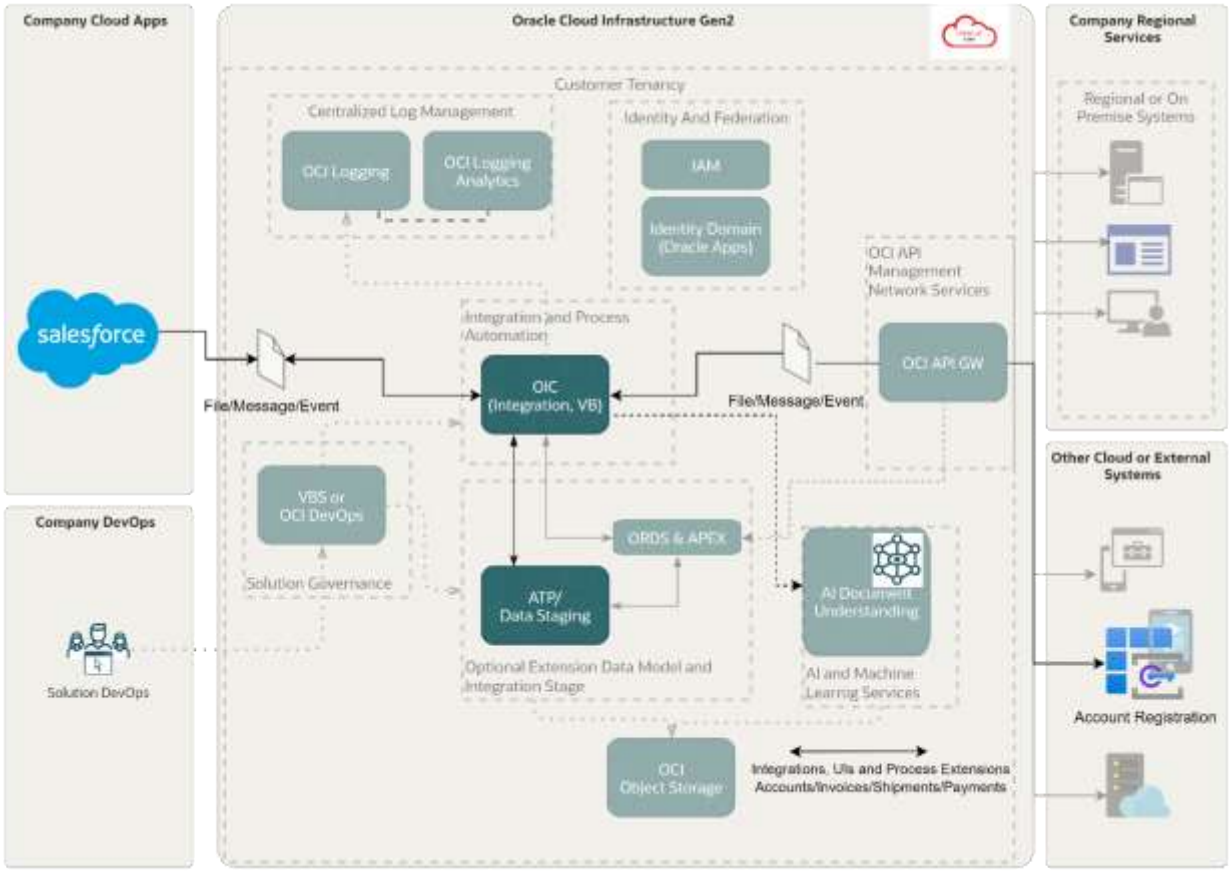
Using the Oracle Autonomous Transaction Processing Adapter to store and retrieve binary objects.

Contents

Purpose

This simple template project reflects some real use cases. E.g. customer/partner uses mobile/web application to register account based on photo of the Identity document. In our case it will be implementation of the REST services that stores the attached document to ATP and retrieval of the stored content based on some business key.



Overview

This simple template project demonstrates how you can send multipart mixed content payload through Oracle Integration to an Autonomous Database and then later retrieve the attachment from DB. All this can be invoked/consumed from an external application through REST API.

This is an application-driven recipe that sends a multipart message when it receives a REST request with JSON payload and file attachment. Attachment is stored in the table in the ATP and service replies to the processing result with ID that can be used to retrieve the attachment REST request to read the attachment from the ATP and return the content.

To use the project, you must import the car file and configure the connections within the project. Subsequently, you can activate the integration flows and send a REST request with the required payload to the integration's endpoint URL from an external application.

When triggered, the project's integration flow sends a hello message to the email address specified in the request payload and returns a response to the external application.

System and Access Requirements

- Oracle Integration, Version 23.2.0.0.0, or higher

Solution prerequisites

ATP Database 19c or 23i and the Table Created:

```
CREATE TABLE "DEMO"."ACCOUNTDOCUMENT"

( "ID" NUMBER(*,0) GENERATED ALWAYS AS IDENTITY MINVALUE 1 MAXVALUE 9999999999999999999999999999 INCREMENT BY 1 START
WITH 1 CACHE 20 NOORDER  NOCYCLE  NOKEEP  NOSCALE ,
	"ACCOUNTID" VARCHAR2(30 BYTE) COLLATE "USING_NLS_COMP",
	"DOCUMENTTYPE" VARCHAR2(100 BYTE) COLLATE "USING_NLS_COMP",
	"CONFIDENCE" NUMBER(*,0),
	"DOCUMENTNUMBER" VARCHAR2(20 BYTE) COLLATE "USING_NLS_COMP",
	"FIRSTNAME" VARCHAR2(20 BYTE) COLLATE "USING_NLS_COMP",
	"LASTNAME" VARCHAR2(20 BYTE) COLLATE "USING_NLS_COMP",
	"COUNTRY" VARCHAR2(20 BYTE) COLLATE "USING_NLS_COMP",
	"NATIONALITY" VARCHAR2(20 BYTE) COLLATE "USING_NLS_COMP",
	"BIRTHDATE" VARCHAR2(20 BYTE) COLLATE "USING_NLS_COMP",
	"EXPIRYDATE" VARCHAR2(20 BYTE) COLLATE "USING_NLS_COMP",
	"GENDER" VARCHAR2(20 BYTE) COLLATE "USING_NLS_COMP",
	"DOCUMENTBINARY" BLOB,
	"DOCUMENTNAME" VARCHAR2(200 BYTE) COLLATE "USING_NLS_COMP"
  ) DEFAULT COLLATION "USING_NLS_COMP";
CREATE UNIQUE INDEX "DEMO"."ACCOUNTDOCUMENT_PK" ON "DEMO"."ACCOUNTDOCUMENT" ("ID");
ALTER TABLE "DEMO"."ACCOUNTDOCUMENT" MODIFY ("ID" NOT NULL ENABLE);
ALTER TABLE "DEMO"."ACCOUNTDOCUMENT" ADD CONSTRAINT "ACCOUNTDOCUMENT_PK" PRIMARY KEY ("ID")
USING INDEX "DEMO"."ACCOUNTDOCUMENT_PK"  ENABLE;
```

Install and Configure the Project

On your Oracle Integration instance import the project archive(.car) and configure the integration and associated resources.

A message confirms that the project was successfully imported, and project can be listed in the OIC Projects.

Click Edit icon on every connection defined in the project.

- Configure the REST Trigger Connection “Sample REST Endpoint Interface”.
  - In the Connections section, click the connection name.
  - Click Test to ensure that your connection is successfully configured.
  - A message confirms if your test is successful. Click Save.
  - If prompted, click Save again.

To return to the project workspace, click Go back Back icon

Activate and Run the Recipe

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

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 integrations have been activated. Refresh the page to view the updated status of the integration.

Run the recipe from an external application.

In the Integrations section of the project workspace, click Actions on the integration flow, then select Run.

On the Configure and run page, click Endpoint metadata.

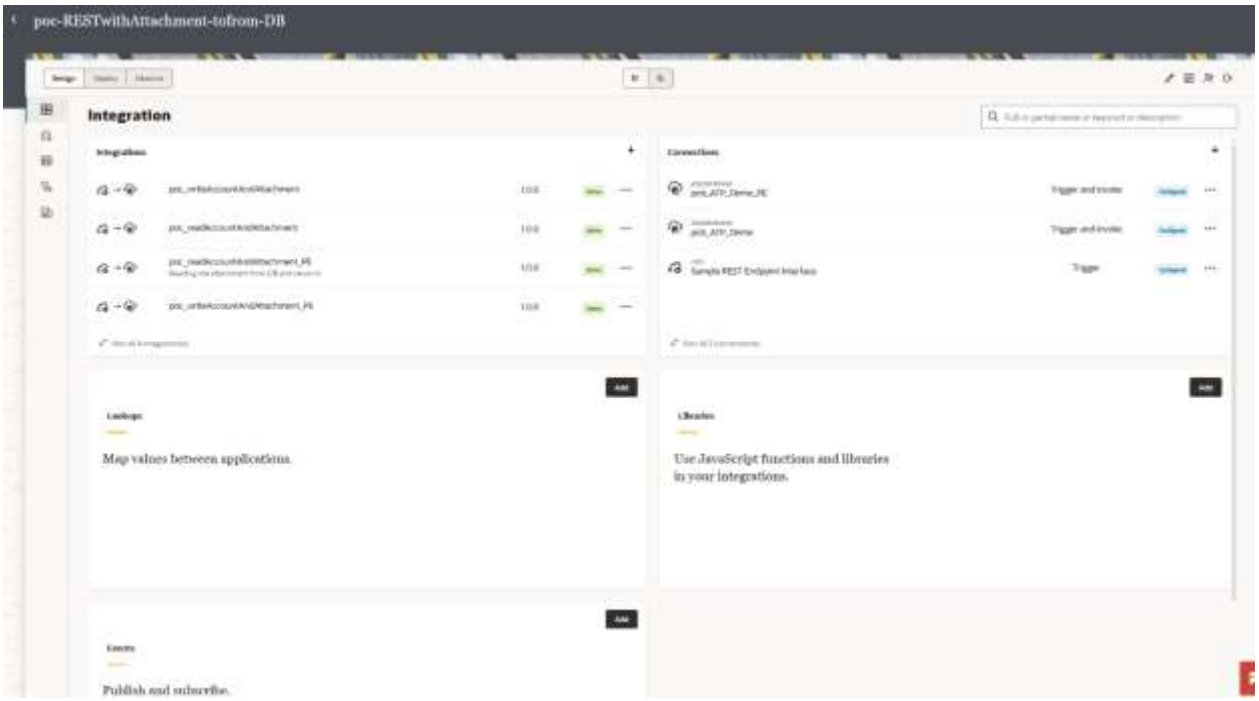
In the panel that opens, copy the Endpoint URL value. This is the integration flow's endpoint URL.

From the external application, send a POST request to this endpoint URL along with the following request parameters:

- File (chose any file from your disk – text or binary)
- payload (see example which can be used e.g. as Oracle AI Document understanding response – create account based on scanned document e.g. Passport, ID, Driver License)

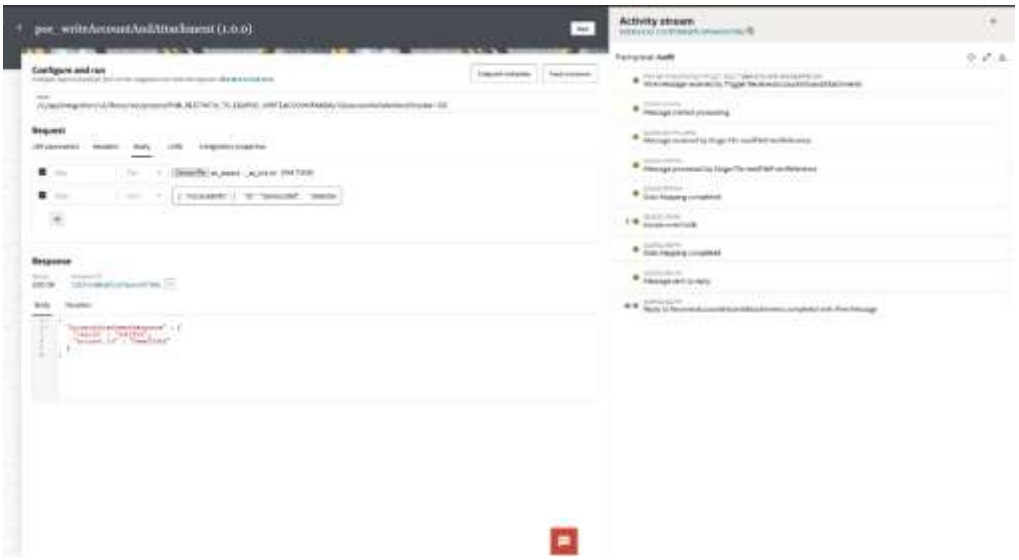
Test the project flows in the Oracle Integration.

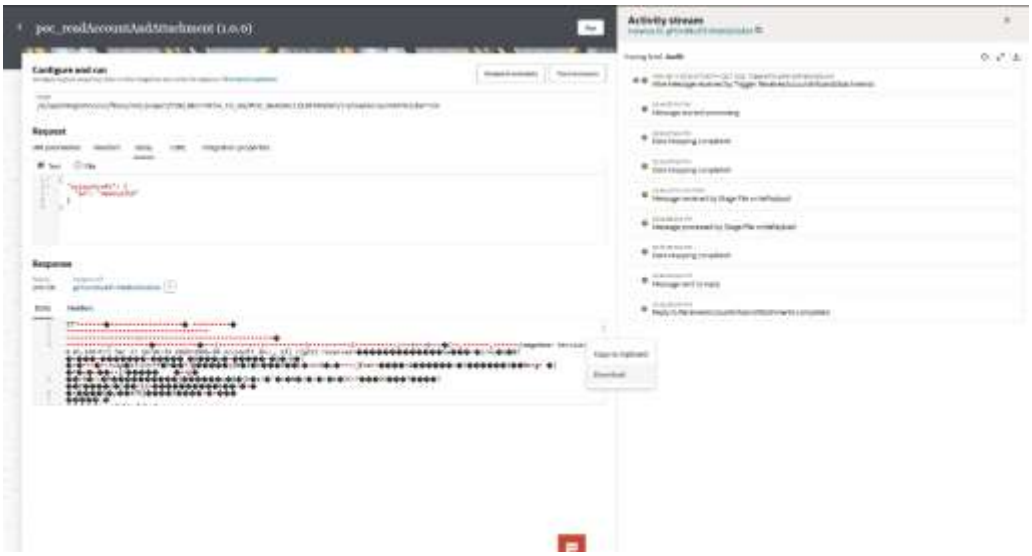
In the Integrations section of the project workspace, click Actions on the integration flow, then select Run.



On the Configure and run page, in the Request section, under the URI parameters tab, enter the name of the user and specify the email address to which the message must be sent.

Click Run

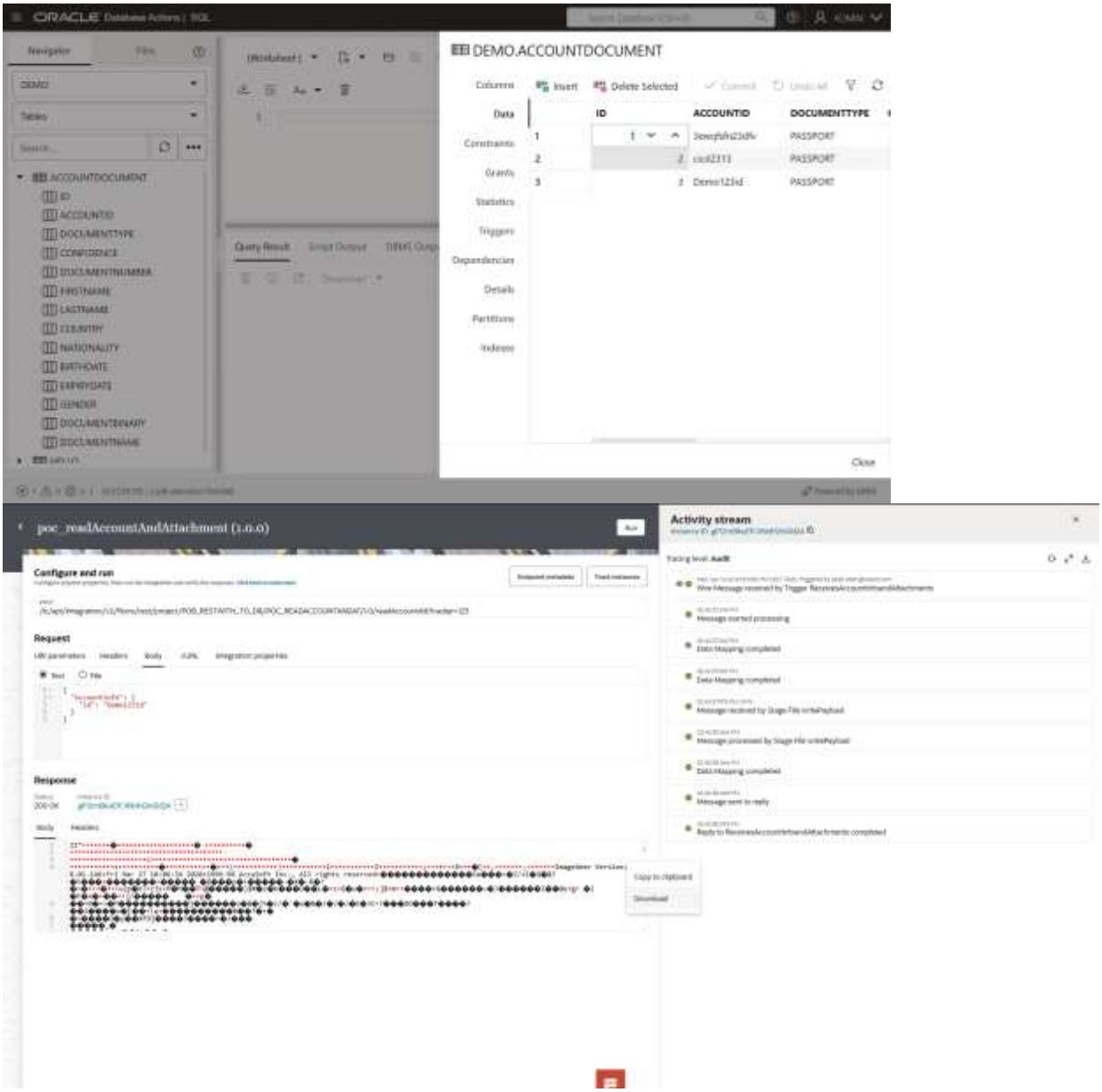




The Activity stream pane appears displaying the status of the integration instance's execution.

In the Response section of the Configure and run page, you'll find a success status, 200 OK, and the response returned.

Check if a row was populated in the table and corresponds with the request payload.



Related Documentation

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