

Salesforce Certified MuleSoft Hyperautomation Developer

Salesforce Mule-Dev-202

Version Demo

Total Demo Questions: 7

Total Premium Questions: 72

Buy Premium PDF

<https://passqueen.com>

support@passqueen.com

PASSQUEEN.COM

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Designing RPA Processes	13
Topic 2, Building RPA Processes	5
Topic 3, Testing and Debugging RPA Processes	4
Topic 4, Deploying and Managing RPA Processes	11
Topic 5, Mix Questions	39
Total	72

QUESTION NO: 1

Northern Trail Outfitters (NTO) wants to automate a multi-step process that spans several departments.

How do Interactive Steps in Flow Orchestration help NTO involve users at key steps of the process?

- A. They allow the user to interact directly with external systems through the Salesforce UI.
- B. They allow the user to interact with the process in between automated backend steps.
- C. They enable users to collaborate on specific work items.
- D. They leverage AI processing to automatically interact with the customer and collect customer data.

ANSWER: B

Explanation:

They allow the user to interact with the process in between automated backend steps. Interactive Steps are the human-work portion of Salesforce Flow Orchestration. An orchestration can combine background steps, which run without a person's involvement, with interactive steps that pause the orchestration and create work items for assigned users. A user opens the work item and completes the associated screen flow, providing information, making a decision, or performing the action required at that point in the business process. When the assigned work is completed, Flow Orchestration records the outcome and can continue to the next step, including subsequent automated processing or another user-facing task. This lets an organization coordinate work across departments while keeping process sequencing, assignments, and progress centralized in Salesforce. For example, an automated step can prepare or update records, an interactive step can request a manager's review or an employee's data entry, and a later background step can use the submitted information to complete downstream actions. Salesforce describes orchestration stages and steps as a way to coordinate automated and user-driven work across a process. See [Salesforce Help: Flow Orchestration](#) and [Salesforce Trailhead: Flow Orchestration Basics](#).

QUESTION NO: 2

An AnyAirlines employee regularly performs a manual process to extract customer and flight information from multiple legacy systems. AnyAirlines recently purchased MuleSoft automation and wants to automate this process using MuleSoft RPA.

During an evaluation of the candidate RPA process, which two key qualifiers should be considered? (Choose two.)

- A. Is the process rule-based?
- B. Is the process risky?
- C. Is the process data-driven?
- D. Is the process speed-sensitive?

ANSWER: A C

Explanation:

Is the process rule-based? and **Is the process data-driven?** are key qualifications for a MuleSoft RPA candidate process. RPA bots work best when a person can perform the work by following explicit, consistent business rules: for example, opening a legacy application, locating a customer or flight record, applying defined selection criteria, and copying specified values. Clear rules enable the bot to execute the same workflow reliably and predictably.

The process should also be data-driven because MuleSoft RPA is designed to interact with applications and handle information moving among them. Extracting customer and flight information from legacy systems is a representative RPA use case when the relevant data is available in a consistent digital format and can be identified through predictable screens, fields, or reports. The bot can then retrieve, validate, and enter that information into the required target system or downstream workflow. Assessing both the decision logic and the nature of the data helps determine whether the manual activity is suitable for dependable automation. MuleSoft describes RPA as automating repetitive tasks by mimicking human interaction with systems in its [MuleSoft RPA documentation](#); see also the [MuleSoft RPA Essentials](#) learning module.

QUESTION NO: 3

The MuleSoft development team at AnyAirlines has created a working application that implements a standard pattern for invoking an API, starting an RPA process, and processing the result. Other development teams need a reusable starting point that they can import and adapt for similar implementations.

Which Anypoint Exchange asset type should the team publish?

- A. API asset
- B. Connector asset
- C. Template asset
- D. Custom asset

ANSWER: C

Explanation:

Template asset is correct because an Anypoint Exchange template provides a reusable working application pattern that other teams can use as the basis for new projects. The consuming teams can adapt the template's endpoint configuration, credentials, mappings, and business logic to their own implementation.

An API asset publishes an API contract, but it does not provide the complete application implementation. A connector is intended to provide reusable connectivity, while a custom asset is not the purpose-built option for distributing an adaptable Mule application starter.

QUESTION NO: 4

Northern Trail Outfitters wants to automate two related business requirements without custom development. First, a business user must synchronize selected account updates between Salesforce and a supported SaaS application. Second, several internal departments must complete assigned work in sequence before an account review is finalized.

Which two hyperautomation tools should be used? (Choose two.)

- A. MuleSoft Composer
- B. Flow Orchestration
- C. MuleSoft RPA
- D. API Manager

ANSWER: A B

Explanation:

MuleSoft Composer is appropriate for the no-code synchronization between Salesforce and a supported SaaS application. It provides trigger-and-action integration flows that business users can configure without building a custom Mule application.

Flow Orchestration is appropriate for the departmental review because it coordinates ordered stages, automated work, interactive user tasks, assignments, and completion criteria inside Salesforce. MuleSoft RPA is not required when the involved systems expose supported integration capabilities, and API Manager governs APIs rather than coordinating human workflow.

QUESTION NO: 5

AnyAirlines uses API Functional Monitoring for a booking API that depends on a reservation database. The team wants to detect both endpoint outages and responses that are technically successful but contain incorrect booking data.

Which two monitor validations should be configured? (Choose two.)

- A. Verify that the response returns the expected HTTP status code.
- B. Verify that a response-body value matches the expected booking status.
- C. Verify that the API gateway policy configuration has not changed.
- D. Verify that no ERROR entries exist in the Mule application logs.
- E. Verify that every client application has an active SLA tier.

ANSWER: A B

Explanation:

Validating an expected HTTP status code confirms that the endpoint responds successfully to the synthetic request. Validating expected response-body content confirms that the API returns meaningful business data rather than only a successful transport response.

Reviewing logs and changing API policies can be useful operational activities, but they are not response assertions that API Functional Monitoring performs as part of a scheduled test execution.

QUESTION NO: 6

AnyAirlines wants to create a new marketing campaign that sends customers special offers every month based on their accrued loyalty points. There is an existing integration for customer data using MuleSoft's API-led three-tier strategy. Loyalty information exists in an external system that can be accessed via an HTTP endpoint provided by the system, but has no current integration. The external ID used will be email address.

The desired output is a CSV file containing customers that includes only the top 10 percent of loyalty point holders.

What is the most efficient way to meet this requirement?

- A. 1. Have the MuleSoft team develop a new integration that includes a System API to the Loyalty system and uses the existing Customer System API.2. Create a Process API to output the final results.3. Create an Experience API for the business consumers to initiate the integration.
- B. 1. Create a MuleSoft Composer flow that utilizes the current Customer integration to select all customers.2. Create an additional MuleSoft Composer flow that retrieves all the Loyalty information.3. Create a MuleSoft Composer flow that combines the two previous results and outputs the top 10 percent to a CSV file.
- C. 1. Have the MuleSoft team develop a new integration that includes a new System API to both the Customer and Loyalty systems.2. Create a Process API to output the final results.3. Create an Experience API for the business consumers to initiate the integration.
- D. 1. Create a Salesforce Flow that retrieves the Contact data.2. Create a Salesforce Flow that retrieves the Loyalty data.3. Create a Flow Orchestration that uses the two flows and outputs the result to a CSV file.

ANSWER: A

Explanation:

Have the MuleSoft team develop a new integration that includes a System API to the Loyalty system and uses the existing Customer System API, then create a Process API to output the final results, is the appropriate API-led approach. The existing Customer System API should be reused because it already provides the managed, reusable access layer for customer data. A new Loyalty System API encapsulates the external loyalty system's HTTP endpoint, isolating its connection details, request and response formats, and any future endpoint changes from downstream consumers.

The Process API is the right layer to orchestrate the use case: it retrieves customer and loyalty records, correlates them using the email address, evaluates accrued points, determines the highest ten percent of customers, and transforms the selected results into CSV. A monthly scheduled invocation can trigger this process consistently for the campaign. An

Experience API provides a consumer-oriented interface through which business-facing consumers can initiate or access the campaign integration without being coupled to the underlying systems or orchestration.

This structure follows API-led connectivity by separating system access, cross-system business logic, and consumer-specific access while maximizing reuse of the already available customer integration. See MuleSoft's [API-led connectivity overview](#) and the Mule runtime [Scheduler documentation](#).

QUESTION NO: 7

Which Connected App scope should be used to connect RPA Manager with an Anypoint Platform account?

- A. Application Creator
- B. API Catalog Contributor
- C. RPA Integrator
- D. Exchange Administrator

ANSWER: C

Explanation:

RPA Integrator is the Connected App scope intended for establishing the connection between MuleSoft RPA Manager and an Anypoint Platform organization. MuleSoft RPA Manager uses this authorization to integrate with Anypoint Platform securely, allowing the RPA environment to associate with the organization and use the platform capabilities required for RPA-related integration and management. The scope is selected when configuring the connected app that RPA Manager uses for the connection, rather than relying on a broad general-purpose platform permission. This follows MuleSoft's least-privilege model: the connected app receives the specific authorization designed for the RPA Manager integration. After the connected app is created with this scope, its client credentials can be supplied as part of the RPA Manager connection configuration so the two products can authenticate and communicate. MuleSoft's RPA documentation describes connecting RPA Manager to Anypoint Platform and identifies the connected-app setup needed for that integration. See the [MuleSoft RPA Manager documentation](#) and the [Anypoint Platform Connected Apps documentation](#) for configuration and authorization details.