

DUMPS ARENA

Oracle Cloud Infrastructure 2026 Application Integration Professional

Oracle 1z0-1042-25

Version Demo

Total Demo Questions: 8

Total Premium Questions: 86

Buy Premium PDF

<https://dumpsarena.co>

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

QUESTION NO: 1

When defining the frequency for scheduled orchestration style integration runs, you can choose either Basic or the Advanced option. Which expression format is used for defining advanced scheduled runs?

- A. iCal
- B. cron
- C. YAML
- D. Groovy

ANSWER: A

Explanation:

iCal is the expression format used when defining an Advanced schedule for a scheduled orchestration integration in Oracle Integration. The Basic scheduling interface is intended for common recurrence patterns and provides guided fields for selecting timing details. Advanced scheduling supports more precise recurrence definitions through an iCalendar-based recurrence expression, commonly called an iCal expression. This format represents recurrence using structured properties such as frequency, interval, days of the week, days of the month, and an end condition. For example, an advanced recurrence can define a daily, weekly, or monthly cadence and further constrain the specific dates or weekdays on which the integration is eligible to run. Oracle Integration uses this format so that schedules can express patterns beyond the combinations exposed in the Basic interface while retaining a standardized recurrence model. Oracle's scheduling documentation identifies the Advanced option as using iCal expressions and provides guidance for creating and validating the schedule. The underlying recurrence conventions align with the iCalendar standard's recurrence-rule model, which defines properties such as `FREQ`, `INTERVAL`, and `BYDAY`. See Oracle's [Schedule Integrations documentation](#) and [RFC 5545: Internet Calendaring and Scheduling Core Object Specification](#).

QUESTION NO: 2

You are exposing an order-management API through an Oracle Integration REST trigger. Which three design statements are consistent with RESTful service design? Select THREE.

- A. Resource-oriented URIs should identify business resources such as orders or customers.
- B. HTTP methods such as GET, POST, PUT, PATCH, and DELETE should express the requested operation.
- C. HTTP status codes should communicate the outcome of a request.
- D. Every request should be wrapped in a SOAP envelope before it is processed.
- E. The server should maintain a separate conversational session for every API client.

ANSWER: A B C

Explanation:

RESTful APIs identify business resources with URIs, use standard HTTP methods to express operations, and use HTTP status codes to communicate the result of a request. A REST request should contain the information needed to process that request; the server should not depend on a conversational client session to determine the request state.

SOAP envelopes are part of SOAP messaging, not a REST requirement. Maintaining server-side client session state conflicts with the REST statelessness constraint.

QUESTION NO: 3

Which two statements are true regarding messaging adapters in Oracle Integration Cloud (OIC)?

- A. Both the Apache Kafka and OCI Streaming adapters support XML, JSON, and AVRO message formats.

- B. The OCI Streaming adapter does not support consuming messages from a specified consumer group.
- C. The Apache Kafka Adapter can be configured to support Transactions in case of Producers.
- D. It is not mandatory to install a connectivity agent to use the OCI Streaming Adapter for Inbound polling.

ANSWER: A C

Explanation:

Both the Apache Kafka and OCI Streaming adapters support XML, JSON, and AVRO message formats. This enables integrations to exchange structured business payloads with messaging systems while preserving the format expected by publishers and subscribers. In Oracle Integration, the adapters provide configuration for the message's serialization and deserialization requirements, including AVRO schema-related handling where applicable. This lets an integration consume or publish messages to Kafka topics or OCI Streaming streams using formats commonly used in event-driven architectures.

The Apache Kafka Adapter can be configured to support Transactions in case of Producers. Producer transactions provide atomic publication behavior: records sent as part of a transaction are committed together, allowing downstream consumers that are configured for committed reads to avoid seeing incomplete or aborted transactional output. This capability is especially useful when an integration must publish a consistent set of related events and requires stronger delivery guarantees than independent, nontransactional sends. Oracle documents the available Kafka Adapter producer configuration, including transaction support, and the supported messaging payload formats in its adapter guidance. See the [Oracle Integration Kafka Adapter documentation](#) and the [Oracle Integration OCI Streaming Adapter documentation](#).

QUESTION NO: 4

You must make customer order numbers and source-system request IDs available to operations staff when they search Oracle Integration instances. Which two statements are true regarding business identifiers and tracking fields? Select TWO.

- A. Tracking fields can expose business values that users can use to search for integration instances.
- B. Tracking fields automatically transform the source payload before it reaches an invoke connection.
- C. Business identifiers should be selected from values available in the integration's message data.
- D. Tracking fields eliminate the need to configure fault handling for failed instances.
- E. Business identifiers can be used only for scheduled integrations.

ANSWER: A C

Explanation:

Tracking fields are configured from data available in the integration and are used as business identifiers for monitoring and searching instances. Selecting meaningful business values, such as an order number or source request ID, enables operations staff to locate the relevant instance without first knowing its Oracle Integration instance identifier.

Tracking fields support observability; they do not transform the request payload or replace an integration's fault-handling logic. They should therefore be chosen as stable business values useful for correlation and investigation.

QUESTION NO: 5

You are an Oracle Integration Cloud (OIC) specialist working on a project to integrate two systems using OIC. During testing, you notice that the integration instance fails to execute and returns an error message. Which step would you take first to troubleshoot the issue?

- A. Check the instance logs for specific error messages and details.
- B. Increase the number of instances to handle the error.
- C. Immediately restart the integration instance.

D. Contact Oracle Support and wait for a response.

ANSWER: A

Explanation:

Check the instance logs for specific error messages and details. Oracle Integration provides monitoring pages where developers can locate the failed instance and inspect its error information, activity stream, payload details where available, and execution audit trail. This diagnostic information identifies the integration action at which processing stopped and typically supplies the immediate technical context needed to resolve the failure, such as a connection problem, invalid mapping data, authentication issue, fault returned by an invoked endpoint, or configuration error.

Beginning with the failed instance's monitoring details is the appropriate first troubleshooting action because it is evidence-based and preserves the context of the actual execution. From the Integrations page, users can access monitoring, locate instances by status and time range, and open an individual instance to investigate its flow and fault details. Oracle's guidance for diagnosing integration failures similarly centers on monitoring integration instances and reviewing their activity stream or errors before applying a corrective action. Once the reported fault and the affected step are understood, the specialist can make a targeted change, retest the integration, and confirm through a new instance that the issue is resolved.

See Oracle's [Monitor Integrations](#) documentation and its [Troubleshoot Integrations](#) guidance.

QUESTION NO: 6

Your design of an Asynchronous Oracle Integration Cloud (OIC) integration flow includes the requirement for additional processing logic that is conditionally based on the results returned to the integration instance from earlier external service invocations. You decide to leverage the switch action to facilitate this implementation. Which is a valid consideration concerning the configuration and capabilities of the switch action?

- A. Nested Switch actions are not supported and, therefore, you may need to create more complex conditional expressions, if necessary.
- B. Data objects returned from actions within a branch of a switch action are not visible in the mainflow of the integration after the switch action.
- C. The switch action must include an otherwise branch.

ANSWER: B

Explanation:

Data objects returned from actions within a branch of a switch action are not visible in the mainflow of the integration after the switch action. This reflects the branch-specific scope used by Oracle Integration's Switch action. A Switch action evaluates configured XPath conditions and executes only the actions in the selected branch. Because an action inside a branch may not execute at all—for example, when another condition is selected—its returned data cannot be treated as data that is reliably available to subsequent actions in the main integration flow. This behavior prevents downstream mappings from depending on a response that is conditional or absent at runtime.

When processing performed after the Switch action requires a value produced in a branch, the integration should explicitly place the needed value into a variable that is available at the appropriate integration scope. Each applicable branch can then assign a compatible value to that variable, allowing later actions to map from a known, consistently scoped source. This design also makes the conditional data flow clear and ensures the integration can be validated and mapped predictably. Oracle documents the Switch action's conditional branching behavior and its use for organizing actions under distinct conditions in the [Oracle Integration documentation for Switch actions](#). For variable handling and mapper availability, see the [Oracle Integration documentation for variables](#).

QUESTION NO: 7

You are designing a new integration in OIC that needs to expose a RESTful interface trigger for client applications. Which is a valid consideration while choosing request and response options using the REST Adapter Endpoint Configuration wizard?

- A. Both standard and custom HTTP headers can be defined for both the request and the response.
- B. File attachments can be accepted as a request option for the trigger interface, but you cannot include attachments as a response to the client.
- C. If you select a PUT or POST action (HTTP method), you must define a JSON response payload format.
- D. Although you can define both query and template parameters, only template parameters are available for mapping in the mapper because query parameters are considered optional.

ANSWER: A

Explanation:

Both standard and custom HTTP headers can be defined for both the request and the response. This is a valid REST Adapter trigger configuration consideration because headers are often part of an interface contract, carrying protocol metadata, authentication-related information, correlation identifiers, business context, and client-specific values independently of the message body. In the REST Adapter Endpoint Configuration Wizard, request headers can be configured so that inbound clients provide the required header values to the integration. Response headers can also be configured when the integration must return header-based information to the invoking client.

Oracle Integration represents configured headers in the integration message structure, enabling the values to be accessed and mapped as appropriate. Standard HTTP headers can be selected from the supported header definitions, while custom headers allow an API designer to expose application-specific metadata. This lets a REST trigger define a clear and controlled interface without requiring every value to be embedded in a JSON or XML payload. Header definitions should therefore be chosen deliberately as part of the endpoint contract, including their names, data types, and whether they are required for the request. Oracle documents REST Adapter endpoint configuration and REST trigger request/response capabilities in the [Oracle REST Adapter documentation](#) and the [Oracle Integration documentation](#).

QUESTION NO: 8

You are configuring an Oracle Integration REST Adapter invoke connection for a partner API. The partner accepts JSON request payloads and requires OAuth 2.0 access tokens. Which two statements are valid? Select TWO.

- A. The REST Adapter can be configured to invoke an API that accepts JSON payloads.
- B. The REST Adapter can use an OAuth 2.0 security policy for the partner API.
- C. The partner API must expose a WSDL before it can be invoked through the REST Adapter.
- D. A Connectivity Agent is mandatory for every REST Adapter invoke connection.
- E. The REST Adapter can invoke only APIs that use Basic Authentication.

ANSWER: A B

Explanation:

The REST Adapter can be configured to invoke REST APIs that exchange JSON payloads. Its endpoint configuration defines the resource operation and message structures used by the integration mapper to create the outgoing request and process the response.

The REST Adapter also supports OAuth 2.0 security policies for applicable REST interactions. Configuring the appropriate OAuth policy enables Oracle Integration to authenticate to the partner API using the credentials and token flow required by the selected policy.

SOAP is not required merely because the API uses OAuth, and the Connectivity Agent is needed only when private network access is required; it is not a general prerequisite for invoking public REST APIs.