Table Definition

This tab allows the user to see the basic definition of the Database table in a simplified format presented in a tabular form.

To view a table's definition, select the required table from the list of tables from the left pane of the screen. Clicking on the table name will load the definition on the display screen under the Table Definition tab.



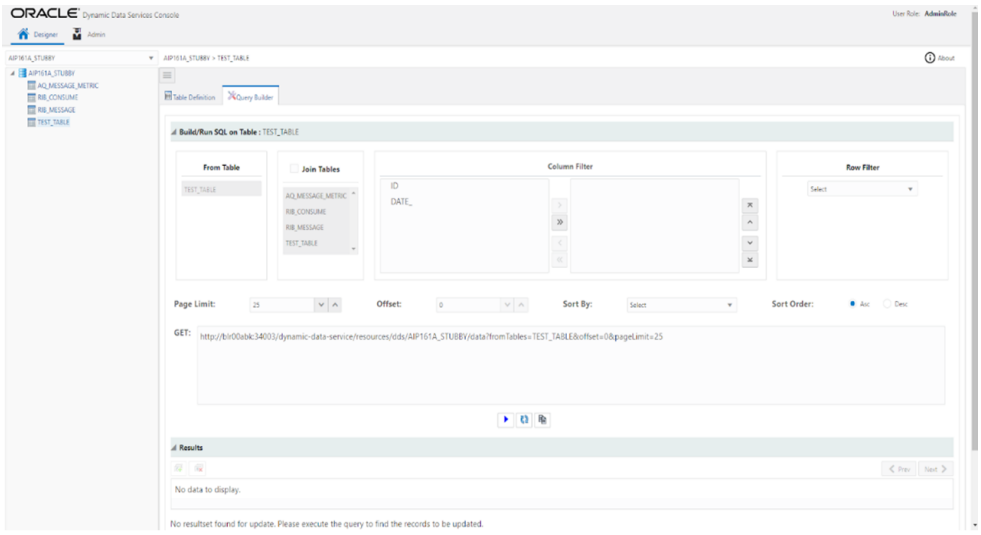
Query Builder

This tab holds the functionality for data manipulation. Functions include Querying records to creation, modification and deletion of records.

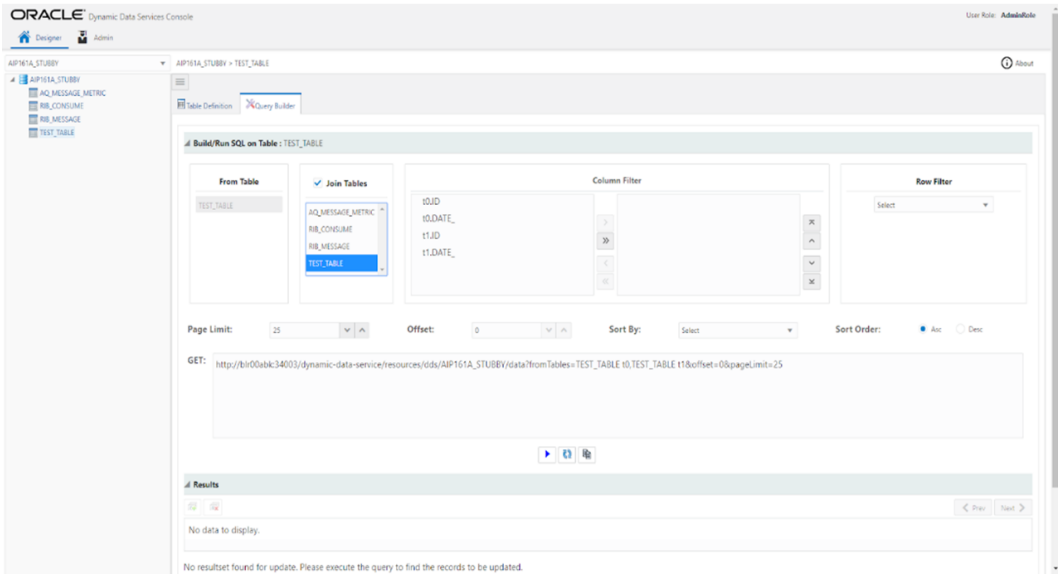
Data Selection

To Query records from a table:

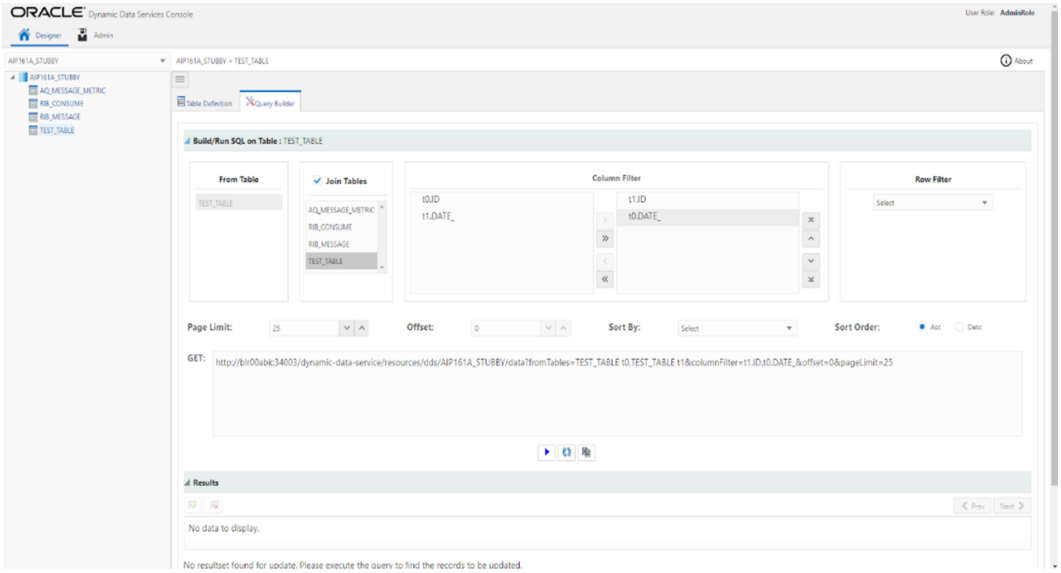
1. Select the table from the list of tables on the left pane of the screen.



2. Enable join if the join operation is necessary.
3. Select the table to which the join is to be done, select the same table as selected in (1) to do a self-join.

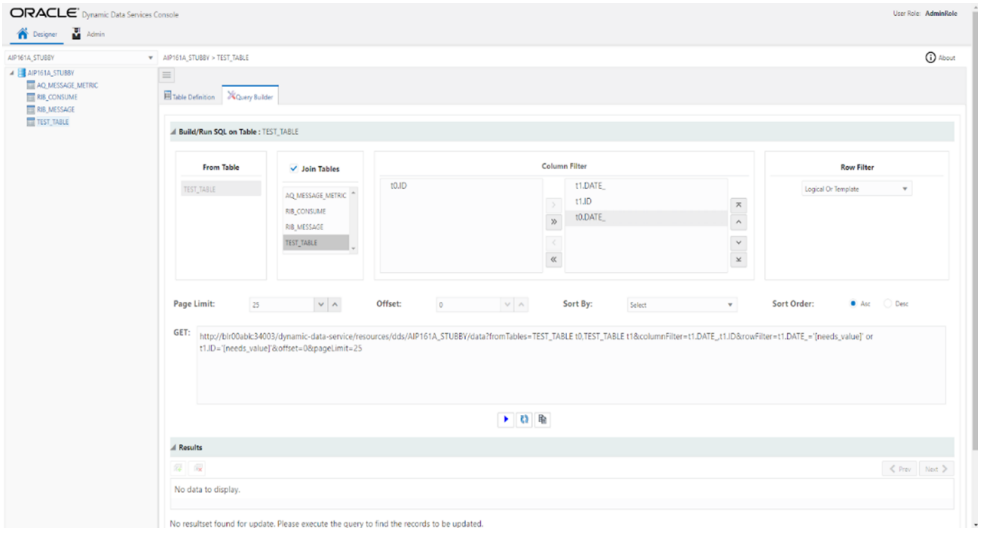


4. Select the columns which are to be displayed in the column filter. Only the column names on the right side of the box will be displayed. If all columns are to be displayed, do not move any columns to the right box.

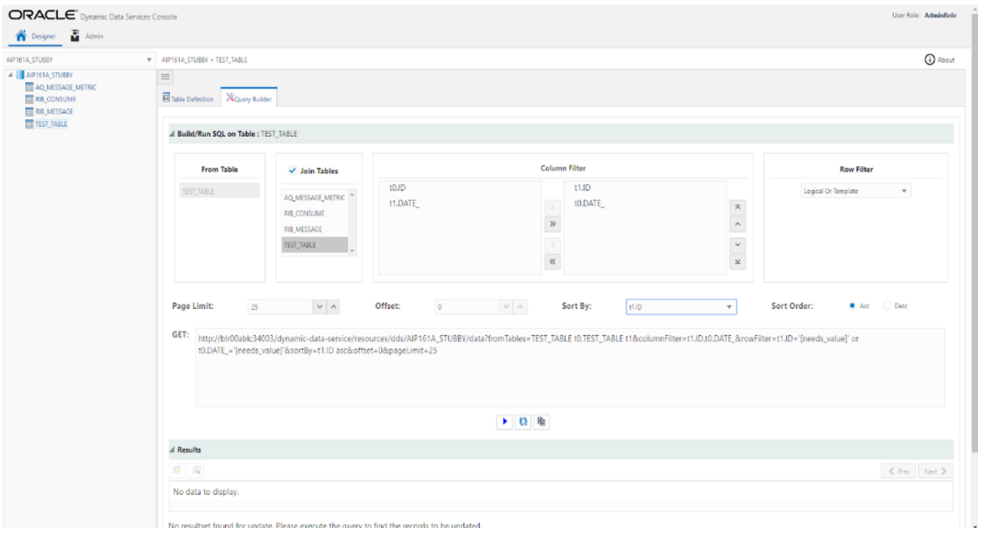


5. Select the row filter.

> **Note:**
>
> The row filter is applied only to the first two columns in the table, so reorder the columns such that the ones that have to be used as row filter are in the two positions.



6. Set the page limit, this is the number of records that are to be displayed on one page, the remaining records will be paginated and displayed upon request.
7. The Sort by drop down list works if there are columns selected for a column filter. Select the desired column, and then select the sort order, ascending or descending.

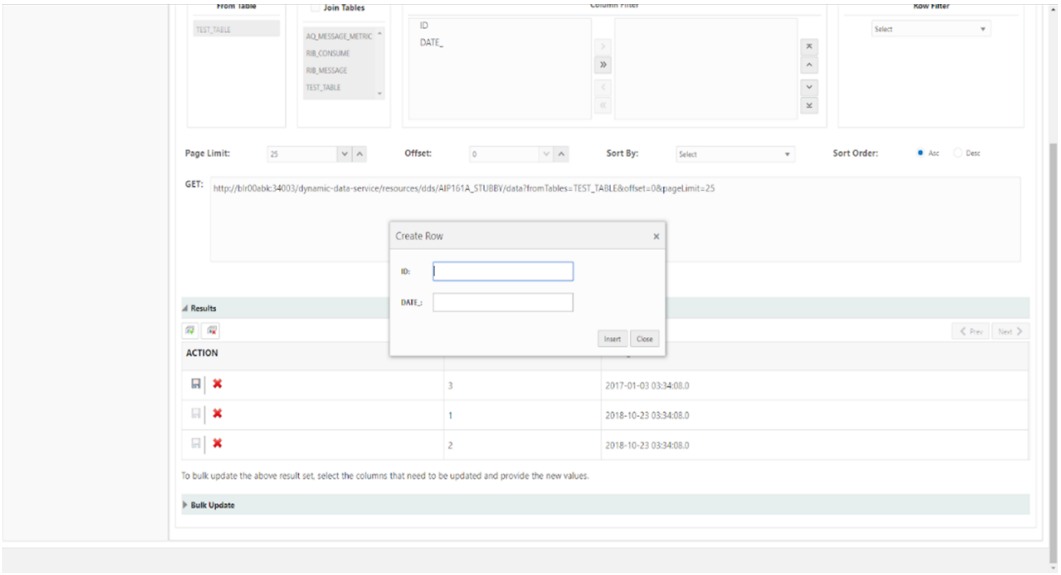


8. Click on the run button to get the Query results using the settings configured earlier.
9. The reset button sets the settings back to default.
10. The copy query button is used to copy the GET query generated in the "GET" box. Click on the button and the link is copied. Paste the link in a browser and press enter/return to execute the query.

Data Insertion

To insert the data in the table:

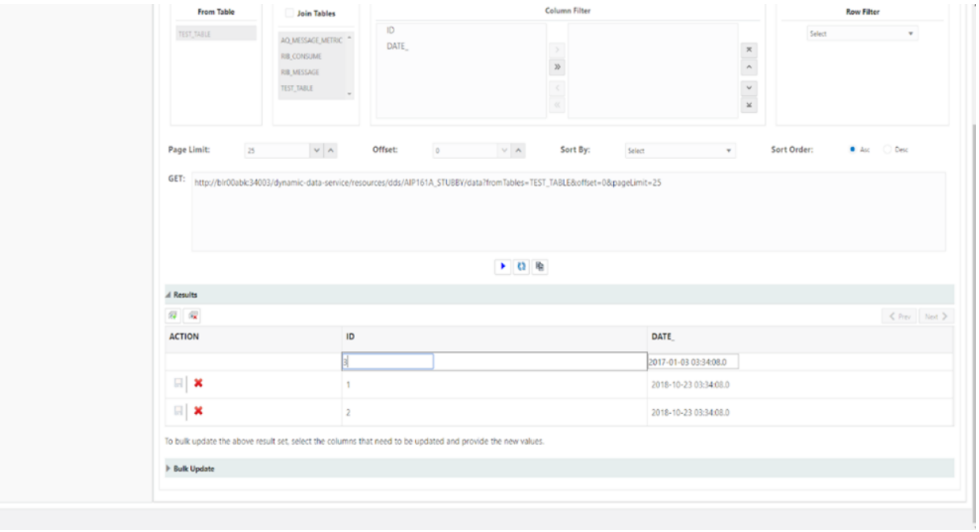
1. The query results from the GET call are displayed here in an editable tabulated format.
2. To insert a record click on the insert record button .
3. In the pop-up, enter the data. Make sure to enter all values for not null columns.
4. Click **Insert**.
5. The record will be successfully inserted after validation.



Data Modification

To modify existing data, select the data from desired table.

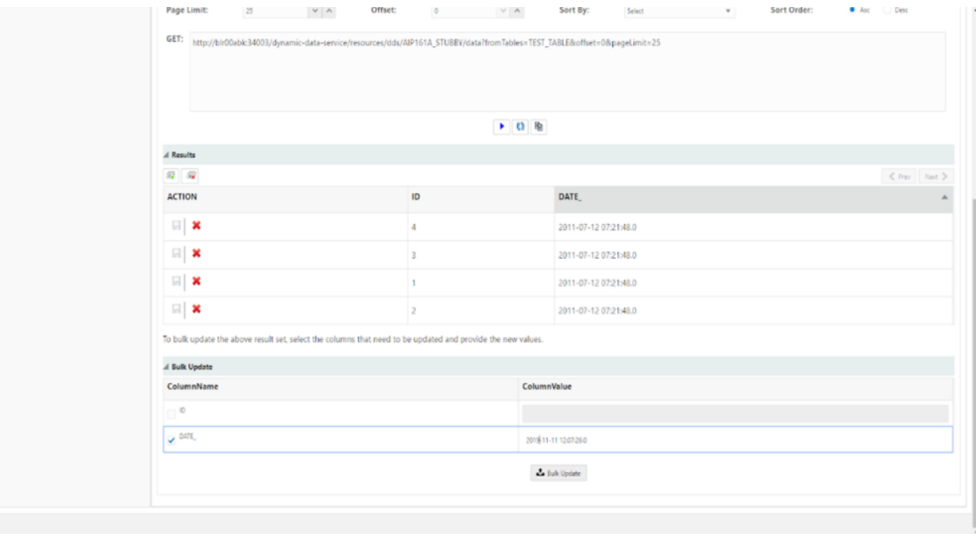
1. Double click on the record to be modified
2. Edit the contents
3. Press escape on the keyboard to exit edit mode and click on the save button to save the changes.
4. Message - Record updated successfully will be displayed after validation

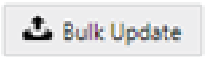


Bulk Updations of Data

The option is available to update bulk records on selected data. To bulk update column(s):

1. In the bulk update collapsible pane one or more columns for all records in the queried table can be updated with a single value, if the column is not unique.
2. Select the checkbox of the required column to enable edit mode.
3. Enter the data as necessary.



4. Click the Bulk Update option 

to update the selected columns for all the records in the table.
5. Records will be updated after successful validation of data.

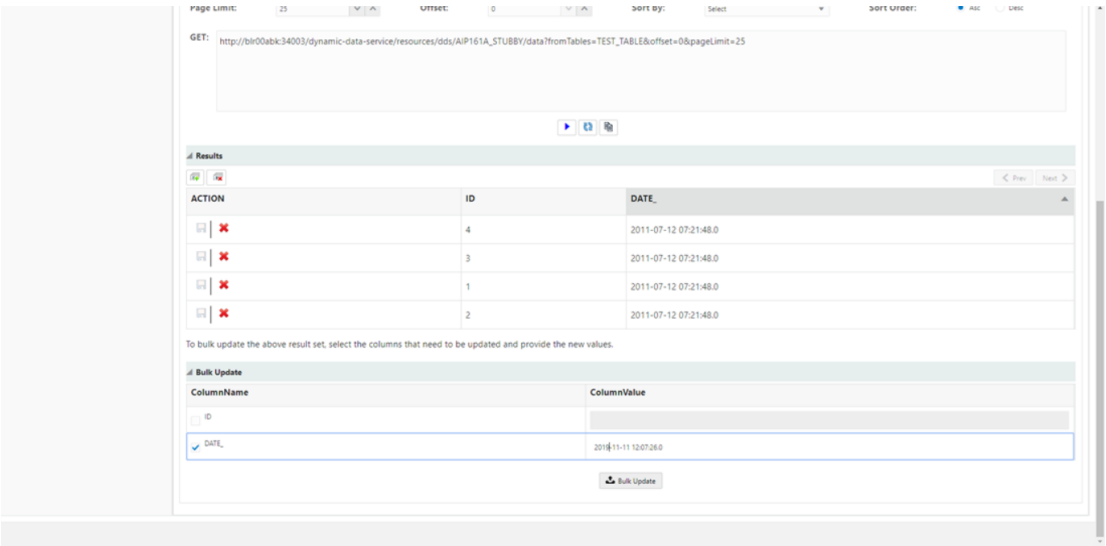
Data Deletion

This option is available to perform individual and bulk deletions of records.

Individual Delete - To delete a single record, click the delete button on the corresponding record.

Bulk Delete -To delete all records, click the delete all button and confirm once the confirmation dialog pops up.

The number of records deleted message will be displayed upon successful deletion of records.



5

Advanced Backend Features

The DDS backend service supports many SQL query commands. Some of these commands can be added to a query using the components available in the DDS UI, while others must be added by manually typing them into the query itself. Once that is done, the query can then be run through the DDS UI, or a browser, or a cURL commands to get the results.

Follow the instructions in the "Query Builder" section within the "Designer Tab Guide" section of the "User Interface" chapter for instructions on setting up a basic query in the DDS UI. Once those steps have been followed, a query is generated in the **GET** box. The query can be edited to add the following statements individually or in a combination as required.

> **Note:**
>
> Please maintain the formatting of the GET request, making sure all the special characters (like the separating ampersand `&`) are in place, a lack of which may cause unexpected errors.

> **Note:**
>
> However, before any of the following commands can be run we need an access_token, which is mandatory when using applications using OAuth2.0 authentication. The process to create an OAuth2.0 application and getting the access token is discussed in the chapter Security Setup.

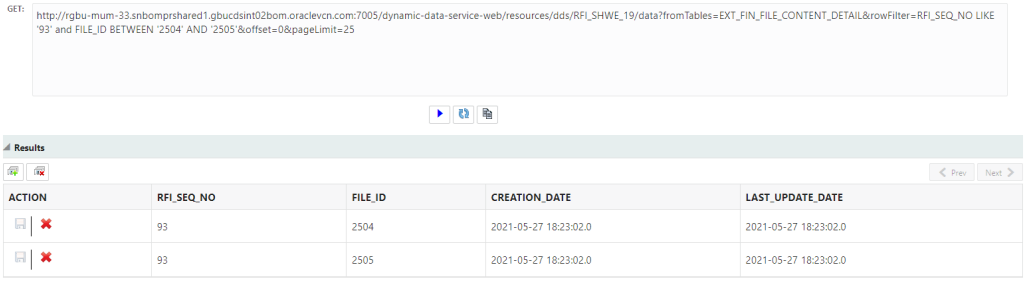
LIKE Statement

The LIKE statement can be added separately to the query request to view only the results that are similar to the LIKE value. A simple GET request and result with LIKE statement can be like the following:

```
http://<dds_host>:<port>/dynamic-data-service-web/resources/dds/<selected_schema>/data?
fromTables=<selected_table>&rowFilter=<COULMN_NAME> LIKE
'<data_to_search>'&offset=0&pageLimit=25
```

A `curl` equivalent of the above example:

```
Curl -i -H "Authorization: Bearer $AccessToken" -H "Content-Type: application/
json;charset=UTF-8" --request GET  "http://<dds_host>:<port>/dynamic-data-service-web/
resources/dds/<selected_schema>/data?
fromTables=<selected_table>&rowFilter=<COULMN_NAME>%20LIKE%20'<data_to_search>'&offset=
0&pageLimit=25"
```



IN Statement

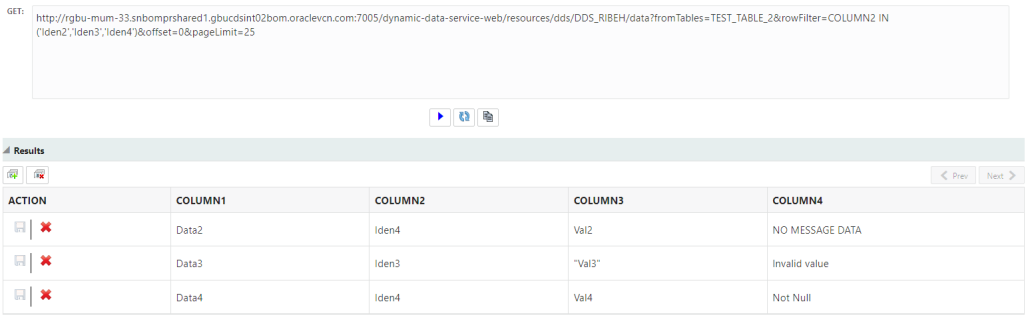
The IN statement can be added separately to the query request to only view results containing one of a set of possible values. A simple GET request and result with IN statement can be like the following:

```
http://<dds_host>:<port>/dynamic-data-service-web/resources/dds/
<selected_schema>/data?fromTables=<selected_table>&rowFilter=<COULMN_NAME> LIKE
'<data_to_search>'&offset=0&pageLimit=25
```

A `curl` equivalent of the above example:

```
Curl -i -H "Authorization: Bearer $AccessToken" -H "Content-Type: application/
json;charset=UTF-8" --request GET  "http://<dds_host>:<port>/dynamic-data-
service-web/resources/dds/<selected_schema>/data?
fromTables=<selected_table>&rowFilter=<COULMN_NAME>%20IN%20('<value1>','<value2>'
,'<value3>', …)&offset=0&pageLimit=25" -o filename.txt
```

> **Note:**
>
> The IN Statement supports searching using comma-separated values in parentheses. Sub-queries are **NOT** supported by the DDS design.



Comparative Operators

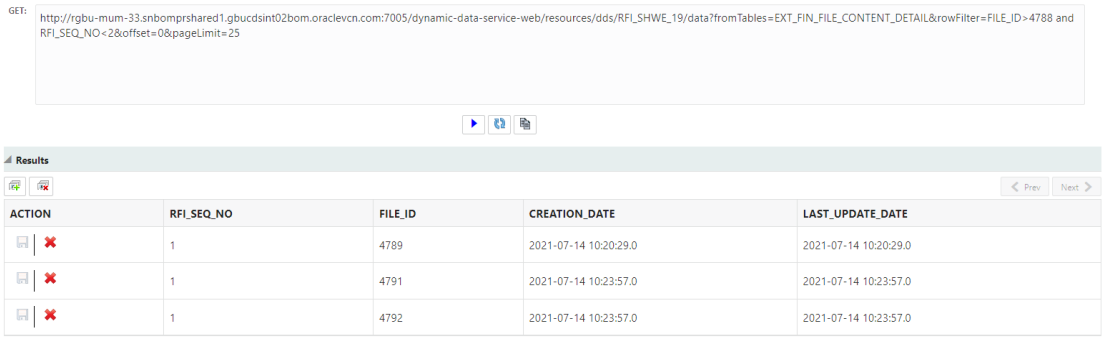
Comparative operators (like < and >) can be used to filter data for a particular range. A GET request and result with these operators can be like the following:

```
http://<dds_host>:<port>/dynamic-data-service-web/resources/dds/
<selected_schema>/data?
```

```
fromTables=<selected_table>&rowFilter=COULMN_NAME<'data_to_search'&offset=0&pageLimit=2
5
```

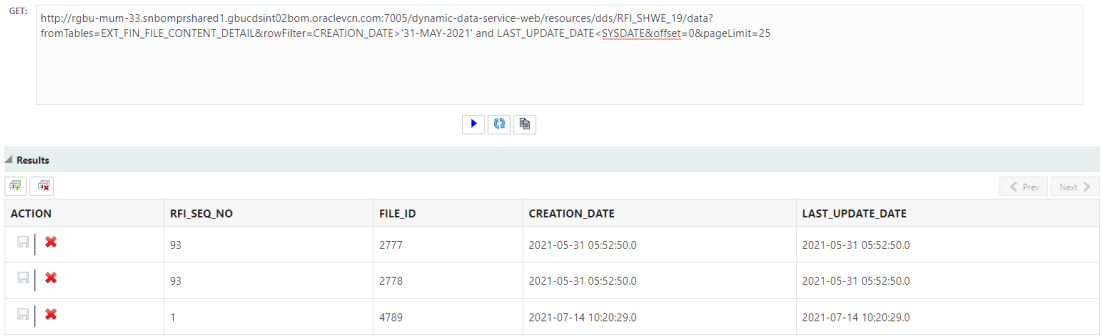
A `curl` equivalent of the above example:

```
Curl -i -H "Authorization: Bearer $AccessToken" -H "Content-Type: application/
json;charset=UTF-8" --request GET  "http://<dds_host>:<port>/dynamic-data-service-web/
resources/dds/<selected_schema>/data?
fromTables=<selected_table>&rowFilter=COULMN_NAME<'data_to_search'&offset=0&pageLimit=2
5" -o filename.txt
```



Filtering DATE Column

Date columns can be filtered using the `'DD-MMM-YYYY'` date format. Additionally, `SYSDATE±N` can also be used to query date columns.



Example with the **SYSDATE** keyword:

```
http://<dds_host>:<port>/dynamic-data-service-web/resources/dds/<selected_schema>/data?
fromTables=<selected_table>&rowFilter=CREATION_DATE<SYSDATE-1&offset=0&pageLimit=25
```

A `curl` equivalent of the above example:

```
Curl -i -H "Authorization: Bearer $AccessToken" -H "Content-Type: application/
json;charset=UTF-8" --request GET  "http://<dds_host>:<port>/dynamic-data-service-web/
resources/dds/<selected_schema>/data?
fromTables=<selected_table>&rowFilter=CREATION_DATE<SYSDATE-1&offset=0&pageLimit=25" -
o file-name.txt
```

Example with **'DD-MMM-YYYY'** date format:

```
http://<dds_host>:<port>/dynamic-data-service-web/resources/dds/
<selected_schema>/data?fromTables=<selected_table>&rowFilter=CREATION_DATE>'01-
JAN-1976'&offset=0&pageLimit=25
```

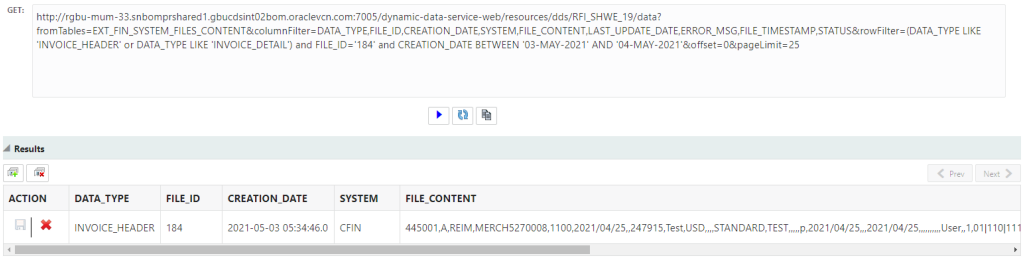
A `curl` equivalent of the above example:

```
Curl -i -H "Authorization: Bearer $AccessToken" -H "Content-Type: application/
json;charset=UTF-8" --request GET  "http://<dds_host>:<port>/dynamic-data-
service-web/resources/dds/<selected_schema>/data?
fromTables=<selected_table>&rowFilter=CREATION_DATE>'01-
JAN-1976'&offset=0&pageLimit=25" -o file-name.txt
```

Combinations of OR, AND & BETWEEN Statements

The conditional statements OR, AND & BETWEEN can be used in combination with each other using parentheses to further filter results based on required conditions. The **Filter DDS** row in the UI can be used to add a single conditional statement, upon which more conditions can be added manually by typing them in the GET box. A simple GET request with result can look like this:

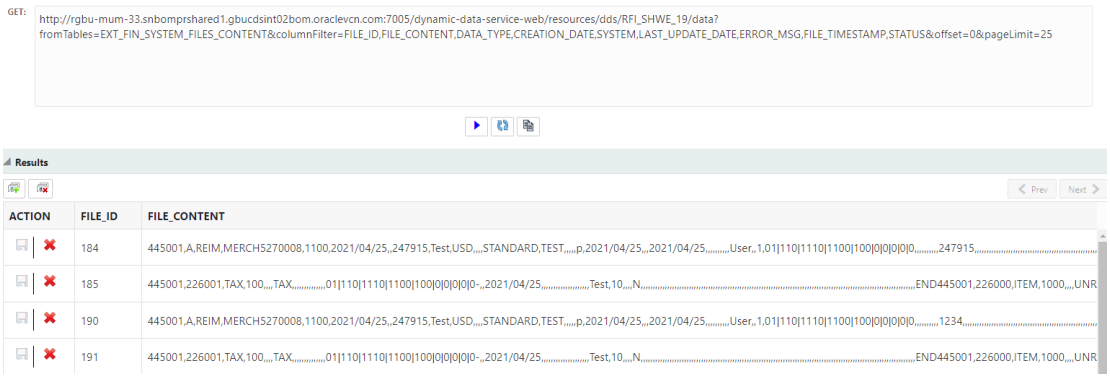
```
http://<dds_host>:<port>/dynamic-data-service-web/resources/dds/
<selected_schema>/data?fromTables=<selected_table>&rowFilter=(DATA_TYPE LIKE
'INVOICE_HEADER' or DATA_TYPE LIKE 'INVOICE_DETAIL') and FILE_ID='184' and
CREATION_DATE BETWEEN '03-MAY-2021' AND '04-MAY-2021'&offset=0&pageLimit=25
```



A `curl` equivalent of the above example:

```
Curl -i -H "Authorization: Bearer $AccessToken" -H "Content-Type: application/
json;charset=UTF-8" --request GET  "http://<dds_host>:<port>/dynamic-data-
service-web/resources/dds/<selected_schema>/data?
fromTables=<selected_table>&rowFilter=(DATA_TYPE%20LIKE%20'INVOICE_HEADER'%20or%2
0DATA_TYPE%20LIKE%20'INVOICE_DETAIL')
%20and%20FILE_ID='184'%20and%20CREATION_DATE%20BETWEEN%20'03-
MAY-2021'%20AND%20'04-MAY-2021'&offset=0&pageLimit=25" -o filename.txt
```

Viewing Clob Data


GET:
http://rgbu-mum-33.snbomprshared1.gbucdsint02bom.oraclevcn.com:7005/dynamic-data-service-web/resources/dds/RFI_SHWE_19/data?
fromTables=EXT_FIN_SYSTEM_FILES_CONTENT&columnFilter=FILE_ID,FILE_CONTENT,DATA_TYPE,CREATION_DATE,SYSTEM,LAST_UPDATE_DATE,ERROR_MSG,FILE_TIMESTAMP,STATUS&offset=0&pageLimit=25
Results
Prev
Next
ACTION
FILE_ID
FILE_CONTENT
184
445001,A,REIM,MERCH5270008,1100,2021/04/25,,247915,Test,USD,,,,STANDARD,TEST,,,,,p,2021/04/25,,,2021/04/25,,,,,,,,,User,,1,01|110|1110|1100|100|0|0|0|0|0,,,,,,,,247915,
185
445001,226001,TAX,100,,,,TAX,,,,,,,,,,,,01|110|1110|1100|100|0|0|0|0|0-,,2021/04/25,,,,,,,,,,,,,,,,,,,,Test,10,,,,N,,,,,,,,,,,,,,,,,,,

```
web/resources/dds/<selected_schema>/data?
fromTables=<selected_table>&offset=0&pageLimit=25" -H "Accept:application/json" -
o file_name.txt
```

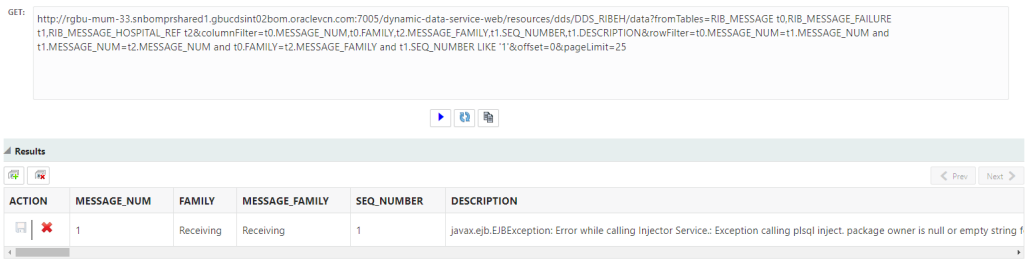
> **Note:**
>
> Ensure that the required DDS application username/password is supplied.

JOINing Two or More Tables

DDS has the ability to join two or more tables. This is done using the JOIN element in the Query Builder Tab under the Designer section.

Use the check box to enable table joins. Once enabled, click inside the visible text field to view a list of tables. Select one or more tables as required.

> **Note:**
>
> The row filter must be added to this, which can either be done by using the Row Filter in the UI or adding your own `rowFilter` by typing the required conditions in the GET query box. You must also ensure that all required columns are selected in the column filter, otherwise an error will display when you try to run the query.



A sample GET query with result is as follows:

```
http://<dds_host>:<dds_port>/dynamic-data-service-web/resources/dds/
<SELECTED_TABLE>/data?fromTables=TABLE1 t0,TABLE2 t1,TABLE3
t2&columnFilter=t0.COLUMN1,t1.COLUMN1,t2.COLUMN1,t0.COLUMN2,t1.COLUMN2,t2.COLUMN2
,t0.COLUMN3,t1.COLUMN3,t2.COLUMN3&rowFilter=t0.COLUMN1=t1.COLUMN1 and
```

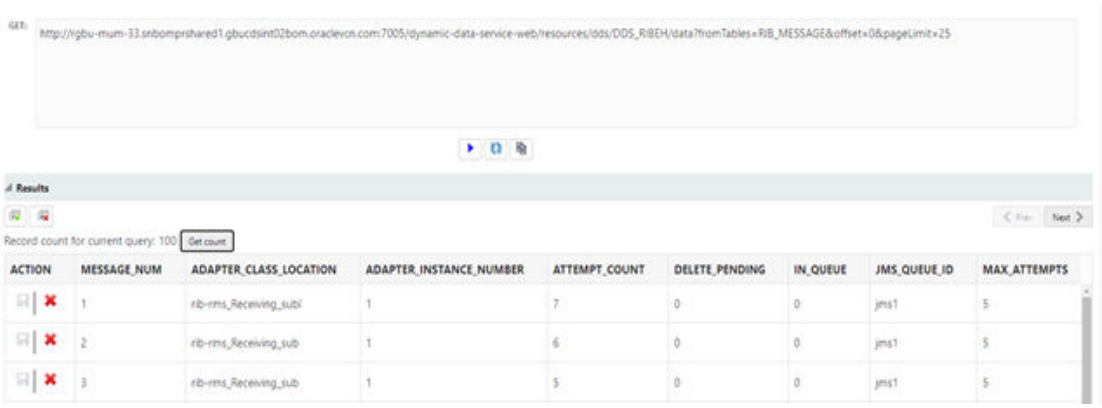
t1.COLUMN1=t2.COLUMN1 and t1.COLUMN2=t2.COLUMN3 and t1.COLUMN3 LIKE '1'&offset=0&pageLimit=25

A `curl` equivalent of the above example:

```
Curl -i -H "Authorization: Bearer $AccessToken" -H "Content-Type: application/
json;charset=UTF-8" --request GET  "http://<dds_host>:<dds_port>/dynamic-data-service-
web/resources/dds/<SELECTED_TABLE>/data?
fromTables=TABLE1%20t0,TABLE2%20t1,TABLE3%20t2&columnFilter=t0.COLUMN1,t1.COLUMN1,t2.CO
LUMN1,t0.COLUMN2,t1.COLUMN2,t2.COLUMN2,t0.COLUMN3,t1.COLUMN3,t2.COLUMN3&rowFilter=t0.CO
LUMN1=t1.COLUMN1%20and%20t1.COLUMN1=t2.COLUMN1%20and%20t1.COLUMN2=t2.COLUMN3%20and%20t1
.COLUMN3%20LIKE%20'1'&offset=0&pageLimit=25" -o filename.txt
```

Count of Records

DDS has the ability to fetch the count of records for a particular query. This can be used to fetch count of all records or a limited set of records based on conditions and parameters as required.



A sample GET query with result is as follows:

```
http://<dds_host>:<dds_port>/dynamic-data-service-web/resources/dds/<SELECTED_SCHEMA>/
data?fromTables=TABLE1&offset=0&pageLimit=25
```

> **Note:**
>
> The count query that is actually being run is slightly different to the one displayed in GET box. The correct one is given below in the curl command equivalent.

A curl equivalent of the above example:

```
Curl -i -H "Authorization: Bearer $AccessToken" -H "Content-Type: application/
json;charset=UTF-8" --request GET "http://<dds_host>:<dds_port>/dynamic-data-service-
web/resources/dds/<SELECTED_SCHEMA>/data/count?
fromTables=TABLE1&rowFilter=<CONDITIONS_AS_REQUIRED>" -o filename.txt
```