Oracle® Cloud

Fetch Data in Chunks from Oracle ATP

G18407-02
November 2024

Oracle Cloud Fetch Data in Chunks from Oracle ATP,

G18407-02

Primary Author: Oracle Corporation

Contents

Preface

This document describes how to install, configure, and run this recipe in Oracle Integration 3.

Topics:

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

For more information, see these Oracle resources:

- Oracle Integration documentation on the Oracle Help Center.
- Oracle Cloud at http://cloud.oracle.com.

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.

Convention	Meaning
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

About This Recipe

Use this recipe to fetch data in small chunks from Oracle ATP.

> **Note:**
>
> Oracle provides this recipe as a sample only. The recipe is meant only for guidance, and is not warranted to be error-free. No support is provided for this recipe.

Overview

This recipe demonstrates how to use *stored procedures* to fetch supplier data in small chunks from Oracle ATP. You can use the recipe to implement use cases where due to service limits data cannot be fetched in bulk, but have to be fetched in smaller chunks.

To use the recipe, you must install the recipe and configure the connections and other resources within it.

The recipe uses a database package **XX_SUPPLIERS_PAGINATION_PKG**. The package uses a PL/SQL procedure `GET_SUPPLIERS_PAGINATION_PROC` which accepts the following input parameters:

- `i_limit`: A numeric parameter that specifies how many records are to be fetched from the ATP database at a time.
- `i_page`: A numeric parameter that specifies which page's data have to be fetched.

System and Access Requirements

- Oracle Integration 3
- Oracle ATP
- An account on Oracle ATP with the Administrator role

2
Before You Install the Recipe

Log in to your Oracle ATP database instance as an **Administrator** and perform the following tasks.

1. Perform general configuration tasks. See Prerequisites for Creating a Connection.
2. Create necessary objects.

   To create necessary objects, run the following script:

```
CREATE TABLE "ADMIN"."SUPPLIERS"
   (    "ID" NUMBER,
    "SUPPLIER" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
    "SUPPLIER_NUMBER" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
    "TAX_ORGANIZATION_TYPE_CODE" VARCHAR2(200 BYTE) COLLATE
"USING_NLS_COMP",
    "TAX_ORGANIZATION_TYPE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
    "SUPPLIER_TYPE_CODE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
    "SUPPLIER_STATUS" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
    "BUSINESS_RELATIONSHIP_CODE" VARCHAR2(200 BYTE) COLLATE
"USING_NLS_COMP",
    "BUSINESS_RELATIONSHIP" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
    "TAX_REGISTRATION_COUNTRY" VARCHAR2(200 BYTE) COLLATE
"USING_NLS_COMP",
    "TAXPAYER_COUNTRY_CODE" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
    "TAXPAYER_COUNTRY" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
    "TAXPAYER_ID" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP",
    "CREATION_DATE" TIMESTAMP (6) DEFAULT ON NULL SYSDATE,
    "UPDATION_DATE" TIMESTAMP (6) DEFAULT ON NULL SYSDATE
   )  DEFAULT COLLATION "USING_NLS_COMP" ;

CREATE UNIQUE INDEX "ADMIN"."SUPPLIERS_PK" ON "ADMIN"."SUPPLIERS" ("ID")
  ;

ALTER TABLE "ADMIN"."SUPPLIERS" ADD CONSTRAINT "SUPPLIERS_PK" PRIMARY KEY
("ID")
  USING INDEX  ENABLE;

ALTER TABLE "ADMIN"."SUPPLIERS" MODIFY ("CREATION_DATE" NOT NULL ENABLE);

ALTER TABLE "ADMIN"."SUPPLIERS" MODIFY ("UPDATION_DATE" NOT NULL ENABLE);

CREATE OR REPLACE EDITIONABLE TRIGGER "ADMIN"."SUPPLIERS_TRIGGER"
BEFORE INSERT ON SUPPLIERS
FOR EACH ROW
    WHEN (new.ID IS NULL) BEGIN
  :new.ID := SUPPLIERS_SEQ.NEXTVAL;
END;
/
ALTER TRIGGER "ADMIN"."SUPPLIERS_TRIGGER" ENABLE;
```

```
CREATE OR REPLACE EDITIONABLE TRIGGER "ADMIN"."SUPPLIERS_UPDATE_DATE"
BEFORE UPDATE ON SUPPLIERS
FOR EACH ROW
BEGIN
 :NEW.UPDATION_DATE := SYSDATE;
END;

CREATE OR REPLACE EDITIONABLE PACKAGE
"ADMIN"."XX_SUPPLIERS_PAGINATION_PKG" AS
    PROCEDURE GET_SUPPLIERS_PAGINATION_PROC (
        I_PAGE      IN NUMBER,
        I_LIMIT     IN NUMBER,
        I_PAGE_SIZE OUT NUMBER,
        P_DATA      OUT SYS_REFCURSOR
    );

END XX_SUPPLIERS_PAGINATION_PKG;
/

CREATE OR REPLACE EDITIONABLE PACKAGE BODY
"ADMIN"."XX_SUPPLIERS_PAGINATION_PKG" AS

    PROCEDURE GET_SUPPLIERS_PAGINATION_PROC (
        I_PAGE      IN NUMBER,
        I_LIMIT     IN NUMBER,
        I_PAGE_SIZE OUT NUMBER,
        P_DATA      OUT SYS_REFCURSOR
    ) AS
        COUNT1 NUMBER;
    BEGIN
        SELECT
            COUNT(*)
        INTO COUNT1
        FROM
            SUPPLIERS;

        I_PAGE_SIZE := CEIL(COUNT1 / I_LIMIT);
        OPEN P_DATA FOR SELECT
                                            *
                                        FROM
                                            SUPPLIERS
                        ORDER BY
                            ID
                        OFFSET NVL((I_PAGE - 1), 0) * I_LIMIT ROWS FETCH
NEXT I_LIMIT ROWS ONLY;

    END;

END XX_SUPPLIERS_PAGINATION_PKG;
```

3

Install and Configure the Recipe

On your Oracle Integration instance, install the recipe to deploy and configure the integration and associated resources.

1. On the Oracle Integration Home page, in the **Get started** section, click **Browse store**.
2. Find the recipe you want to install, then click **Get**.

   A message confirms that the recipe was successfully installed, and the recipe card shows **In use**.

3. Click **Configure**  on the recipe to configure its resources.

   The project workspace opens, displaying all the resources of the recipe.

Configure the Oracle ATP Connection

1. In the Connections section, click the connection name.
2. In the Properties section, specify the database service name in the **Service Name** field. See Before You Install the Recipe.
3. In the Security section, enter the following details:

Field	Information to Enter
Security Policy	Select **JDBC Over SSL**.
Wallet	Select the check box, then click **Upload** to upload the wallet file. See Before You Install the Recipe.
Wallet Password	Enter the wallet password.
Database Service Username	Enter the database service username to connect to the ATP database. The database service username is the schema username for the user to log in to the database. The database service username is not the same as the database service name that you specify in the connection Properties section.
Database Service Password	Enter the database service password to connect to the ATP database.

4. Click **Save**. If prompted, click **Save** again.
5. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again.

   A message confirms if your test is successful.

6. To return to the project workspace, click **Go back**  .

4

Activate and Run the Recipe

After you've configured the connections and other resources, you can activate and run the recipe.

1. In the project workspace, click **Activate**. In the Activate project panel, with the default project deployment selected, choose an appropriate tracing option, then click **Activate**.

   A message confirms that the integration has been activated. Refresh the page to view the updated status of the integration.

2. Run the recipe.

   a. In the Integrations section of the project workspace, click **Actions** ⋮ on the **Extract Data from ATP in Chunk** integration flow, then select **Run**.

   b. On the Configure and run page, click **Run**.

      You've now successfully submitted the integration for execution.

3. Monitor the running of the integration flow in Oracle Integration.

   a. In the project workspace, click **Observe**. You'll see the integration flow being triggered and running successfully.

   b. To manage errors in your project, see Manage Errors in a Project.

4. Verify that data is fetched from Oracle ATP in chunks according to the parameters defined in the stored procedure.

Related Documentation

- *Using the Oracle Autonomous Transaction Processing Adapter with Oracle Integration 3*