

Salesforce.Plat-Con-201.v2026-05-14.q100

Exam Code:	Plat-Con-201
Exam Name:	Salesforce Certified Omnistudio Consultant
Certification Provider:	Salesforce
Free Question Number:	100
Version:	v2026-05-14
# of views:	122
# of Questions views:	1000
https://www.dumpsdb.com/dumps/Salesforce/Plat-Con-201/Salesforce.Plat-Con-201.v2026-05-14.q100	

NEW QUESTION: 1

A consultant is working with a client who wants to display specific account information on a Flexcard. The client wants to include account status, priority level, creation date, and company logo. They also want to include actions such as closing a case, opening a new case, or creating a task. What is the best approach for the consultant to meet the client's requirements?

- A. Use OmniScripts to create a script that displays the required information.
- B. Use the Flexcard Designer to create a Flexcard, adding the necessary fields and actions.
- C. Use Integration Procedures to integrate the necessary data
- D. Use Data Mappers to transfer and transform the necessary data.

Answer: B (LEAVE A REPLY)

FlexCards are the best choice because they are designed to display account-related information such as status, priority level, creation date, and company logo while also allowing the inclusion of actions (e.g., closing a case, opening a new case, creating a task).

FlexCards provide a user-friendly UI with clickable actions that can trigger OmniScripts or Integration Procedures to process user interactions.

NEW QUESTION: 2

A company wants to track how OmniStudio component changes affect Salesforce record data over time for audit purposes. Which native Salesforce feature should be used alongside OmniStudio?

- A. Field History Tracking
- B. OmniStudio Cache Partition
- C. Type Ahead Block
- D. FlexCard Action Framework

Answer: A ([LEAVE A REPLY](#))

Field History Tracking is a native Salesforce feature that tracks changes to field values over time, regardless of whether the change originated from OmniStudio processes, Apex, or manual edits, supporting audit requirements.

NEW QUESTION: 3

A company has an existing OmniScript that gathers customer information. User feedback suggests that the current process requires too much typing because all the data is entered into input text fields. The users also are concerned because there is no validation in the data that is entered, and users can sometimes enter data that contains errors. The required inputs are first name, last name, age, and email, users are also presented a brief questionnaire regarding customer satisfaction.

Which three input elements can the consultant recommend to improve usability and to provide validation?

Choose 3 answers

- A. Headline
- B. Radio Group
- C. Email
- D. Text Area
- E. Range

Answer: B,C,E ([LEAVE A REPLY](#))

The three input elements that can be recommended to improve usability and provide validation are Radio Group, Email, and Range. Radio Group allows the user to select one option from a list of values, which reduces typing and ensures valid input. Email validates that the input is a valid email address format. Range allows the user to select a numeric value within a specified range using a slider, which reduces typing and ensures valid input. Headline is not an input element, but a display element that shows text in large font size. Text Area is not suitable for this scenario, as it allows the user to enter multiple lines of text without validation.

NEW QUESTION: 4

A business is creating a new OmniScript that will allow agents to launch a guided selling process from an account detail page. The consultant reviews the initial design proposed for the OmniScript and sees that the process the following steps:

- * Step 1: Enter account name to retrieve account information
- * Step 2: Review and edit account information
- * Step 3: Select products

* Step 4: Enter payment information

Following best practices, what suggestion should the consultant make to improve the design?

- A. Remove Step 1 and prefill the account information automatically.
- B. Divide Step 2 into two steps, so each task is distinct.
- C. Use an Edit Block in Step 4 to enter payment information.
- D. Use Multi-select elements in Step 3 for the product list.

Answer: (SHOW ANSWER)

The consultant should suggest removing Step 1 and prefilling the account information automatically to improve the design. Since the OmniScript will be launched from an account detail page, there is no need to ask the user to enter the account name again. The consultant can use an Input Parameter element to pass the account ID from the page to the OmniScript, and then use a DataRaptor Extract Action element to retrieve the account information and display it in Step 2. This will save time and avoid errors

NEW QUESTION: 5

A company has a requirement to create a 360° view of their customers using FlexCards. At this company, customer data is stored in Salesforce but also in external legacy systems. A consultant reviews the use cases needed and recommends a FlexCard canvas that contains multiple child cards inside the state of a parent FlexCard.

What is the advantage of embedding multiple child cards in this scenario?

- A. Enables easier global styling changes
- B. Allows multiple data sources to be used
- C. Enables the FlexCard to be used on a community page
- D. Allows conditional views with the Toggle element on each child card

Answer: D (LEAVE A REPLY)

The advantage of embedding multiple child cards in this scenario is that it allows multiple data sources to be used. A child card can have its own data source independent of the parent card or other child cards. This way, the consultant can use different data sources for each child card, such as Salesforce objects or external systems, and display them in one FlexCard canvas

NEW QUESTION: 6

An insurance company decides to implement a sales console for the sales representatives that displays or provide access to customer information. The consultant reviews all the information required and determines that FlexCards would be the best solution to display part of the required information. Which two pieces of data should the consultant recommend displaying using FlexCards? Choose 2 answers

- A. Policies including type, issue date, amount
- B. Terms and Conditions for each policy
- C. Guided troubleshooting process

D. Customer name, title, phone, email

Answer: A,D (LEAVE A REPLY)

The two pieces of data that the consultant should recommend displaying using FlexCards are:

Policies including type, issue date, amount and Customer name, title, phone, email. A FlexCard is a tool that can display data and actions in a card format. The consultant can use FlexCards to show important and summary information about the customer and their policies, such as name, title, phone, email, type, issue date, and amount. These pieces of data are suitable for FlexCards because they are concise and relevant for the agents

NEW QUESTION: 7

An OmniScript saves data to Salesforce and to an external system. What OmniScript element could save all this data?

A. HTTP Action

B. Integration Procedure Action

C. DataRaptor Post Action

D. DataRaptor Transform Action

Answer: B (LEAVE A REPLY)

The OmniScript element that could save all this data is Integration Procedure Action. An Integration Procedure Action can execute an Integration Procedure that can combine multiple DataRaptor actions, such as Extract, Transform, and Load, and also invoke REST or SOAP services. This way, the Integration Procedure Action can write data to Salesforce using a DataRaptor Load action and to an external system using a REST or SOAP service

NEW QUESTION: 8

Which OmniScript element retrieves Salesforce data that is then returned in Value/Label pairs and becomes available for selection in a dropdown list?

A. Lookup

B. Calculation Action

C. DataRaptor Extract Action

D. Select

Answer: D (LEAVE A REPLY)

Functionality of the Select Element: The Select element in OmniScript creates a dropdown list (or similar UI control) that allows users to choose from a set of options. It can retrieve Salesforce data directly and present it as Value/Label pairs, where the "Value" is the underlying data (e.g., a picklist value's API name or a record ID) and the "Label" is the user-friendly display text (e.g., the picklist label or a record name). The Select element supports three option source types:

Manual: Hardcoded options entered by the designer.

SObject: Retrieves options dynamically from a Salesforce object field, such as a picklist or a query result.

Custom: Uses Apex or other custom logic for advanced scenarios. When configured with the

"SObject" option source, the Select element queries Salesforce data (e.g., picklist values from a field like Industry on Account) and returns it as Value/Label pairs for the dropdown.

How It Works:

In the OmniScript Designer, you set the Select element's "Option Source" to "SObject" and specify the object (e.g., Case) and field (e.g., Reason). The element then pulls all active picklist values from that field (e.g., Value: Billing, Label: "Billing Issue") and populates the dropdown.

Alternatively, it can use a DataRaptor Extract to fetch a list of records (e.g., SELECT Id, Name FROM Account), where Id becomes the Value and Name becomes the Label. The retrieved data is automatically formatted as Value/Label pairs for user selection.

Meeting the Requirement: The Select element both retrieves Salesforce data (via direct SObject access or a DataRaptor) and presents it in a dropdown, fulfilling the question's criteria perfectly.

NEW QUESTION: 9

A consultant is working with a client who wants to display specific account information on a Flexcard.

The client wants to include account status, priority level, creation date, and company logo. They also want to include actions such as closing a case, opening a new case, or creating a task.

What is the best approach for the consultant to meet the client's requirements?

A. Use Omniscripts to ,,, script that display .. required information

B. Use the Flexcard Designer to create a Flexcard, adding the necessary fields and actions.

C. Use Integration Procedures to integrate the necessary data

D. Use Data Mappers to transfer and transform the necessary data.

Answer: B (LEAVE A REPLY)

* FlexCards (B) - FlexCards are the best choice because they are designed to display account-related information such as status, priority level, creation date, and company logo while also allowing the inclusion of actions (e.g., closing a case, opening a new case, creating a task).

* FlexCards provide a user-friendly UI with clickable actions that can trigger OmniScripts or Integration Procedures to process user interactions.

* A. Use OmniScripts - OmniScripts are for step-by-step guided interactions but are not primarily used for displaying static account information.

* C. Use Integration Procedures - Integration Procedures fetch and transform data but do not provide a user interface. They can be used within FlexCards but are not the solution on their own.

* D. Use Data Mappers - Salesforce does not have a native "Data Mappers" tool; data transformation is handled by DataRaptors or Integration Procedures.

Comprehensive and Detailed In-Depth Explanation: Why not the other options??

Salesforce OmniStudio Reference:

* FlexCards Overview

* Configuring Actions in FlexCards

NEW QUESTION: 10

A company implements an integration procedure that is invoked from an OmniScript. The integration procedure includes a very long-running process that makes users impatient. Which feature should the consultant recommend to improve performance and address users' concerns?

- A. Try Catch
- B. Batch jobs
- C. Chaining
- D. Conditions

Answer: C ([LEAVE A REPLY](#))

The feature that the consultant should recommend to improve performance and address users' concerns is chaining. Chaining is a property that allows an Integration Procedure Action element to run asynchronously in the background, without blocking the user interface. The consultant can use chaining to execute the long-running process in parallel with other actions or steps in the OmniScript, and notify the user when it is completed. This will improve the user experience and reduce waiting time

NEW QUESTION: 11

A project requires displaying data from a legacy system on a FlexCard. The external system is slow, taking more than 5 seconds for a response. The data only changes once a day. Which configuration should an OmniStudio Consultant recommend to improve the user experience?

- A. Configure a REST data source with an increased timeout setting.
- B. Configure an Integration Procedure with Chain On Step enabled.
- C. Configure a Data Mapper with filtering logic.
- D. Configure an Integration Procedure with caching enabled.

Answer: ([SHOW ANSWER](#))

Caching is the correct design because the source system is slow and the data changes only once per day. An Integration Procedure can call the external legacy API, shape the response, and cache the result so users do not wait for the same slow call every time the FlexCard loads.

Increasing the REST timeout only allows the slow response to continue; it does not improve the user experience. Data Mapper filtering is not relevant because the source is an external legacy system, not just Salesforce object retrieval. Chain On Step is for

breaking long-running Integration Procedure processing, not for reusing stable data. Salesforce Trailhead states that Integration Procedure cache stores frequently accessed, infrequently updated data, reducing round trips and improving performance.

NEW QUESTION: 12

A company implements an integration procedure that is invoked from an OmniScript. The integration procedure includes a very long-running process that makes users impatient. Which feature should the consultant recommend to improve performance and address users' concerns?

- A.** Try Catch
- B.** Batch jobs
- C.** Chaining
- D.** Conditions

Answer: C (LEAVE A REPLY)

The feature that the consultant should recommend to improve performance and address users' concerns is chaining. Chaining is a property that allows an Integration Procedure Action element to run asynchronously in the background, without blocking the user interface. The consultant can use chaining to execute the long-running process in parallel with other actions or steps in the OmniScript, and notify the user when it is completed. This will improve the user experience and reduce waiting time.

NEW QUESTION: 13

A consultant needs to design a solution that displays a customer's active insurance policies, recent claims, and available actions in a single compact view on the Account record page. Which OmniStudio component should the consultant recommend?

- A.** OmniScript
- B.** FlexCard
- C.** Integration Procedure
- D.** DataRaptor Extract

Answer: B (LEAVE A REPLY)

FlexCards are designed to display contextual, summarized information (like policies and claims) in a compact card format on a record page, along with actions the user can take. OmniScripts are for guided data-capture processes, Integration Procedures orchestrate backend logic, and DataRaptor Extract is a data retrieval tool used to power other components, not a display component itself.

NEW QUESTION: 14

How many levels deep can reusable OmniScripts be nested?

- A.** 2
- B.** 5
- C.** Unlimited

D. 1

Answer: D (LEAVE A REPLY)

Reusable OmniScripts are OmniScripts that can be embedded within other OmniScripts to create modular and reusable components. They can be nested up to unlimited levels deep, as long as there are no circular references or conflicts between the parent and child OmniScripts.

The other options are not correct because:

* 2: This is too low and does not reflect the full potential of reusable OmniScripts.

* 5: This is also too low and does not reflect the full potential of reusable OmniScripts.

* 1: This is incorrect and does not make sense, as reusable OmniScripts are meant to be nested within other OmniScripts.

NEW QUESTION: 15

A pop-up window will display additional information about the campaign, such as details about the members. A button must initiate a guided process to add new leads to the campaign. Which two FlexCard features should the consultant recommend to meet these requirements? Choose 2 answers

A. Block Element

B. OmniScript Action

C. Interaction Launcher

D. Data Table

E. Flyout Action

Answer: B,E (LEAVE A REPLY)

Flyout Action (E) - The Flyout Action in a FlexCard is used to display additional information in a pop-up window. Since the requirement mentions that a pop-up window must display additional details, the Flyout Action is the best choice for this functionality.

OmniScript Action (B) - The OmniScript Action launches an OmniScript, which is a guided interaction tool for multi-step processes. Since the requirement states that a button should initiate a guided process to add new leads, an OmniScript is the ideal choice.

NEW QUESTION: 16

Which OmniScript element can be used to retrieve data from more than one source?

A. Calculation Action

B. Integration Procedure Action

C. DataRaptor Extract Action

D. Matrix Action

Answer: B (LEAVE A REPLY)

The OmniScript element that can be used to retrieve data from more than one source is Integration Procedure Action. An Integration Procedure Action can execute an Integration Procedure that can combine multiple DataRaptor actions, such as Extract, Transform, and Load, and also invoke REST or SOAP services. This way, the Integration Procedure Action

can retrieve data from different sources, such as Salesforce objects or external systems, and store it in a JSON object for the OmniScript to display

Valid Plat-Con-201 Dumps shared by TrainingQuiz.com for Helping Passing Plat-Con-201 Exam! TrainingQuiz.com now offer the **newest Plat-Con-201 exam dumps**, the TrainingQuiz.com Plat-Con-201 exam **questions have been updated** and **answers have been corrected** get the **newest** TrainingQuiz.com Plat-Con-201 dumps with Test Engine here: <https://www.trainingquiz.com/Plat-Con-201-practice-quiz.html> (222 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 17

A business implements a simple OmniScript in their call center that allows agents to quickly create a case when on the phone with customers. The OmniScript has been running successfully in the call center for over a year. The business decides it wants to allow partners to create cases directly from their Community portal.

What is the most efficient solution that the consultant can propose to meet this new requirement?

- A. Deploy the existing OmniScript to the Community portal using OmniOut.
- B. Embed the existing OmniScript as a reusable component within a new Community OmniScript.
- C. Clone the existing OmniScript to a new LWC OmniScript and deploy to the Community portal.
- D. Duplicate the existing OmniScript and deploy the new script on the website.

Answer: (SHOW ANSWER)

The most efficient solution is to clone the existing OmniScript to a new LWC OmniScript and deploy it to the Community portal. LWC OmniScripts are designed for web-based interactions and can be easily embedded in any web page. OmniOut is not a tool for deploying OmniScripts, but a feature that allows users to export data from an OmniScript. Embedding the existing OmniScript as a reusable component within a new Community OmniScript would require creating an unnecessary wrapper script. Duplicating the existing OmniScript would create redundant code and maintenance issues.

NEW QUESTION: 18

Which OmniStudio tool creates a Chatter post and sends to a Chatter feed?

- A. DataRaptor Load
- B. Integration Procedure
- C. Calculation Procedure
- D. FlexCards

Answer: B (LEAVE A REPLY)

The OmniStudio tool that creates a Chatter post and sends it to a Chatter feed is Integration Procedure. An Integration Procedure can execute multiple DataRaptor actions, such as Extract, Transform, and Load, and also invoke REST or SOAP services. One of the REST services that an Integration Procedure can invoke is Chatter API, which allows creating and sending Chatter posts to users or groups

NEW QUESTION: 19

An enterprise architecture team is deciding between using Salesforce Flow or OmniStudio Integration Procedures for a specific integration scenario involving complex JSON transformation and multiple external API calls. Which justification supports choosing an Integration Procedure in this landscape?

- A.** Restrict JSON handling.
- B.** Perform server-side orchestration and JSON transformation.
- C.** Require HTTP callouts exclusively.
- D.** Execute entirely on the client side.

Answer: [\(SHOW ANSWER\)](#)

Integration Procedures are the stronger fit when the scenario requires server-side orchestration, multiple external API calls, and JSON transformation. They are designed to coordinate Data Mapper actions, HTTP actions, conditional logic, response shaping, and integration steps in a single server-side process. That makes them suitable for OmniStudio solutions where the client should receive a clean, optimized response rather than manage multiple complex service calls.

Option A is the opposite of the requirement because Integration Procedures support JSON handling rather than restrict it. Option C is too narrow because Integration Procedures do more than HTTP callouts. Option D is incorrect because Integration Procedures execute server-side, not entirely in the browser. Salesforce describes Integration Procedures as server-side processes for automating data operations between Salesforce and external systems.

NEW QUESTION: 20

A consultant has a project with the following requirement: " Agents need to follow a standard customer greeting script in order to ensure brand consistency. " Which OmniScript element should the consultant recommend to ensure the agents can see and follow these standard scripts?

- A.** Input Block
- B.** Text Area
- C.** Headline
- D.** Step

Answer: **C** [\(LEAVE A REPLY\)](#)

The consultant should recommend using a Headline element to ensure the agents can see and follow these standard scripts. A Headline element can display a text message with an

icon and a color, such as blue for information or yellow for warning. The consultant can use a Headline element to show the customer greeting script in the OmniScript, and instruct the agents to read it aloud

NEW QUESTION: 21

An auto insurance company has different rates for each state in the country. The company needs to manage the rates separately, but the formula to calculate the premiums is the same. All the input and output are the same; only the values in the rating tables differ. Which OmniStudio tool should the consultant recommend to meet this requirement?

- A. DataRaptor Transform with an interface Map
- B. Aggregation Steps in a Calculation Procedure
- C. Class-based Calculation Procedure
- D. Grouped Calculation Matrix

Answer: (SHOW ANSWER)

The OmniStudio tool that the consultant should recommend to meet this requirement is Grouped Calculation Matrix. A Grouped Calculation Matrix is a tool that can perform complex calculations based on multiple input variables and output values within groups. The consultant can use a Grouped Calculation Matrix to define the input variables, such as state and other factors, and the output values, such as rates. The Grouped Calculation Matrix can then apply rules and formulas to calculate the rates for each state separately, but using the same formula

NEW QUESTION: 22

A consultant is designing a solution for a client using Omnistudio. The client wants to display data and launch actions from their Salesforce org. Which OmniStudio tool should the consultant use to meet this requirement?

- A. DataRaptors
- B. FlexCards
- C. OmniScripts
- D. Integration Procedures

Answer: (SHOW ANSWER)

* FlexCards (B) - FlexCards are designed to display data from Salesforce and launch actions such as OmniScripts, Integration Procedures, or external API calls.

* FlexCards allow users to interact with data dynamically while keeping the UI simple and intuitive.

* A. DataRaptors - These are used for extracting, transforming, and loading data, but they do not have a UI component to display data.

* C. OmniScripts - OmniScripts are used for step-by-step guided interactions but are not meant for displaying data at a glance like FlexCards.

* D. Integration Procedures - These are used for server-side processing but do not display data or provide UI elements.

Comprehensive and Detailed In-Depth Explanation: Why not the other options??

Salesforce OmniStudio Reference:

* FlexCards Overview

* Launching Actions from FlexCards

NEW QUESTION: 23

An insurance company wants to create an OmniScript that allows the user to review and change account number such as phone number and website. In this process, the following functionality is needed:

- Enter the company's website
- Enter the account phone number
- Each field should display on a separate line of the page

Which three elements should the consultant include in the OmniScript design solution?

Choose 3 answers

- A. Number
- B. Text Area
- C. Telephone
- D. Line Break
- E. Text

Answer: C,D,E (LEAVE A REPLY)

The three elements that the consultant should include in the OmniScript design solution are:

Telephone, Line Break, and Text. A Telephone element can display a field for the user to enter a phone number, with validation and formatting options. A Line Break element can insert a line break between two elements, making them display on separate lines. A Text element can display a field for the user to enter text, such as a website URL

NEW QUESTION: 24

Which statement correctly describes the deployment relationship between OmniStudio components and Salesforce DX / Change Sets?

- A. OmniStudio components cannot be deployed using any Salesforce deployment tool
- B. OmniStudio components are stored as custom object records but can be deployed via Salesforce DX using the OmniStudio-specific metadata API support, or via data-based migration tools
- C. OmniStudio components can only be deployed manually by re-creating them in each org
- D. Change Sets are the only supported deployment method

Answer: B (LEAVE A REPLY)

OmniStudio components are internally stored as custom object records (not traditional metadata), but Salesforce DX provides specific support to retrieve/deploy them as

metadata-like artifacts, and data-based tools (like the OmniStudio migration tool sfdx plugin) can also be used for deployment across orgs.

NEW QUESTION: 25

A company has an existing OmniScript running in production. The business decides additional customer information is needed and wants to add more questions to the process.

What is the most efficient approach that the consultant can recommend to the business to meet this requirement?

- A.** Add a new Step element with Input elements
- B.** Add an LWC component
- C.** Add an Input Action to a Step
- D.** Add an embedded FlexCard

Answer: A (LEAVE A REPLY)

The most efficient approach that the consultant can recommend to the business to meet this requirement is to add a new Step element with Input elements. A Step element can group related fields and elements in an OmniScript, and display them in a page or a section. An Input element can display a field for the user to enter data, such as text, number, date, etc. The consultant can use a Step element to add a new page or section to the OmniScript, and then use Input elements to collect the additional customer information

NEW QUESTION: 26

An investment portfolio manager wants to build a console to display:

- Client information
- Profile attributes
- Investment instrument information

There are more than 20 profile attributes. Each investment instrument has 15 fields of information to display. The fields should be grouped logically.

How should the consultant design a solution to meet the requirements?

- A.** A header FlexCard for client information with a Flyout for profile attributes Child FlexCards for investment instruments with a Flyout for additional information
- B.** A header FlexCard for client information and profile attributes FlexCards for investment instrument information
- C.** An OmniScript with a Step for client information Block Elements for investment instruments
- D.** A header FlexCard for client information and profile attributes Child FlexCards for investment instruments with a Flyout for additional information

Answer: C (LEAVE A REPLY)

The consultant should design a solution using an OmniScript with multiple Step elements to meet the requirements. An OmniScript is a tool that can design customer interactions using elements and actions. The consultant can use an OmniScript to display client

information, profile attributes, and investment instrument information. A Step element can group related fields and elements in an OmniScript, and display them in a page or a section. The consultant can use multiple Step elements to organize the information into logical groups, such as client information, profile attributes, and investment instruments

NEW QUESTION: 27

Which OmniScript element retrieves Salesforce data that is then returned in Value/Label pairs and becomes available for selection in a dropdown list?

- A.** Lookup
- B.** Calculation Action
- C.** DataRaptor Extract Action
- D.** Select

Answer: D (LEAVE A REPLY)

Comprehensive and Detailed In-Depth Explanation: The requirement is to identify an OmniScript element that retrieves Salesforce data, formats it as Value/Label pairs, and makes it available for selection in a dropdown list. In Salesforce OmniStudio, the Select element is explicitly designed for this purpose, making it the correct answer.

Here's why D. Select is the correct answer:

* **Functionality of the Select Element:** The Select element in OmniScript creates a dropdown list (or similar UI control) that allows users to choose from a set of options. It can retrieve Salesforce data directly and present it as Value/Label pairs, where the "Value" is the underlying data (e.g., a picklist value's API name or a record ID) and the "Label" is the user-friendly display text (e.g., the picklist label or a record name). The Select element supports three option source types:

* **Manual:** Hardcoded options entered by the designer.

* **SObject:** Retrieves options dynamically from a Salesforce object field, such as a picklist or a query result.

* **Custom:** Uses Apex or other custom logic for advanced scenarios. When configured with the

"SObject" option source, the Select element queries Salesforce data (e.g., picklist values from a field like Industry on Account) and returns it as Value/Label pairs for the dropdown.

* **How It Works:**

* In the OmniScript Designer, you set the Select element's "Option Source" to "SObject" and specify the object (e.g., Case) and field (e.g., Reason). The element then pulls all active picklist values from that field (e.g., Value: Billing, Label: "Billing Issue") and populates the dropdown.

* Alternatively, it can use a DataRaptor Extract to fetch a list of records (e.g., SELECT Id, Name FROM Account), where Id becomes the Value and Name becomes the Label. The retrieved data is automatically formatted as Value/Label pairs for user selection.

* Meeting the Requirement: The Select element both retrieves Salesforce data (via direct SObject access or a DataRaptor) and presents it in a dropdown, fulfilling the question's criteria perfectly.

Now, let's examine why the other options are incorrect:

* A. Lookup: The Lookup element in OmniScript allows users to search for and select a Salesforce record (e.g., finding an Account by typing its name). While it retrieves Salesforce data and displays a list of matching records, it's designed for record selection, not for presenting a predefined set of Value/Label pairs in a dropdown. The Lookup element returns a selected record's ID and optionally other fields, but it doesn't natively format data as a dropdown list of Value/Label pairs-it's more interactive and search-driven.

* B. Calculation Action: A Calculation Action performs computations or data manipulations within an OmniScript (e.g., adding numbers or setting variables). It doesn't retrieve Salesforce data on its own (that's the role of a DataRaptor) nor does it present data in a UI component like a dropdown. It's a backend action, not a user-facing element, so it doesn't meet the requirement.

* C. DataRaptor Extract Action: This option (likely a typo for "DataRaptor Extract Action" given the OmniStudio context) refers to an OmniScript action that uses a DataRaptor Extract to retrieve Salesforce data. While it can fetch data and potentially structure it as Value/Label pairs (if the DataRaptor is configured to query a picklist field or map Id and Name), it's not an "element" that displays a dropdown-it's an action that supplies data to other elements (like Select). The Select element uses this data, but the DataRaptor Extract Action itself doesn't render the UI.

Key Distinction:

The Select element is the only option listed that is both an OmniScript element (a UI component) and capable of retrieving Salesforce data (either directly or via a DataRaptor) to populate a dropdown with Value/Label pairs. Other tools like DataRaptors support the process, but Select is the end-point for display and interaction.

References:

* Salesforce OmniStudio Documentation: OmniScript Elements Reference - Details the Select element's ability to retrieve Salesforce data as Value/Label pairs for dropdowns.

* Salesforce OmniStudio Developer Guide: Select Element Configuration - Explains SObject and DataRaptor integration for populating options.

* Salesforce Help: OmniScript Designer - Describes how Select differs from Lookup and actions like Calculation or DataRaptor Extract.

NEW QUESTION: 28

A company needs to create a quoting process for its internal agents. During quoting, the agent selects a product that is passed to the OmniScript with product details including the unit price, and specifies the grade (A, B, C, D, E) and size (Small, Medium, Large, X-Large). The process should use the grade and size to look up a discount factor, then

multiply the unit price by that discount to return the quote. Which two tools should be used to meet these requirements?

Choose 2 answers

- A.** DataRaptor Transform
- B.** Expression Set
- C.** Decision Matrix
- D.** DataRaptor Extract

Answer: ([SHOW ANSWER](#))

Comprehensive and Detailed In-Depth Explanation: The requirement involves creating a quoting process within an OmniScript where an agent selects a product (with a unit price), specifies grade and size, looks up a discount factor based on those inputs, and calculates the final quote. Two OmniStudio tools are needed to achieve this: Decision Matrix and DataRaptor Extract.

Here's why C. Decision Matrix and D. DataRaptor Extract are the correct answers:

* C. Decision Matrix:

* Purpose: A Decision Matrix in OmniStudio is a tool used to look up values based on multiple input conditions, returning a result that can be used in calculations or processes. It's essentially a configurable lookup table stored as a Salesforce custom object (DecisionMatrixDefinition).

* Application: In this scenario, the Decision Matrix is ideal for determining the discount factor based on the combination of grade (A, B, C, D, E) and size (Small, Medium, Large, X-Large).

For example:

* Grade A + Small = 0.9 (10% discount)

* Grade B + Large = 0.85 (15% discount)

* And so on for all 20 combinations (5 grades × 4 sizes).

* Integration with OmniScript: The OmniScript can call the Decision Matrix via an Integration Procedure or directly reference it in a Calculation Action, passing Grade and Size as inputs and receiving the DiscountFactor as output.

* Why It Fits: The requirement explicitly states "look up a discount factor," and Decision Matrix is purpose-built for such multi-variable lookups, making it more efficient than manual coding or other tools.

* D. DataRaptor Extract:

* Purpose: A DataRaptor Extract retrieves data from Salesforce objects and maps it into a JSON format usable by OmniScripts or other components.

* Application: The product details, including the UnitPrice, are passed to the OmniScript when the agent selects a product. A DataRaptor Extract is needed to fetch this product data from a Salesforce object (e.g., Product2 or a custom object like QuoteLineItem__c) based on the selected product's ID. The extracted data (e.g., { "ProductId": "01t...", "UnitPrice": 100 }) is then available in the OmniScript's data JSON.

* **Role in Calculation:** After the Decision Matrix provides the discount factor, the OmniScript can use a Calculation Action to multiply the UnitPrice (from the DataRaptor Extract) by the DiscountFactor (from the Decision Matrix) to compute the quote (e.g., $100 * 0.9 = 90$).

* **Why It Fits:** The process starts with product selection, and DataRaptor Extract is the standard OmniStudio tool for retrieving Salesforce data like unit price.

* **How They Work Together:**

* **DataRaptor Extract:** Fetches product details (e.g., UnitPrice) when the agent selects a product.

* **Decision Matrix:** Looks up the discount factor based on Grade and Size inputs.

* **Calculation Action in OmniScript:** Multiplies UnitPrice by DiscountFactor to return the quote.

Now, let's examine why the other options are incorrect:

* **A. DataRaptor Transform:** A DataRaptor Transform manipulates or reshapes data (e.g., converting JSON structures or applying formulas) but doesn't retrieve data from Salesforce or perform lookups like a Decision Matrix. While it could theoretically calculate the quote after data is fetched, it's not needed here since a Calculation Action within OmniScript can handle the multiplication, and it doesn't address the lookup requirement.

* **B. Expression Set:** An Expression Set defines reusable formulas or conditions in OmniStudio, often used in Integration Procedures or Calculations. While it could compute the final quote (e.g., $\text{UnitPrice} * \text{DiscountFactor}$), it doesn't retrieve data or perform lookups based on grade and size. It's a supporting tool, not a primary solution for this requirement.

References:

* **Salesforce OmniStudio Documentation: Decision Matrix Overview** - Describes using Decision Matrices for multi-condition lookups.

* **Salesforce OmniStudio Developer Guide: DataRaptor Extract** - Details retrieving Salesforce data for OmniScript use.

NEW QUESTION: 29

A client wants historical versions of their Integration Procedures and DataRaptors to be tracked and easily rolled back if needed. What should the consultant recommend as part of the implementation approach?

A. Rely solely on Setup Audit Trail

B. Use a version control strategy with Salesforce DX and source-tracked metadata (e.g., Git)

C. Manually export data weekly

D. Disable versioning within OmniStudio

Answer: (SHOW ANSWER)

OmniStudio components are metadata that can be retrieved and deployed using Salesforce DX, enabling proper version control through Git and standard release management practices, including rollback capability.

NEW QUESTION: 30

A consultant wants to proceed with a FlexCard canvas design that will display data from a legacy billing system. The legacy billing system is being extended to add a SOAP endpoint that will allow for the data to be retrieved in XML format. However, the endpoint is not ready yet.

Which data source should the team use to make progress on the design?

- A. Apex Remote
- B. DataRaptor
- C. Integration Procedures
- D. Apex REST

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 31

Which element allows a user to retrieve data from a single field and display it in a dropdown list?

- A. Calculation Action
- B. DataRaptor Extract Action
- C. Lookup
- D. Select

Answer: C ([LEAVE A REPLY](#))

The element that allows a user to retrieve data from a single field and display it in a dropdown list is Lookup. Lookup allows the user to search for a value from a Salesforce object or an external system and select it from a list of suggestions. Calculation Action is used to perform calculations on data. DataRaptor Extract Action is used to retrieve data from multiple fields or records. Select is used to display a list of predefined values.

Valid Plat-Con-201 Dumps shared by TrainingQuiz.com for Helping Passing Plat-Con-201 Exam! TrainingQuiz.com now offer the **newest Plat-Con-201 exam dumps**, the TrainingQuiz.com Plat-Con-201 exam **questions have been updated** and **answers have been corrected** get the **newest** TrainingQuiz.com Plat-Con-201 dumps with Test Engine here: <https://www.trainingquiz.com/Plat-Con-201-practice-quiz.html> (222 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 32

A company needs to create a new quote estimation process for its services team. This process involves multi; steps. The user needs to enter a variable number of rows of data in order to calculate the correct final price f the quote. The data that needs to be entered in each row includes:

* Service category

* Number of hours

* Hourly rate

The number of hours should be multiplied with the hourly rate to calculate the total for each row.

Which three OmniScript elements should the consultant recommend to meet these requirements?

Choose 3 answers

A. Edit Block

B. Multi-select

C. Formula

D. Step

E. Input Block

Answer: A,C,E (LEAVE A REPLY)

To create a new quote estimation process that involves multiple steps and a variable number of rows of data, the consultant should recommend the following OmniScript elements:

* Edit Block: This is an element that allows the user to add, edit, or delete multiple rows of data in a table-like format. It can be used to capture the service category, number of hours, and hourly rate for each row.

* Formula: This is an element that allows the user to perform calculations on data using mathematical expressions. It can be used to calculate the total for each row by multiplying the number of hours with the hourly rate.

* Input Block: This is an element that allows the user to enter data using various input types, such as text, number, date, picklist, etc. It can be used to display the formula result for each row and the final price for the quote.

The elements that are not suitable for this scenario are:

* Multi-select: This is an element that allows the user to select multiple options from a list. It is not relevant for this scenario, as the user needs to enter data in each row, not select from predefined options.

* Step: This is an element that organizes the OmniScript into one or more pages. It is not relevant for this scenario, as it does not capture or display any data.

NEW QUESTION: 33

A Salesforce OmniStudio Consultant is creating an OmniScript for a client's customer service process. The consultant wants to enhance the performance and usability of the OmniScript. What is one best practice the consultant can follow to achieve this?

A. Use the Salesforce Lightning App Builder.

B. Use custom Lightning Web Components with LWC OmniScripts.

C. Use Angular OmniScripts.

D. Use the Classic OmniScript Designer.

Answer: B (LEAVE A REPLY)

LWC OmniScripts Overview: OmniStudio supports two types of OmniScripts: Standard OmniScripts (built on AngularJS) and LWC OmniScripts (built on Lightning Web Components).

LWC OmniScripts are the newer, recommended approach, offering better performance due to their lightweight framework and Salesforce's native LWC architecture.

Custom LWCs: By integrating custom Lightning Web Components into an LWC OmniScript, the consultant can:

Enhance Performance: LWCs are faster than Angular-based components because they use the browser's native Web Components standard, reducing overhead and improving load times.

Improve Usability: Custom LWCs allow for highly tailored UI elements or logic (e.g., a dynamic form or real-time validation) that go beyond standard OmniScript elements, making the customer service process more intuitive and efficient.

Example: A custom LWC could display a collapsible FAQ section or a real-time chat widget within the OmniScript, improving the agent's workflow.

Best Practice Alignment: The OmniStudio documentation explicitly recommends transitioning to LWC OmniScripts for new implementations and using custom LWCs to extend functionality, as they align with Salesforce's long-term platform strategy and provide a responsive, scalable solution.

NEW QUESTION: 34

A healthcare company wants to enable its subscribers to add, edit, or delete dependents related to their policy via their community portal. The project team decides to use OmniStudio tools to provide this functionality. In this scenario, which two OmniStudio features should the consultant recommend? Choose 2 answers

- A.** Datatable
- B.** Remote Action
- C.** Response Action
- D.** Edit Block

Answer: C,D (LEAVE A REPLY)

The two OmniScript features that the consultant should recommend are Response Action and Edit Block. A Response Action is an action that can save or submit data from an OmniScript to a Salesforce object or an external system. The consultant can use a Response Action to allow the user to add, edit, or delete dependents related to their policy via their Community portal. An Edit Block is an element that can display a list of records for the user to view, edit, or delete, using an OmniScript or an Integration Procedure. The consultant can use an Edit Block to display the dependents related to the policy, and allow the user to modify them

NEW QUESTION: 35

An organization needs to control pages or groups of fields in an OmniScript based on the choices made by the user. What should an OmniStudio Consultant do to meet this requirement?

- A. Use the Visibility Rules property on a Step or Block element.
- B. Create child OmniScripts for branched flow.
- C. Use the Conditional View property on a Step or Block element.
- D. Use the Radio Input property on a Step or Block element.

Answer: C ([LEAVE A REPLY](#))

The verified answer is the Conditional View property on a Step or Block element. In OmniScript, conditional display logic is used to hide or show an element, a group of elements, a block, or a step based on user input or other data conditions. This directly satisfies the requirement to control pages or groups of fields based on choices made by the user. Creating child OmniScripts is excessive for normal branching visibility. Radio Input is only an input control; it can provide the selected value, but it does not itself control page or field visibility. "Visibility Rules" is not the precise OmniStudio property name used in the official documentation for this function.

NEW QUESTION: 36

What is the primary purpose of a Type Ahead Block element in an OmniScript?

- A. To pre-fill an entire form with existing record data
- B. To validate user input against a regular expression
- C. To trigger a Calculation Procedure on field change
- D. To provide auto-complete suggestions as a user types, pulling data from a DataRaptor or Integration Procedure

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 37

A business has a requirement to display cases in a console for service agents. Cases can have a variety of statuses, including Active, Closed, or Escalated. When a case is Closed, agents need to be able to reopen the case. When the case is Active or Escalated, agents should not have the option to reopen the case. Which FlexCard functionality can be used to meet this requirement?

- A. Conditional View
- B. Flyouts
- C. State
- D. Styling

Answer: ([SHOW ANSWER](#))

The FlexCard functionality that can be used to meet this requirement is Conditional View. A Conditional View is a property that determines whether a FlexCard element, such as a field or an action, is visible or hidden based on certain conditions. The consultant can use a Conditional View to show or hide the reopen case action based on the status of the case.

NEW QUESTION: 38

A consultant is working with a client who wants to use a Flexcard to display data from a specific Salesforce object. The client is unsure which data source. What advice should the consultant give to the client?

- A.** The client should choose a data source based on the speed of data retrieval
- B.** The client should choose the data source that retrieves the most data.
- C.** The client should choose a data source based on the number of records it can retrieve.
- D.** The client should choose a data source that can retrieve data from the specific Salesforce object.

Answer: D (LEAVE A REPLY)

A FlexCard retrieves data from a Salesforce object using a data source such as DataRaptor Extract, Integration Procedure, or a direct Salesforce Object Query.

The correct approach is to choose a data source that specifically retrieves data from the required Salesforce object (D).

This ensures relevant data is fetched efficiently while maintaining performance and scalability.

NEW QUESTION: 39

An OmniStudio Consultant is reviewing a solution where an OmniScript calls 10 separate Data Mappers directly from the client-side browser to fetch various data points. Which architectural change should the consultant recommend to align with enterprise best practices?

- A.** Consolidate the data requests into a single Integration Procedure to reduce browser network traffic.
- B.** Replace the Data Mappers with Apex classes that are called directly from the browser.
- C.** Keep the design as is, because client-side calls are always faster than server-side calls.
- D.** Use a FlexCard to fetch the data and pass it to the OmniScript.

Answer: A (LEAVE A REPLY)

The consultant should consolidate the 10 client-side calls into one server-side Integration Procedure. Calling many Data Mappers directly from the browser increases network traffic, creates more round trips, complicates error handling, and exposes the UI to unnecessary latency.

An Integration Procedure can orchestrate multiple Data Mapper actions and external calls in a single server-side transaction, then return a shaped response to the OmniScript.

Replacing the Data Mappers with Apex is unnecessary unless the requirement cannot be met declaratively.

Keeping the current design is weak architecture because client-side calls are not automatically faster. A FlexCard is a display component, not the correct orchestration layer for OmniScript data fetching. Salesforce describes Integration Procedures as server-side

processes that support API calls, transformations, conditional logic, and chaining in a single server call.

NEW QUESTION: 40

A healthcare company wants to enable its subscribers to add, edit, or delete dependents related to their policy via their community portal. The project team decides to use OmniStudio tools to provide this functionality.

In this scenario, which two OmniStudio features should the consultant recommend?

Choose 2 answers

- A.** Datatable
- B.** Remote Action
- C.** Response Action
- D.** Edit Block

Answer: ([SHOW ANSWER](#))

The two OmniScript features that the consultant should recommend are Response Action and Edit Block. A Response Action is an action that can save or submit data from an OmniScript to a Salesforce object or an external system. The consultant can use a Response Action to allow the user to add, edit, or delete dependents related to their policy via their Community portal. An Edit Block is an element that can display a list of records for the user to view, edit, or delete, using an OmniScript or an Integration Procedure. The consultant can use an Edit Block to display the dependents related to the policy, and allow the user to modify them

NEW QUESTION: 41

A company needs to create a quoting process for its internal agents. During quoting, the agent selects a product that is passed to the OmniScript with product details including the unit price, and specifies the grade (A, B, C, D, E) and size (Small, Medium, Large, X-Large). The process should use the grade and size to look up a discount factor, then multiply the unit price by that discount to return the quote. Which two tools should be used to meet these requirements?

Choose 2 answers

- A.** DataRaptor Transform
- B.** Expression Set
- C.** Decision Matrix
- D.** DataRaptor Extract

Answer: ([SHOW ANSWER](#))

C). Decision Matrix:

Purpose: A Decision Matrix in OmniStudio is a tool used to look up values based on multiple input conditions, returning a result that can be used in calculations or processes. It's essentially a configurable lookup table stored as a Salesforce custom object (DecisionMatrixDefinition).

Application: In this scenario, the Decision Matrix is ideal for determining the discount factor based on the combination of grade (A, B, C, D, E) and size (Small, Medium, Large, X-Large). For example:

Grade A + Small = 0.9 (10% discount)

Grade B + Large = 0.85 (15% discount)

And so on for all 20 combinations (5 grades * 4 sizes).

Integration with OmniScript: The OmniScript can call the Decision Matrix via an Integration Procedure or directly reference it in a Calculation Action, passing Grade and Size as inputs and receiving the DiscountFactor as output.

Why It Fits: The requirement explicitly states "look up a discount factor," and Decision Matrix is purpose-built for such multi-variable lookups, making it more efficient than manual coding or other tools.

D). DataRaptor Extract:

Purpose: A DataRaptor Extract retrieves data from Salesforce objects and maps it into a JSON format usable by OmniScripts or other components.

Application: The product details, including the UnitPrice, are passed to the OmniScript when the agent selects a product. A DataRaptor Extract is needed to fetch this product data from a Salesforce object (e.g., Product2 or a custom object like QuoteLineItem__c) based on the selected product's ID. The extracted data (e.g., { "ProductId": "01t...", "UnitPrice": 100 }) is then available in the OmniScript's data JSON.

Role in Calculation: After the Decision Matrix provides the discount factor, the OmniScript can use a Calculation Action to multiply the UnitPrice (from the DataRaptor Extract) by the DiscountFactor (from the Decision Matrix) to compute the quote (e.g., $100 * 0.9 = 90$).

Why It Fits: The process starts with product selection, and DataRaptor Extract is the standard OmniStudio tool for retrieving Salesforce data like unit price.

How They Work Together:

DataRaptor Extract: Fetches product details (e.g., UnitPrice) when the agent selects a product.

Decision Matrix: Looks up the discount factor based on Grade and Size inputs.

Calculation Action in OmniScript: Multiplies UnitPrice by DiscountFactor to return the quote.

NEW QUESTION: 42

Which three use cases should be implemented using Calculation Procedures and matrices?

Choose 3 answers

A. Use a house's address, size, and age of the building to determine an insurance premium.

B. Use rules to determine eligible insurance products based on a house's address and age of the building.

C. Use location and past usage to determine the monthly cost for an energy product.

D. Use the product color and capacity to determine the price of a product.

E. Use risk factors for an insured item to determine different insurance product options.

Answer: A,C,E (LEAVE A REPLY)

Calculation Procedures & Matrices are used to perform complex calculations based on input data and predefined rules. They are suitable for use cases that involve pricing, rating, scoring, or eligibility determination. Therefore, the use cases that should be implemented using Calculation Procedures & Matrices are:

Use a house's address, size, and age of the building to determine an insurance premium.

(This involves rating based on multiple factors) Use location and past usage to determine the monthly cost for an energy product. (This involves pricing based on variable inputs)

Use risk factors for an insured item to determine different insurance product options. (This involves scoring and eligibility based on criteria) The use cases that should not be implemented using Calculation Procedures & Matrices are:

Use rules to determine eligible insurance products based on a house's address and age of the building. (This can be done using Business Rules Engine or OmniScript logic) Use the product color and capacity to determine the price of a product. (This can be done using simple formulas or lookup tables)

NEW QUESTION: 43

A company has a process that requires a birthday validation. At the beginning of the process, the user is asked to input their birthday. If the user is less than 18 years old. then the process should display an error stating that the age cannot be less than 18. The process should restrict users from continuing. If the user is 18 or older, then the process should proceed without any error message.

Which two OmniScript features should the consultant recommend to meet this validation requirement?

Choose 2 answers

A. Calculation

B. Messaging

C. Alert

D. Formula

Answer: (SHOW ANSWER)

The two OmniScript features that the consultant should recommend to meet this validation requirement are:

Messaging and Formula. A Messaging element is an element that can display a text message with an icon and a color, such as blue for information or yellow for warning. The consultant can use a Messaging element to display an error message stating that the age cannot be less than 18 if the user's birthday indicates they are under 18 years old. A Formula element is an element that can perform calculations or validations on data values using expressions or functions. The consultant can use a Formula element to validate the

user's birthday by subtracting it from the current date and checking if the result is greater than or equal to 18 years

NEW QUESTION: 44

A company has an existing OmniScript that agents use to create new billing accounts. It currently has three steps to capture the required information:

Step 1: account name and legal number

Step 2: billing cycle and monthly due date

Step 3: email and telephone number

After each step, a DataRaptor is used to update the account information in Salesforce.

Following best practices, which two improvements can the consultant recommend for this OmniScript? Choose 2 answers

A. Create a reusable OmniScript for this process

B. Use a single DataRaptor to save the information

C. Combine the three steps into one step

D. Configure the save for later property

Answer: B,C (LEAVE A REPLY)

When designing OmniScripts, the best practices that should be followed to optimize performance and user experience are:

Use a single DataRaptor to save the information: Instead of using multiple DataRaptors to update the account information after each step, it is better to use a single DataRaptor at the end of the OmniScript to save all the information at once. This reduces the number of database operations and improves performance.

Combine the three steps into one step: Instead of having three separate steps to capture the required information, it is better to combine them into one step with a simple layout. This reduces the number of page transitions and improves user experience.

NEW QUESTION: 45

What is the purpose of Step elements in an OmniScript?

A. Organizes the script into one or more pages

B. Groups elements that extract data

C. Enables the use of repeatable blocks

D. Allows the user to input data

Answer: A (LEAVE A REPLY)

Step elements (A) - The Step element is used in OmniScripts to break down the script into multiple pages or sections. This helps organize the user flow and enhances the user experience by making long processes more manageable.

NEW QUESTION: 46

Which two OmniScript components should the consultant recommend using to meet this validation requirement? Choose 2 answers

- A. Messaging
- B. Alert
- C. Calculation
- D. Formula

Answer: C,D (LEAVE A REPLY)

C: The Calculation component (specifically, a Calculation Action) in OmniScript is used to perform operations on data, including validation logic. It allows you to execute calculations, manipulate data, and set conditions based on user inputs or retrieved data. For example, a Calculation Action can check if a numeric input exceeds a threshold (e.g., Quantity > 100) or if a text field matches a required format. It can then set a flag (e.g., isValid = true/false) that can be used to control the OmniScript flow--such as displaying an error or blocking navigation. Calculation Actions are highly versatile because they support OmniScript's JSON data structure and can integrate with external data sources via DataRaptors or Integration Procedures. According to the OmniStudio documentation, Calculation Actions are ideal for complex validations requiring multiple steps or data transformations.

D: The Formula component is a lightweight, inline element in OmniScript used to evaluate expressions and perform simple validations directly within the script. It leverages OmniScript's formula syntax (similar to Salesforce formulas) to compute values or check conditions. For instance, a Formula can validate that a date input is not in the past (e.g., TODAY() <= InputDate) or that a text field contains a specific substring. Unlike Calculation Actions, Formulas are embedded within a specific field or step and are best for straightforward, single-expression validations. The result of a Formula can be used to show/hide elements, set field values, or trigger other actions, making it a key tool for real-time validation.

Valid Plat-Con-201 Dumps shared by TrainingQuiz.com for Helping Passing Plat-Con-201 Exam! TrainingQuiz.com now offer the **newest Plat-Con-201 exam dumps**, the TrainingQuiz.com Plat-Con-201 exam **questions have been updated** and **answers have been corrected** get the **newest** TrainingQuiz.com Plat-Con-201 dumps with Test Engine here: <https://www.trainingquiz.com/Plat-Con-201-practice-quiz.html> (222 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 47

Service agents must confirm customer contact information in the first step of a payment OmniScript. Contact information includes name, telephone number, mobile number, and email. None of the contact information on first step is required.

On the last step, after taking payment, the agent can optionally email the receipt to the customer. If the customer says yes, the agent selects a checkbox. If the agent selects the

box but the email address field is empty, the process must require the user to return to the first step and enter the customer's email address.

What should the consultant recommend to meet this requirement?

- A. Add a Set Errors element
- B. Add a DataRaptor to retrieve the email address
- C. Add a Validation Rule to the Contact object
- D. Add a Conditional View to the last step

Answer: A (LEAVE A REPLY)

The consultant should recommend adding a Set Errors element to meet this requirement.

A Set Errors element can display an error message and prevent the OmniScript from proceeding if certain conditions are not met.

The consultant can use a Set Errors element to check if the email address field is empty when the agent selects the checkbox to email the receipt. If it is empty, the Set Errors element can show an error message and direct the user to return to the first step and enter the customer's email address

NEW QUESTION: 48

A company has designed a process that extracts a large amount of data from a Salesforce object.

Due to the volume of data that will be retrieved, it is important that Governor limits are not exceeded. What solution should the consultant recommend?

- A. DataRaptor Extract with Chaining
- B. DataRaptor Turbo Extract
- C. Calculation Procedure and DataRaptor Turbo Extract
- D. Integration Procedure and DataRaptor Turbo Extract

Answer: B (LEAVE A REPLY)

The solution that the consultant should recommend is DataRaptor Turbo Extract.

DataRaptor Turbo Extract allows the designer to retrieve a large amount of data from a Salesforce object in batch mode without hitting governor limits. DataRaptor Extract is not suitable for this scenario, as it can only retrieve data from a single record or a small set of records. Calculation Procedure and DataRaptor Turbo Extract are not needed for this scenario, as there is no calculation involved.

Integration Procedure and DataRaptor Turbo Extract are not needed for this scenario, as there is no data integration involved.

NEW QUESTION: 49

A company needs to create an OmniScript with three steps.

- * Step 1
- * Step 2
- * Step 3

Enter customer details

Select a product

Display a price retrieved from a REST API

The product's price should automatically display in Step 3.

What should the consultant add to the design of the OmniScript to meet these requirements?

- A. DataRaptor Post Action between Step 2 and Step 3
- B. Integration Procedure Action between Step 2 and Step 3
- C. DataRaptor Post Action inside Step 2
- D. Integration Procedure Action inside Step 3

Answer: D (LEAVE A REPLY)

The consultant should add an Integration Procedure Action inside Step 3 to meet the requirements. An Integration Procedure Action can execute an Integration Procedure that invokes a REST API and retrieves the product price. The Integration Procedure Action can be configured to run automatically when Step 3 is loaded, and display the price in a field or label element

NEW QUESTION: 50

A company has an existing OmniScript that agents use to create new billing accounts. It currently has three steps to capture the required information:

- * Step 1: account name and legal number
- * Step 2: billing cycle and monthly due date
- * Step 3: email and telephone number

After each step, a DataRaptor is used to update the account information in Salesforce.

Following best practices, which two improvements can the consultant recommend for this OmniScript?

Choose 2 answers

- A. Create a reusable OmniScript for this process
- B. Use a single DataRaptor to save the information
- C. Combine the three steps into one step
- D. Configure the save for later property

Answer: B,C (LEAVE A REPLY)

When designing OmniScripts, the best practices that should be followed to optimize performance and user experience are:

- * Use a single DataRaptor to save the information: Instead of using multiple DataRaptors to update the account information after each step, it is better to use a single DataRaptor at the end of the OmniScript to save all the information at once. This reduces the number of database operations and improves performance.
- * Combine the three steps into one step: Instead of having three separate steps to capture the required information, it is better to combine them into one step with a simple layout. This reduces the number of page transitions and improves user experience.

The suggestions that are not best practices are:

- * Create a reusable OmniScript for this process: This may not be necessary if the process is specific to creating new billing accounts and not used by other OmniScripts. Creating a reusable OmniScript may add complexity and maintenance overhead.
- * Configure the save for later property: This may not be useful if the process is short and simple and does not require the user to resume it later. Configuring the save for later property may add unnecessary storage and retrieval logic.

NEW QUESTION: 51

A business Implements FlexCards in their customer 360° view. The business wants to add a new action to on the FlexCards that redirects users to an external web page. Which type of action should the consultant recommend to meet this new requirement?

- A.** Redirect
- B.** Event
- C.** Navigate
- D.** URL

Answer: C (LEAVE A REPLY)

The type of action that the consultant should recommend to meet this new requirement is Navigate. A Navigate action is a button or a link that can redirect the user to another web page or URL. The consultant can use a Navigate action to add a new action to the FlexCards that redirects users to an external web page

NEW QUESTION: 52

A Salesforce Omnistudio Consultant is validating an Omniscript they have developed for a client. The client has specific business requirements that the script must meet.

What is the best way for the consultant to ensure that the Omniscript meets the client's business requirements?

- A.** Implement the Omniscript without comparing it with the client's business requirements.
- B.** Assume that the Omniscript meets the client's business requirements without conducting a review.
- C.** Conduct a thorough review of the Omniscript, comparing each step with the client's business requirements.
- D.** Ask the client to review the Omniscript with their internal team.

Answer: (SHOW ANSWER)

* Conducting a thorough review (C) - The best way to ensure the OmniScript meets the business requirements is to review each step and compare it with the documented requirements.

* This includes validating the data flow, user interactions, and integration points to make sure the script functions as intended.

* A. Implement the OmniScript without comparing it with the client's business requirements
- Skipping the review process can lead to errors and misalignment with business needs.

* B. Assume that the OmniScript meets the client's business requirements without conducting a review - This is risky and unprofessional; assumptions should not replace validation.

* D. Ask the client to review the OmniScript with their internal team - While client feedback is important, the consultant must first conduct their own thorough review before involving the client.

Comprehensive and Detailed In-Depth Explanation: Why not the other options??

Salesforce OmniStudio Reference:

* OmniScript Best Practices & Validation

* OmniScript Testing & Debugging

NEW QUESTION: 53

A solution architect is working on a digital transformation project for a client. The client has complex business requirements that need to be implemented using Omnistudio tools. The solution architect needs to decide which Omnistudio tool to use.

Which Omnistudio tool should the solution architect use to automate the client's business processes and create guided interactions using data from the Salesforce org and external sources?

A. Integration procedures

B. Flexcards

C. Expression Sets

D. Omniscripts

Answer: (SHOW ANSWER)

* OmniScripts (D) are designed to automate complex business processes and guide users through interactions using both Salesforce and external data sources.

* They can fetch data dynamically, apply business logic, and execute actions such as record updates, API calls, or guided user inputs.

* OmniScripts integrate seamlessly with DataRaptors, Integration Procedures, and FlexCards to provide a complete business solution.

* A. Integration Procedures - These are used for server-side processing and API calls, but they do not provide user interaction like OmniScripts.

* B. FlexCards - These display data but do not automate guided interactions.

* C. Expression Sets - These are used for calculations and business rules, but they do not provide guided interactions.

Comprehensive and Detailed In-Depth Explanation: Why not the other options??

Salesforce OmniStudio Reference:

* OmniScripts Overview

* Using OmniScripts to Automate Business Processes

NEW QUESTION: 54

What business problem does DataRaptor solve?

- A. It removes the need to code data mappings for data transformations.
- B. It allows developers to create complex API queries declaratively.
- C. It combines multiple steps and processes into a single server call.
- D. It guides humans through a complex business process.

Answer: ([SHOW ANSWER](#))

The business problem that DataRaptor solves is that it removes the need to code data mappings for data transformations. DataRaptor allows the designer to declaratively define how data is read, transformed, and written between Salesforce and external systems. It does not allow developers to create complex API queries declaratively, as this is done by Integration Procedures. It does not combine multiple steps and processes into a single server call, as this is done by Calculation Procedures. It does not guide humans through a complex business process, as this is done by OmniScripts.

NEW QUESTION: 55

Coral Cloud Retail Stores Inc. is planning a new digital engagement project. Its IT policy requires using platform-native features that receive automatic updates with Salesforce seasonal releases to minimize manual maintenance. Which OmniStudio runtime environment should an OmniStudio Consultant recommend to meet this requirement?

- A. A custom-built runtime environment using unlocked packages.
- B. The Standard OmniStudio runtime, because its updates are part of Salesforce releases.
- C. The OmniStudio Managed Package, because it provides the latest features.
- D. The OmniStudio Managed Package with an automated installation script.

Answer: ([SHOW ANSWER](#))

The consultant should recommend the Standard OmniStudio runtime. The key requirement is platform-native behavior with automatic updates through Salesforce seasonal releases. Salesforce states that core releases apply automatically to orgs using the OmniStudio standard runtime and not the managed package model. A custom-built runtime with unlocked packages would increase maintenance, not reduce it. The managed package model historically required package-based upgrades and is not the best fit when the stated policy is to minimize manual maintenance through Salesforce release alignment. An automated installation script still depends on package-style maintenance. Standard runtime is the correct fit because it aligns OmniStudio execution with Salesforce's native release lifecycle.

NEW QUESTION: 56

A business has been experiencing a downturn in customer satisfaction due to billing Issues. The business learn that when customers are allowed to schedule an inquiry call with an agent, customer satisfaction improves.

For this reason, the business decides to create an OmniScript that asks the customer to rate their customer satisfaction using a 1-5 rating scale. If the customer satisfaction is less than 2, it should allow the customer to request a call back on a certain date and then create

a case that includes a list of bills from an external system the last 5 months in the case description.

In what order should the consultant design the elements of the OmniScript to meet these requirements?

- A.** Radio Input, Date Input, HTTP Action, DataRaptor Post Action
- B.** Date Input, Radio Input, HTTP Action, DataRaptor Post Action
- C.** HTTP Action, DataRaptor Post Action, Date Input, Radio Input
- D.** HTTP Action, Radio Input, Date Input, DataRaptor Post Action

Answer: ([SHOW ANSWER](#))

The order that the consultant should design the elements of the OmniScript to meet these requirements is:

Radio Input, Date Input, HTTP Action, DataRaptor Post Action. A Radio Input element can display a list of options for the user to choose from, such as customer satisfaction rating. A Radio Input element allows only one option to be selected at a time, and can also have icons for each option. A Date Input element can display a field for the user to enter a date value, such as call back date. A Date Input element can also have validation and formatting options, such as minimum and maximum dates, calendar picker, etc. An HTTP Action element can invoke a REST or SOAP service to retrieve data from an external system, such as a list of bills from the last 5 months. An HTTP Action element can store the response data in a JSON object for further processing. A DataRaptor Post Action element can write data to a Salesforce object or invoke a Salesforce API, such as creating a case with the list of bills in the case description

NEW QUESTION: 57

A business wants to create a FlexCard for mobile plans to add to their Customer 360° console application.

The FlexCard needs to include the following actions:

- * Start a process to retrieve plan consumption data
- * Create a new case
- * Open a promotions web page
- * Change the SIM card

which combination should the consultant use in designing the solution?

- A.** Custom Event and Redirect URL
- B.** OmniScript and Navigate
- C.** Event, Navigate and Card
- D.** Flyout and OmniScript

Answer: ([SHOW ANSWER](#))

The combination that the consultant should use in designing the solution is OmniScript and Navigate. An OmniScript is a tool that can design customer interactions using elements and actions. The consultant can use an OmniScript to display the mobile plans for the Customer 360° console application. A Navigate action is a button or a link that can redirect

the user to another web page or URL. The consultant can use Navigate actions to add the following actions to the FlexCard: start a process to retrieve plan consumption data, create a new case, open a promotions web page, and change the SIM card

NEW QUESTION: 58

A business implements a simple OmniScript in their call center that allows agents to quickly create a case when on the phone with customers. The OmniScript has been running successfully in the call center for over a year. The business decides it wants to allow partners to create cases directly from their Community portal. What is the most efficient solution that the consultant can propose to meet this new requirement?

- A.** Deploy the existing OmniScript to the Community portal using OmniOut.
- B.** Embed the existing OmniScript as a reusable component within a new Community OmniScript.
- C.** Clone the existing OmniScript to a new LWC OmniScript and deploy to the Community portal.
- D.** Duplicate the existing OmniScript and deploy the new script on the website.

Answer: C (LEAVE A REPLY)

The most efficient solution is to clone the existing OmniScript to a new LWC OmniScript and deploy it to the Community portal. LWC OmniScripts are designed for web-based interactions and can be easily embedded in any web page. OmniOut is not a tool for deploying OmniScripts, but a feature that allows users to export data from an OmniScript. Embedding the existing OmniScript as a reusable component within a new Community OmniScript would require creating an unnecessary wrapper script. Duplicating the existing OmniScript would create redundant code and maintenance issues.

NEW QUESTION: 59

Service agents must confirm customer contact information in the first step of a payment OmniScript. Contact information includes name, telephone number, mobile number, and email.

None of the contact information on first step is required. On the last step, after taking payment, the agent can optionally email the receipt to the customer. If the customer says yes, the agent selects a checkbox. If the agent selects the box but the email address field is empty, the process must require the user to return to the first step and enter the customer's email address. What should the consultant recommend to meet this requirement?

- A.** Add a Set Errors element
- B.** Add a DataRaptor to retrieve the email address
- C.** Add a Validation Rule to the Contact object
- D.** Add a Conditional View to the last step

Answer: A (LEAVE A REPLY)

The consultant should recommend adding a Set Errors element to meet this requirement. A Set Errors element can display an error message and prevent the OmniScript from proceeding if certain conditions are not met. The consultant can use a Set Errors element to check if the email address field is empty when the agent selects the checkbox to email the receipt. If it is empty, the Set Errors element can show an error message and direct the user to return to the first step and enter the customer's email address

NEW QUESTION: 60

In which two cases should an integration procedure be used as a data source for FlexCards and OmniScripts, Choose 2 answers

- A.** To achieve elastic scaling
- B.** To retrieve multiple data sources in a single response
- C.** To minimize the number of elements to be configured
- D.** To separate the user interface from changes in the data sources

Answer: B,D (LEAVE A REPLY)

The two cases where an integration procedure should be used as a data source for FlexCards and OmniScripts are: To retrieve multiple data sources in a single response and To separate the user interface from changes in the data sources. An integration procedure is a tool that can execute multiple DataRaptor actions, such as Extract, Transform, and Load, and also invoke REST or SOAP services. The consultant can use an integration procedure as a data source for FlexCards and OmniScripts when they need to retrieve data from different sources, such as Salesforce objects or external systems, and store it in a JSON object for display. This way, the integration procedure can provide multiple data sources in a single response, without requiring multiple elements or actions in the FlexCard or OmniScript. The consultant can also use an integration procedure as a data source for FlexCards and OmniScripts when they want to separate the user interface from changes in the data sources. This way, the integration procedure can handle any changes in the data sources without affecting the FlexCard or OmniScript design or configuration.

NEW QUESTION: 61

A consultant is designing a FlexCard for a client. The client wants to customize the look of each element on the FlexCard. What tool should the consultant use to meet the client's requirements?

- A.** Use DataRaptor to customize the look of each element.
- B.** Use Integration Procedures to customize the look of each element.
- C.** Use the style panel in the FlexCard Designer to customize the look of each element.
- D.** Use OmniScripts to customize the look of each element.

Answer: C (LEAVE A REPLY)

Style Panel Overview: The FlexCard Designer includes a Style Panel (accessed in the properties pane) that allows consultants to customize the visual properties of individual elements (e.g., Text, Image, Button, Datatable) on a FlexCard. This includes settings like:

Font size, color, and weight.

Background color or image.

Borders (style, width, color).

Padding and margins.

Custom CSS for advanced styling.

Meeting the Requirement:

Element-Level Customization: The Style Panel enables granular control over each element. For example, a Text element displaying a customer name can be styled with a bold red font, while a Button can have a blue background and rounded corners.

Ease of Use: It's a no-code solution within the FlexCard Designer, requiring no external tools or coding expertise, aligning with OmniStudio's low-code philosophy.

Preview: Changes made in the Style Panel are instantly previewable in the designer, ensuring the client's vision is met efficiently.

Example: If the FlexCard shows case details, the consultant could use the Style Panel to make the case number bold and blue, add a red border to an "Open" status badge, and adjust the Datatable's row height--all tailored to the client's preferences.

Valid Plat-Con-201 Dumps shared by TrainingQuiz.com for Helping Passing Plat-Con-201 Exam! TrainingQuiz.com now offer the **newest Plat-Con-201 exam dumps**, the TrainingQuiz.com Plat-Con-201 exam **questions have been updated** and **answers have been corrected** get the **newest** TrainingQuiz.com Plat-Con-201 dumps with Test Engine here: <https://www.trainingquiz.com/Plat-Con-201-practice-quiz.html> (222 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 62

An OmniScript steps through a wizard. In Step 2, the user needs to see details of a specific Case selected in Step 1. How should an OmniStudio Consultant fetch the data efficiently?

- A. Use a Data Mapper Load to read the Case data.
- B. Fetch all Cases in the Init action and filter them in Step 2, and use a new Caseld as the input filter.
- C. Use a Data Mapper Extract Action between Step 1 and Step 2, using the Caseld from Step 1 as the input filter.
- D. Ask the user to re-enter the Case details in Step 2.

Answer: C (LEAVE A REPLY)

The efficient pattern is to retrieve only the selected Case when the user reaches Step 2. A Data Mapper Extract Action can use the Caseld selected in Step 1 as an input filter and return only the required Case details. This avoids loading unnecessary records and keeps the OmniScript responsive. A Data Mapper Load is wrong because Loads write data to Salesforce; they do not read Case details. Fetching all Cases in the Init action is inefficient

and can create performance and governor-limit risks. Asking the user to re-enter Case information is poor user experience and creates data-quality problems. Salesforce OmniStudio guidance separates Data Mapper Extract for reading Salesforce data from Data Mapper Load for saving updates.

NEW QUESTION: 63

A Salesforce Omnistudio Consultant is tasked with creating an Omniscript for a client's customer service process. The consultant wants to integrate Salesforce Knowledge into the Omniscript.

What is one way the consultant can configure an Omniscript to search for Salesforce Knowledge articles?

- A.** Using a simple search ..
- B.** Using a complex search algorithm
- C.** Based on dynamic inputs from Omniscript fields
- D.** Based on static values set up during configuration

Answer: (SHOW ANSWER)

The best way to integrate Salesforce Knowledge with an OmniScript is to dynamically retrieve articles based on user inputs.

A DataRaptor Extract or Integration Procedure can fetch Knowledge Articles based on values entered in the OmniScript fields (e.g., case type, keywords).

NEW QUESTION: 64

A consultant needs an OmniScript element to display read-only, formatted information to users without capturing input, such as terms and conditions text. Which element should be used?

- A.** Text Block
- B.** Text
- C.** Radio Button
- D.** HTML Block

Answer: B (LEAVE A REPLY)

The Text element in OmniScript is used to display static or dynamically merged read-only content to the user, such as instructions, disclosures, or terms and conditions.

NEW QUESTION: 65

A company has deployed an OmniScript that is working as designed. During the process, the OmniScript has multiple steps that allow the user to review lists of cases, contacts, quotes, orders, and contracts for an account. Each step uses DataRaptor Extract Actions to retrieve Salesforce data. However, the OmniScript is reaching performance limitations, and the development team warns that they may begin hitting Governor limits. Which two suggestions should the consultant propose to address this concern? Choose 2 answers

- A.** Replace the DataRaptor Extracts with DataRaptor Turbo Extracts

- B. Add Limit filter in DataRaptor Extract
- C. Combine the DataRaptors into an Integration Procedure
- D. Add Order By filter in DataRaptor Extract

Answer: A,C (LEAVE A REPLY)

To improve the performance and avoid hitting Governor limits, the consultant should replace the DataRaptor Extracts with DataRaptor Turbo Extracts and combine them into an Integration Procedure. DataRaptor Turbo Extracts are optimized for speed and efficiency, and can retrieve large amounts of data without using SOQL queries. An Integration Procedure can execute multiple DataRaptor actions in one call, reducing the number of requests and processing time

NEW QUESTION: 66

A business wants to add typeahead functionality to an existing OmniScript. The business wants the user to begin entering the name of a contact in Salesforce. and allow the user to select the contact once it has found a match. Once the contact has been selected, the OmniScript should retrieve the contact's email address for use in a later step of the process. What data source is needed when using the TypeAhead element?

- A. HTTP Action
- B. Integration Procedure Action
- C. DataRaptor Post Action
- D. DataRaptor Extract Action

Answer: D (LEAVE A REPLY)

The data source that is needed when using the TypeAhead element is DataRaptor Extract Action.

A TypeAhead element is an element that can display a list of suggestions for the user to select from, as they type in a text field. A DataRaptor Extract Action is an action that can retrieve data from a Salesforce object or invoke a Salesforce API, using SOQL queries or Apex methods. The consultant can use a DataRaptor Extract Action as a data source for the TypeAhead element, to get the contact names from Salesforce and display them as suggestions

NEW QUESTION: 67

Which element retrieves data from a Salesforce picklist in an org?

- A. DataRaptor Extract Action
- B. Lookup
- C. Select
- D. Calculation Action
- E. Mark this item for later review.

Answer: C (LEAVE A REPLY)

In Salesforce OmniStudio, the Select element within an OmniScript is specifically designed to retrieve and display picklist values from a Salesforce object field, allowing users to

choose from a predefined list of options. This element is used to create a dropdown or picklist interface in an OmniScript, enabling users to interact with Salesforce data by selecting a single value from the available options.

Here's why Select is the correct answer:

- * The Select element in OmniScript supports multiple methods to populate its options, including retrieving values directly from a Salesforce picklist field. According to the official Salesforce OmniStudio documentation, you can configure the Select element's "Option Source" to "SObject," which allows it to fetch picklist values from a specified Salesforce object and field. For example, if you have a picklist field like Industry on the Account object, the Select element can retrieve all active picklist values (e.g., "Technology," "Healthcare," etc.) and present them as a dropdown to the user.
- * The Select element is highly flexible and supports three option source types:
- * Manual: Manually defined label-value pairs.
- * SObject: Retrieves options from a Salesforce object field (such as a picklist).
- * Custom: Uses an Apex controller for more complex logic. When set to "SObject," it directly queries the Salesforce schema to pull the picklist metadata, ensuring the values reflect what's defined in the org.

Now, let's examine why the other options are incorrect:

* A. DataRaptor Extract Action: A DataRaptor Extract is a powerful tool in OmniStudio used to retrieve data from Salesforce objects, transform it, and pass it to an OmniScript or FlexCard. While it can retrieve picklist data as part of a broader dataset (e.g., extracting a record with a picklist field value), it is not an "element" within an OmniScript, nor is it specifically designed to display or interact with picklist values in a user interface. Instead, it operates behind the scenes as a data retrieval mechanism.

The Select element, however, is the UI component that presents those values to the user.

* B. Lookup: The Lookup element in OmniScript is used to search for and select a record from a Salesforce object based on user input (e.g., finding an Account by name). It does not retrieve or display picklist values from a field; it retrieves entire records. While it can indirectly involve picklist fields as part of the record data, its primary purpose is record selection, not picklist value retrieval for display.

* D. Calculation Action: A Calculation Action in OmniScript performs mathematical or logical operations based on user input or data (e.g., adding two numbers or concatenating strings). It does not retrieve data from Salesforce picklist fields or interact with them directly. Its role is computation, not data retrieval or presentation.

The official Salesforce OmniStudio documentation, specifically the "OmniScript Elements" section, highlights the Select element as the appropriate choice for working with picklist fields in a guided process.

For instance, in a scenario where a user needs to choose a Case Reason from a picklist, the Select element fetches the active values (e.g., "Billing Issue," "Technical Support") from the Reason field on the Case object and renders them as a dropdown.

References:

- * Salesforce OmniStudio Documentation: OmniScript Elements Reference - Details the Select element and its SObject option source capabilities.
- * Salesforce OmniStudio Developer Guide: DataRaptors - Explains DataRaptor Extract's role in data retrieval, distinct from UI elements like Select.
- * Salesforce Help: OmniScript Designer - Describes how to configure the Select element to connect to Salesforce picklist fields.

NEW QUESTION: 68

A business process needs to perform a multi-step calculation on each contact record in a list.

Using an Integration Procedure to process the list, what feature can be used to meet this requirement?

- A. Loop Block
- B. Batch Action
- C. Conditional Block
- D. DataRaptor Transform Action

Answer: A (LEAVE A REPLY)

The feature that can be used to meet this requirement in an Integration Procedure is Loop Block.

Loop Block allows the designer to iterate over a collection of data and perform actions on each item. In this case, a Loop Block can be used to loop over each contact record in the list and perform a multi-step calculation using Calculation Actions or Calculation Procedures. Batch Action is not a feature in Integration Procedure, but a type of Action element that can perform bulk operations on data. Conditional Block is used to specify a condition that determines whether the actions inside the block will run or not. DataRaptor Transform Action is used to transform data from one format to another using templates.

NEW QUESTION: 69

A company is designing a new console for contact center agents to display case data for customers. All cases will be fetched using a single DataRaptor. This page needs to display cases as follows:

- * "Open" cases with case description, case open date, case type, assigned to, and priority fields. Open cases should be highlighted with a red border.
- * "Awaiting Closure" cases with case description, last action taken date, resolution, approval reason for closure, and assigned to fields. These cases should be highlighted with a grey border.
- * "Closed" cases with case description, resolution, case closed date fields with a link to duplicate cases.

How should the consultant design the FlexCard solution to meet these requirements?

- A. Using three FlexCards
- B. Using a single FlexCard with three states

C. Using a single FlexCard with multiple card filters

D. Using a single FlexCard with three flyouts

Answer: B (LEAVE A REPLY)

Comprehensive and Detailed In-Depth Explanation: The requirement involves designing a FlexCard solution to display case data fetched by a single DataRaptor, with three distinct categories of cases ("Open," "Awaiting Closure," and "Closed"), each with specific fields and unique styling (e.g., red border for Open, grey border for Awaiting Closure). Based on Salesforce OmniStudio capabilities, the most efficient and appropriate design is a single FlexCard with three states.

Here's why B. Using a single FlexCard with three states is the correct answer:

* **FlexCard States Overview:** In OmniStudio, a FlexCard can have multiple "states," which are conditional views of the card's content and styling based on data conditions. Each state can display different fields, apply unique CSS styles (e.g., borders), and include interactive elements (e.g., links).

States are driven by conditions evaluated against the data retrieved by the DataRaptor, making them ideal for scenarios where the same dataset (in this case, cases from a single DataRaptor) needs to be presented differently based on a field value (e.g., Status).

* **Meeting the Requirements:**

* **Data Source:** All cases are fetched via a single DataRaptor, which aligns with using one FlexCard since FlexCards are designed to work with a single data source (e.g., a DataRaptor Extract) and then manipulate or display that data in various ways.

* **"Open" Cases:** A state can be configured with a condition like Status = 'Open', displaying fields such as CaseDescription, CaseOpenDate, CaseType, AssignedTo, and Priority. Custom CSS can be applied in the state's style settings to add a red border (e.g., border: 2px solid red).

* **"Awaiting Closure" Cases:** Another state with a condition like Status = 'Awaiting Closure' can show CaseDescription, LastActionTakenDate, Resolution, ApprovalReasonForClosure, and AssignedTo, styled with a grey border (e.g., border: 2px solid grey).

* **"Closed" Cases:** A third state with Status = 'Closed' can display CaseDescription, Resolution, and CaseClosedDate, with a hyperlink element added for duplicate cases (e.g., linking to a related DuplicateCaseId field). No specific border color is mentioned, so default styling or a custom style could be applied.

* **Display Logic:** By default, FlexCards in a list format (e.g., a datatable or repeating block) can iterate over the DataRaptor's JSON output, applying the appropriate state to each case record based on its Status. This ensures all cases are displayed dynamically with their respective fields and styles.

* **Why One FlexCard?:** Using a single FlexCard with states keeps the solution efficient, maintainable, and aligned with OmniStudio best practices. It leverages one DataRaptor call, reduces redundancy, and centralizes the logic and styling in a single component.

Now, let's examine why the other options are incorrect:

* A. Using three FlexCards: While it's technically possible to create three separate FlexCards (one for Open, one for Awaiting Closure, and one for Closed), this approach is inefficient and impractical. Each FlexCard would need its own DataRaptor or a filtered subset of the same DataRaptor output, leading to redundant configurations and multiple queries or data manipulations. Additionally, displaying all three categories together in a cohesive console view would require embedding them in an OmniScript or custom layout, complicating the design. The requirement implies a unified display of all cases, which a single FlexCard with states handles more naturally.

* C. Using a single FlexCard with multiple card filters: Card filters in FlexCards allow users to dynamically filter the displayed data (e.g., a dropdown to show only Open cases). However, filters don't inherently change the fields displayed or apply specific styling like red or grey borders per category- they only subset the data. To meet the requirement of showing different fields (e.g., Priority for Open vs. LastActionTakenDate for Awaiting Closure) and unique styling, you'd still need states or custom coding beyond what filters provide. Filters are user-driven, not automatic, and don't fully address the requirement.

* D. Using a single FlexCard with three flyouts: Flyouts are pop-up panels in FlexCards that display additional details or actions when a user interacts with an element (e.g., clicking a case row). While flyouts could show extra case details, they don't suit the requirement of displaying all cases with their respective fields and borders directly on the main card. Flyouts are supplementary, not primary displays, and using three flyouts would require user interaction to view each category, which contradicts the need to "display cases as follows" in a visible, highlighted manner.

Best Practice Alignment:

The OmniStudio documentation emphasizes FlexCard states for scenarios where data from a single source needs conditional rendering and styling. This approach minimizes maintenance overhead, optimizes performance (one DataRaptor call), and provides a seamless user experience for contact center agents viewing all case types in one console.

References:

* Salesforce OmniStudio Documentation: FlexCards Designer - Details how states can conditionally display data and apply custom styles.

* Salesforce OmniStudio Developer Guide: Working with FlexCard States - Explains configuring states based on field values like Status.

* Salesforce Help: FlexCard Styling - Covers applying CSS (e.g., borders) to states for visual differentiation.

NEW QUESTION: 70

A banking client wants to display a list of recent transactions on a customer's profile. The display must be branded to match the bank's website and allow the agent to dispute a transaction directly from the list. What is the primary role of OmniStudio in this scenario?

A. Display interactive data using FlexCards.

B. Present guided steps using OmniScript.

C. Visualize data using Tableau.

D. Query data using Apex.

Answer: A (LEAVE A REPLY)

The primary OmniStudio role here is to display interactive data using FlexCards. The requirement is not just to show static transaction information; the card must present recent transactions in a branded UI and allow the agent to take action, such as disputing a transaction. FlexCards are designed to show contextual information and provide quick actions in a reusable declarative component. OmniScript would be appropriate for a guided multi-step dispute process, but the initial list display and direct action surface are FlexCard responsibilities. Tableau is for analytics and visualization, not transactional service actions. Apex may support custom logic, but it is not the primary declarative OmniStudio UI tool for this use case.

NEW QUESTION: 71

A business is creating an agent console with FlexCards to provide a 360° view of their customers.

The business wants the following information displayed:

- Account information including account name, phone, and website
- Active opportunities related to the account
- Active contracts related to the account
- The ability to view and renew contracts

An Integration Procedure will be used to retrieve Account, Opportunity, and Contract data. How should the consultant design the FlexCards to meet these requirements?

A. Parent FlexCard with multiple Child and Card Actions

B. Parent FlexCard with multiple Child and different Card States

C. Parent FlexCard with single Child and multiple Card States

D. Parent FlexCard with single Child and Card Actions

Answer: A (LEAVE A REPLY)

The consultant should design the FlexCards using a Parent FlexCard with multiple Child and Card Actions to meet these requirements. A Parent FlexCard is a FlexCard that can display data and actions in a card format, and also contain one or more Child FlexCards. The consultant can use a Parent FlexCard to display the account information, including account name, phone, and website. A Child FlexCard is a FlexCard that can display data and actions in a card format within a Parent FlexCard. The consultant can use multiple Child FlexCards to display the active opportunities and contracts related to the account. A Card Action is a button or a link that can invoke an OmniScript or an Integration Procedure from a FlexCard. The consultant can use multiple Card Actions to enable the user to view and renew contracts

NEW QUESTION: 72

A company begins a project to unify its customer data across the enterprise. After completing the discovery and analysis phases of the project, the project team recommends FlexCards as the primary solution. The stakeholder of the project are excited to move forward with this recommendation. However, IT is concerned that creating FlexCards will require custom coding and advanced technical skills they do not have on their team. Which two features should the consultant highlight to address IT's concerns? Choose 2 answers

- A. The FlexCard Wizard
- B. Lightning web components used in FlexCards
- C. Newport Design System used in FlexCards
- D. The FlexCard Designer

Answer: B,D (LEAVE A REPLY)

The two features that the consultant should highlight to address IT's concerns are Custom Styles and FlexCard Designer. Custom Styles are a feature that allows applying custom CSS properties to an OmniScript or a FlexCard element. Custom Styles do not require any coding skills, and can be easily configured using a graphical interface. FlexCard Designer is a tool that allows creating and editing FlexCards using a drag-and-drop interface. FlexCard Designer also does not require any coding skills, and can be used by anyone who has access to OmniStudio tools

NEW QUESTION: 73

An insurance agency wants to enable its call center agents to be more efficient when handling customer inquiries. After analyzing patterns in the call logs, the agency discovers that one of the top customer requests is to find out total insurance premiums paid as well as the difference year over year. Agents need to relay this information to the customer and then send them a summary report by email. The agency decides to implement this process using OmniScript.

What three OmniScripts elements should be used to meet the requirements?

Choose 3 answers

- A. DataRaptor Extract Action
- B. Email Action
- C. DataRaptor Post Action
- D. Messaging Element
- E. Calculation Action

Answer: (SHOW ANSWER)

The three OmniScript elements that should be used to meet the requirements are: DataRaptor Extract Action, Email Action, and Calculation Action. A DataRaptor Extract Action can retrieve data from a Salesforce object, such as total insurance premiums paid and the difference year over year, and store it in an Interface object or a JSON object. An Email Action can send an email to the customer with a summary report attached, using data from an Interface object or a JSON object. A Calculation Action can perform

mathematical operations on data from an Interface object or a JSON object, such as calculating the difference year over year

NEW QUESTION: 74

In OmniScript, how can pages or groups of fields be controlled based on the choices made by the user?

- A.** Add Cancel and Save options for the user
- B.** Create child OmniScripts for branched flow
- C.** Use the Conditional View property on a Step or Block element
- D.** Use the Radio Input property for a Step or Block element

Answer: ([SHOW ANSWER](#))

In OmniScript, pages or groups of fields can be controlled based on the choices made by the user by using the Conditional View property on a Step or Block element. A Conditional View property is a property that determines whether a Step or Block element is displayed or hidden based on certain criteria, such as data values or user roles. The consultant can use the Conditional View property to show or hide different pages or groups of fields in the OmniScript depending on the user's input

NEW QUESTION: 75

Which statement best describes the relationship between OmniScripts and FlexCards?

- A.** FlexCards can launch OmniScripts as actions, and OmniScripts cannot embed FlexCards
- B.** OmniScripts and FlexCards cannot interact with each other
- C.** FlexCards can launch OmniScripts as actions, and OmniScripts can embed FlexCards within Steps
- D.** FlexCards replace OmniScripts entirely in modern implementations

Answer: ([SHOW ANSWER](#))

FlexCards commonly include Action elements that launch OmniScripts (e.g., "File a Claim"), and OmniScripts can embed FlexCards within a Step to display contextual information during a guided process.

NEW QUESTION: 76

A project involves multiple environments, including Development, QA, and Production. What is best practice for managing and moving Data Mapper changes across these environments?

- A.** Deploy components using metadata migration tools and source control.
- B.** Recreate the configurations manually within each environment's designer.
- C.** Treat the components as data records for migration via a data loader tool.
- D.** Extract the JSON definitions for manual import via the browser console.

Answer: **A** ([LEAVE A REPLY](#))

Enterprise OmniStudio implementations should follow established Salesforce application lifecycle management practices. Data Mappers, Integration Procedures, FlexCards, and OmniScripts should be deployed through metadata-based deployment processes, version control systems, and automated pipelines whenever possible. Source control provides traceability, rollback capability, code review support, and environment consistency. Recreating configurations manually introduces human error and deployment drift between environments. Treating OmniStudio assets as ordinary business data is not the intended deployment mechanism and makes change management difficult. Browser- console exports are unsuitable for controlled enterprise deployments. A structured Dev -> QA -> Production promotion process ensures that tested and approved configurations are consistently deployed while supporting governance, auditing, and release-management requirements.

Valid Plat-Con-201 Dumps shared by TrainingQuiz.com for Helping Passing Plat-Con-201 Exam! TrainingQuiz.com now offer the **newest Plat-Con-201 exam dumps**, the TrainingQuiz.com Plat-Con-201 exam **questions have been updated** and **answers have been corrected** get the **newest** TrainingQuiz.com Plat-Con-201 dumps with Test Engine here: <https://www.trainingquiz.com/Plat-Con-201-practice-quiz.html> (222 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 77

A governance board requires that all OmniStudio components be documented to ensure future maintainability. Where is the most effective place to store technical documentation for a specific Integration Procedure?

- A.** In the name of the Integration Procedure.
- B.** OmniStudio components are self-documenting and do not require notes.
- C.** In a separate Word document stored on a shared drive that is rarely updated.
- D.** Within the Description and Internal Notes fields of the Integration Procedure and its elements themselves.

Answer: D (LEAVE A REPLY)

The most effective place for technical documentation is inside the Integration Procedure itself, using the Description and Internal Notes fields on the procedure and its elements. This keeps the explanation directly beside the configuration it describes, which improves maintainability for future consultants, admins, and developers. Storing the documentation only in the component name is too limited and quickly becomes unreadable. Saying OmniStudio components are self- documenting is false; declarative components still need design intent, assumptions, dependencies, and troubleshooting notes. A Word document on a rarely updated shared drive creates documentation drift. Salesforce identifies Internal

Notes as designer-visible descriptive text, making it the proper location for technical implementation notes.

NEW QUESTION: 78

A company plans to use OmniScript to digitally transform its business. During the discovery phase of the project the team reviews all of the business processes including the individual steps of each process. As a result of analysis, it is clear that many processes need to update contact data at different points in the process.

What solution should the consultant recommend to efficiently meet these requirements?

- A.** Create an Integration Procedure to update contact data with "Chain OnStep" enabled and invoke it from all OmniScripts.
- B.** Create an OmniScript that uses the reusable property to update contact data and embed it in other OmniScripts.
- C.** Create an OmniScript to update contact data and invoke it when other OmniScripts complete.
- D.** Create a FlexCard for contacts, and add an Update Contact Details OmniScript as an action.

Answer: [\(SHOW ANSWER\)](#)

The solution that the consultant should recommend to efficiently meet these requirements is to create an OmniScript that uses the reusable property to update contact data and embed it in other OmniScripts. A reusable OmniScript is an OmniScript that can be embedded in another OmniScript as a sub-process. This way, the consultant can create one OmniScript that updates contact data in a consistent way, and then use it in multiple OmniScripts that need to update contact data at different points in the process. This will improve the design and maintenance of the OmniScripts.

NEW QUESTION: 79

A consultant is working on a project that requires the use of Omnistudio tools to automate complex business processes. The consultant is considering using Decision Matrices and Expression Sets.

What is the primary benefit of using Decision Matrices and Expression Sets in Omnistudio?

- A.** They allow for calculations on matrix lookups and variables.
- B.** They facilitate the creation of custom objects.
- C.** They allow for the creation of custom fields.
- D.** They enable the creation of custom code.

Answer: **A** [\(LEAVE A REPLY\)](#)

* Decision Matrices and Expression Sets are used to perform calculations, apply conditional logic, and evaluate business rules in OmniStudio.

* The primary benefit (A) is that they enable calculations on matrix lookups and variables.

* Decision Matrices allow for defining complex rules based on multiple conditions (e.g., determining discount percentages based on customer type).

- * Expression Sets allow for calculations and formulas to process data dynamically.
 - * B. Facilitate the creation of custom objects - Decision Matrices and Expression Sets do not create Salesforce objects; they only manipulate and evaluate data.
 - * C. Allow for the creation of custom fields - Custom fields are created at the Salesforce object level, not via Decision Matrices or Expression Sets.
 - * D. Enable the creation of custom code - OmniStudio is a low-code/no-code platform, meaning these tools are designed to replace the need for custom Apex or JavaScript.
- Comprehensive and Detailed In-Depth Explanation: Why not the other options??
- Salesforce OmniStudio Reference:
- * Decision Matrices Overview
 - * Expression Sets in Calculation Procedures

NEW QUESTION: 80

A company uses calculation procedures to determine product pricing. Due to the company's pricing schedules, there is always more than one calculation procedure active at one time for a given date. In this scenario, how will the calculation engine select which calculation procedure to run?

- A. Date modified
- B. Priority
- C. Sequence
- D. Date created

Answer: (SHOW ANSWER)

The calculation engine will select which calculation procedure to run based on Priority. Priority is a field on the calculation procedure object that determines the order in which calculation procedures are executed when there are multiple active ones for a given date. The lower the priority number, the higher the precedence. Date modified, Sequence, and Date created are not factors that affect the calculation procedure selection logic.

NEW QUESTION: 81

A consultant wants to proceed with a FlexCard canvas design that will display data from a legacy billing system. The legacy billing system is being extended to add a SOAP endpoint that will allow for the data to be retrieved in XML format. However, the endpoint is not ready yet. Which data source should the team use to make progress on the design?

- A. Integration Procedures
- B. DataRaptor
- C. Apex REST
- D. Apex Remote

Answer: A (LEAVE A REPLY)

NEW QUESTION: 82

How many levels deep can reusable OmniScripts be nested?

- A. 2
- B. 5
- C. Unlimited
- D. 1

Answer: D (LEAVE A REPLY)

Reusable OmniScripts are OmniScripts that can be embedded within other OmniScripts to create modular and reusable components. They can be nested up to unlimited levels deep, as long as there are no circular references or conflicts between the parent and child OmniScripts.

NEW QUESTION: 83

A company needs to create a quoting process for its internal agents. During quoting the agent select a product that is passed to the Omniscrypt with product details including the unit price, and specifies the grade (A, B, C, D, E) size (Small, Medium, Large, X-Large). The process should use the grade and size to look up a discount factor, then multiply the unit price by that discount to return the quote.

Which two tools should the to meet these

Choose 2 answers

- A. Data Mapper Transform
- B. Expression Set
- C. Decision Matrix
- D. Data Mapper Extract

Answer: C,D (LEAVE A REPLY)

The requirement involves creating a quoting process within an OmniScript where an agent selects a product (with a unit price), specifies grade and size, looks up a discount factor based on those inputs, and calculates the final quote. Two OmniStudio tools are needed to achieve this: Decision Matrix and DataRaptor Extract.

Here's why C. Decision Matrix and D. DataRaptor Extract are the correct answers:

* C. Decision Matrix:

* Purpose: A Decision Matrix in OmniStudio is a tool used to look up values based on multiple input conditions, returning a result that can be used in calculations or processes. It's essentially a configurable lookup table stored as a Salesforce custom object (DecisionMatrixDefinition).

* Application: In this scenario, the Decision Matrix is ideal for determining the discount factor based on the combination of grade (A, B, C, D, E) and size (Small, Medium, Large, X-Large).

For example:

- * Grade A + Small = 0.9 (10% discount)
- * Grade B + Large = 0.85 (15% discount)
- * And so on for all 20 combinations (5 grades × 4 sizes).

- * **Integration with OmniScript:** The OmniScript can call the Decision Matrix via an Integration Procedure or directly reference it in a Calculation Action, passing Grade and Size as inputs and receiving the DiscountFactor as output.
- * **Why It Fits:** The requirement explicitly states "look up a discount factor," and Decision Matrix is purpose-built for such multi-variable lookups, making it more efficient than manual coding or other tools.
- * **D. DataRaptor Extract:**
 - * **Purpose:** A DataRaptor Extract retrieves data from Salesforce objects and maps it into a JSON format usable by OmniScripts or other components.
 - * **Application:** The product details, including the UnitPrice, are passed to the OmniScript when the agent selects a product. A DataRaptor Extract is needed to fetch this product data from a Salesforce object (e.g., Product2 or a custom object like QuoteLineItem__c) based on the selected product's ID. The extracted data (e.g., { "ProductId": "01t...", "UnitPrice": 100 }) is then available in the OmniScript's data JSON.
 - * **Role in Calculation:** After the Decision Matrix provides the discount factor, the OmniScript can use a Calculation Action to multiply the UnitPrice (from the DataRaptor Extract) by the DiscountFactor (from the Decision Matrix) to compute the quote (e.g., $100 * 0.9 = 90$).
 - * **Why It Fits:** The process starts with product selection, and DataRaptor Extract is the standard OmniStudio tool for retrieving Salesforce data like unit price.
 - * **How They Work Together:**
 - * **DataRaptor Extract:** Fetches product details (e.g., UnitPrice) when the agent selects a product.
 - * **Decision Matrix:** Looks up the discount factor based on Grade and Size inputs.
 - * **Calculation Action in OmniScript:** Multiplies UnitPrice by DiscountFactor to return the quote.

Now, let's examine why the other options are incorrect:

- * **A. DataRaptor Transform:** A DataRaptor Transform manipulates or reshapes data (e.g., converting JSON structures or applying formulas) but doesn't retrieve data from Salesforce or perform lookups like a Decision Matrix. While it could theoretically calculate the quote after data is fetched, it's not needed here since a Calculation Action within OmniScript can handle the multiplication, and it doesn't address the lookup requirement.
- * **B. Expression Set:** An Expression Set defines reusable formulas or conditions in OmniStudio, often used in Integration Procedures or Calculations. While it could compute the final quote (e.g., $\text{UnitPrice} * \text{DiscountFactor}$), it doesn't retrieve data or perform lookups based on grade and size. It's a supporting tool, not a primary solution for this requirement.

References:

- * **Salesforce OmniStudio Documentation: Decision Matrix Overview** - Describes using Decision Matrices for multi-condition lookups.
- * **Salesforce OmniStudio Developer Guide: DataRaptor Extract** - Details retrieving Salesforce data for OmniScript use.

NEW QUESTION: 84

What is the purpose of the "OmniStudio User" permission set license (PSL)?

- A. It provides baseline entitlement to use OmniStudio functionality.
- B. It automatically provisions OmniStudio tools for all users.
- C. It grants direct object-level permissions within OmniStudio components.
- D. It enables assignment of OmniStudio-specific permission sets to a user.

Answer: D ([LEAVE A REPLY](#))

The OmniStudio User permission set license is used as the licensing entitlement that allows OmniStudio-related permission sets to be assigned to a user. A PSL does not automatically provision all OmniStudio tools, and it does not itself replace proper permission set assignments, object permissions, or field-level security. The actual access model still requires assigning the relevant OmniStudio permission sets according to the user's role. Option A is close but too broad; the more precise purpose in this context is that the PSL enables the assignment and use of OmniStudio-specific permission sets. Salesforce documentation and release notes describe OmniStudio permission set licenses and the relationship between OmniStudio User/Admin licensing and permission set assignment.

NEW QUESTION: 85

A public sector agency wants to digitize a paper-based permit application. The process requires a citizen to step through several screens of questions, upload a PDF document, and sign the application digitally. Which OmniStudio tool serves this specific use case?

- A. Calculation Matrices
- B. OmniScript
- C. Integration Procedures
- D. FlexCards

Answer: ([SHOW ANSWER](#)**)**

OmniScript is the correct tool because the requirement is a guided, step-by-step digital interaction. The citizen must move through several screens, answer questions, upload supporting documentation, and complete an application process. Calculation Matrices are for rules-based calculations, not user-facing permit forms. Integration Procedures handle server-side orchestration and integrations, but they do not provide the guided UI experience. FlexCards are used to display contextual information and actions, not to run a multi-screen application journey.

Salesforce specifically positions OmniScripts as guided processes for customers, partners, and internal users, and Salesforce Public Sector documentation describes OmniScripts as tools for dynamic application forms for licenses, permits, programs, and services. Salesforce also supports file upload elements within OmniScripts.

NEW QUESTION: 86

A telecommunications company wants to create a 360° view of their customers, including all customer install products grouped by service address. During the discovery phase of the project, the consultant identifies that installed products can have 4 different statuses. Agents complete different processes, depending on status of installed product. The consultant creates an inventory of all statuses and processes as shown below.

Status = Active

Pay bill

Review usage

Cancel service

Status = Suspended

Pay bill

Reconnect service

Cancel service

Status = Pending installation

Set up billing

Schedule installation

Cancel installation

Status = Canceled

Reconnect service o Cancel service

Status = Pending installation

Set up billing

Schedule installation

Cancel installation

Status = Canceled

Pay bill

Disconnect service

Reconnect service

Which two FlexCard features should the consultant use to meet these requirements?

Choose 2 answers

A. States

B. OmniStudio Action

C. Custom Style

D. Flyout

Answer: A,B (LEAVE A REPLY)

The two FlexCard features that the consultant should use to meet these requirements are States and OmniStudio Action. A State is a condition that determines how a FlexCard displays data and actions based on certain criteria. The consultant can use States to filter the installed products by their status and display different fields and actions accordingly. An OmniStudio Action is a button that can invoke an OmniScript or an Integration Procedure from a FlexCard. The consultant can use OmniStudio Actions to enable the

agents to complete different processes for each installed product, such as pay bill, cancel service, etc

NEW QUESTION: 87

Which element retrieves data from a Salesforce picklist in an org?

- A.** DataRaptor Extract Action
- B.** Lookup
- C.** Select
- D.** Calculation Action
- E.** Mark this item for later review.

Answer: [\(SHOW ANSWER\)](#)

The Select element in OmniScript supports multiple methods to populate its options, including retrieving values directly from a Salesforce picklist field. According to the official Salesforce OmniStudio documentation, you can configure the Select element's "Option Source" to "SObject," which allows it to fetch picklist values from a specified Salesforce object and field. For example, if you have a picklist field like Industry on the Account object, the Select element can retrieve all active picklist values (e.g., "Technology," "Healthcare," etc.) and present them as a dropdown to the user.

The Select element is highly flexible and supports three option source types:

Manual: Manually defined label-value pairs.

SObject: Retrieves options from a Salesforce object field (such as a picklist).

Custom: Uses an Apex controller for more complex logic. When set to "SObject," it directly queries the Salesforce schema to pull the picklist metadata, ensuring the values reflect what's defined in the org.

NEW QUESTION: 88

How should the consultant design the OmniScript solution to allow the user to stop and resume a process at a later time?

- A.** Configure the Save property
- B.** Configure a ResumeStep
- C.** Use an Integration Procedure
- D.** Use a DataRaptor Post Action

Answer: **A** [\(LEAVE A REPLY\)](#)

The consultant should design the OmniScript solution using the Save property to allow the user to stop and resume a process at a later time. The Save property is a property that determines whether an OmniScript can be saved as a draft and resumed later from where it was left off. The consultant can enable the Save property on an OmniScript to allow the user to stop and resume a process at a later time, without losing any data or progress.

NEW QUESTION: 89

A consultant is designing a Flexcard for a client. The client wants the Flexcard to launch a child card with additional information and actions related to the parent card when the agent clicks on a button. How should the consultant design the Flexcard to meet the client's requirements?

- A.** Design the parent card to launch a child card when the specific action is taken.
- B.** Design the parent card to launch a child card through a Flyout Action.
- C.** Design the parent card to launch a new Data Mapper when the specific action is taken.
- D.** Design the parent card to launch a new Omniscrypt when the specific action is taken,

Answer: [\(SHOW ANSWER\)](#)

Flyout Action Overview: A Flyout Action in FlexCards is an interactive feature that displays a pop- up panel (flyout) when triggered, typically by a button or link. This flyout can contain a child FlexCard, OmniScript, or custom content, showing additional details or enabling actions related to the parent card's data.

NEW QUESTION: 90

A business wants to create a reusable OmniScript to capture customer payment information during the order process. The business decides that the first step of the payment process should include:

- Payment type (credit card or bank account)
- Payment amount

Which two elements should the consultant recommend for this step of the process?

Choose 2 answers

- A.** Number
- B.** Radio
- C.** Multi-select
- D.** Currency

Answer: B,D [\(LEAVE A REPLY\)](#)

The two elements that the consultant should recommend for this step of the process are: Radio and Currency. A Radio element can display a list of options for the user to choose from, such as payment type. A Radio element allows only one option to be selected at a time, and can also have icons for each option. A Currency element can display a field for the user to enter a currency value, such as payment amount. A Currency element can also have validation and formatting options, such as minimum and maximum values, decimal places, etc

NEW QUESTION: 91

An OmniStudio Consultant needs to link a new Case to an existing Contact, but the input data only contains the Contact's Name, not the Salesforce ID. How should the consultant use the Data Mapper Load to resolve this Salesforce ID mapping?

- A.** Map the Contact Name to the Case ContactId field in the Mapping tab.
- B.** Enable the Auto-Link property within the Data Mapper Options tab.

- C. Configure a Formula to convert the Name string into a Record ID.
- D. Define a Lookup mapping on the Contact object using the Name field.

Answer: D (LEAVE A REPLY)

The Data Mapper Load must resolve the related Contact record before it can populate the Case ContactId lookup field. Mapping the Contact Name directly into ContactId is invalid because ContactId expects a Salesforce record ID, not a text name. A formula cannot reliably convert a name string into a Salesforce ID. "Auto-Link" is not the correct Data Mapper Load mechanism.

The proper approach is to define a lookup mapping that searches the Contact object using the available field value, then uses the resolved Contact record ID for the Case relationship.

Salesforce Data Mapper Load documentation confirms that Loads write data to Salesforce objects, and related-object/link mapping patterns are used to populate lookup relationships between records. In a real project, a unique field is safer than Name because names may not be unique.

Valid Plat-Con-201 Dumps shared by TrainingQuiz.com for Helping Passing Plat-Con-201 Exam! TrainingQuiz.com now offer the **newest Plat-Con-201 exam dumps**, the TrainingQuiz.com Plat-Con-201 exam **questions have been updated** and **answers have been corrected** get the **newest** TrainingQuiz.com Plat-Con-201 dumps with Test Engine here: <https://www.trainingquiz.com/Plat-Con-201-practice-quiz.html> (222 Q&As Dumps, **40%OFF Special Discount: Exam-Tests**)

NEW QUESTION: 92

Which OmniStudio tool would a consultant use to perform multi-step numeric calculations, such as computing an insurance premium based on several weighted factors and lookup values?

- A. DataRaptor Extract
- B. Calculation Procedure
- C. FlexCard
- D. Type Ahead Block

Answer: B (LEAVE A REPLY)

Calculation Procedures (often paired with Calculation Matrices for lookup-based values) are designed specifically for performing complex, multi-step numeric calculations such as premium rating.

NEW QUESTION: 93

A pop-up window will display additional information about the campaign, such as details about the members.

A button must initiate a guided process to add new leads to the campaign.

Which two FlexCard features should the consultant recommend to meet these requirements?

Choose 2 answers

- A. Block Element
- B. OmniScript Action
- C. Interaction Launcher
- D. Data Table
- E. Flyout Action

Answer: B,E (LEAVE A REPLY)

* Flyout Action (E) - The Flyout Action in a FlexCard is used to display additional information in a pop-up window. Since the requirement mentions that a pop-up window must display additional details, the Flyout Action is the best choice for this functionality.

* OmniScript Action (B) - The OmniScript Action launches an OmniScript, which is a guided interaction tool for multi-step processes. Since the requirement states that a button should initiate a guided process to add new leads, an OmniScript is the ideal choice.

* A. Block Element - Blocks in FlexCards are used for organizing layout and content but do not provide pop-ups or interactive actions.

* C. Interaction Launcher - This launches an interaction but is typically used for navigating to another page or launching an external URL, not a pop-up with additional details.

* D. Data Table - A Data Table is used to display multiple records in a table format but does not trigger a guided process or pop-ups.

Comprehensive and Detailed In-Depth Explanation: Why not the other options??

Salesforce OmniStudio Reference:

- * FlexCards Actions Overview
- * Using Flyout Actions in FlexCards
- * Launching OmniScripts from FlexCards

NEW QUESTION: 94

A company wants to create a new digital interaction process that allows customers to request a quote for service from a local retail energy supplier. The process requires the following actions:

- * Allow the user to select one or more energy products from a list
- * Get current energy usage data from an external system via an API
- * Save the data back to Salesforce as a lead

Which three OmniScript elements should the consultant recommend to meet these requirements?

Choose 3 answers

- A. DataRaptor Post Action
- B. Multi-select Input
- C. Post to Object Action

- D. Radio Input
- E. HTTP Action

Answer: A,B,E (LEAVE A REPLY)

The three OmniScript elements that the consultant should recommend to meet these requirements are:

DataRaptor Post Action, Multi-select Input, and HTTP Action. A DataRaptor Post Action is an action that can write data to a Salesforce object or invoke a Salesforce API. The consultant can use a DataRaptor Post Action to save the data back to Salesforce as a lead. A Multi-select Input is an element that can display a list of options for the user to choose from, such as energy products. A Multi-select Input allows multiple options to be selected at once, and can also have icons for each option. An HTTP Action is an element that can invoke a REST or SOAP service to retrieve data from an external system, such as current energy usage data. An HTTP Action can store the response data in a JSON object for further processing

NEW QUESTION: 95

A Consultant is working on a project that involves using OmniStudio tools to design solutions that meet customer business requirements. The solutions need to be maintainable, scalable, and contribute to long-term customer success. Which OmniStudio tool should the Consultant use to display data and launch actions?

- A. OmniScripts
- B. FlexCards
- C. Integration Procedures
- D. DataRaptors

Answer: B (LEAVE A REPLY)

FlexCards are designed to display data from Salesforce or external sources and launch actions such as OmniScripts, Integration Procedures, or other system interactions. FlexCards provide a reusable, scalable UI component that aligns with the requirements of maintainability and long-term success.

NEW QUESTION: 96

A company needs a guided process for their internal support department that will provide the following functionality:

- Allow the user to enter their employee id
- Retrieve name, mobile phone, title, and email using the employee id
- Display the user's details on a page
- On another page, allow the user to enter case details, including subject, category, severity, and description

Which three elements should the consultant recommend to meet these requirements using an OmniScript? Choose 3 answers

- A. Messaging

- B. DataRaptor Post Action
- C. Calculation Action
- D. DataRaptor Extract Action
- E. Steps

Answer: ([SHOW ANSWER](#))

The three elements that the consultant should recommend to meet these requirements using an OmniScript are: Messaging, DataRaptor Post Action, and Steps. A Messaging element is an element that can display a text message with an icon and a color, such as blue for information or yellow for warning. The consultant can use a Messaging element to display the user's details on a page. A DataRaptor Post Action is an action that can write data to a Salesforce object or invoke a Salesforce API. The consultant can use a DataRaptor Post Action to save the case details to Salesforce. A Step element is an element that can group related fields and elements in an OmniScript, and display them in a page or a section. The consultant can use multiple Step elements to organize the OmniScript into logical steps, such as entering employee id, displaying user details, and entering case details.

NEW QUESTION: 97

A business wants to add typeahead functionality to an existing OmniScript. The business wants the user to begin entering the name of a contact in Salesforce. and allow the user to select the contact once it has found a match. Once the contact has been selected, the OmniScript should retrieve the contact's email address for use in a later step of the process.

What data source is needed when using the TypeAhead element?

- A. HTTP Action
- B. Integration Procedure Action
- C. DataRaptor Post Action
- D. DataRaptor Extract Action

Answer: ([SHOW ANSWER](#))

The data source that is needed when using the TypeAhead element is DataRaptor Extract Action. A TypeAhead element is an element that can display a list of suggestions for the user to select from, as they type in a text field. A DataRaptor Extract Action is an action that can retrieve data from a Salesforce object or invoke a Salesforce API, using SOQL queries or Apex methods. The consultant can use a DataRaptor Extract Action as a data source for the TypeAhead element, to get the contact names from Salesforce and display them as suggestions

NEW QUESTION: 98

A company is designing a new console for contact center agents. The cards in the console need to display the following:

- "Ope"; cases with case description, case open date, case type,

assigned to and priority fields. Open should be highlighted with a red border.

- "Awaiting Closure" cases with case description, last action taken date, resolution, approval reason for closure, and assigned to fields.

These cases should be highlighted with a grey border.

- "Close"; cases with case description, resolution, case closed date fields with a link to duplicate cases.

All cases will be fetched using a single DataRaptor.

How should the consultant design the FlexCard solution to meet these requirements?

A. Using card session variables and a single FlexCard with multiple flyouts

B. Using card session variables and multiple FlexCards

C. Using card filter and a single FlexCard with multiple flyouts

D. Using card filter and multiple FlexCards

Answer: ([SHOW ANSWER](#))

The consultant should design the FlexCard solution using card filter and multiple FlexCards to meet these requirements. A card filter is a property that determines which records are displayed in a FlexCard based on certain criteria. The consultant can use card filter to display only the cases with a specific status, such as Open, Awaiting Closure, or Closed. A FlexCard is a tool that can display data and actions in a card format. The consultant can create multiple FlexCards for each case status, and customize the fields and actions accordingly. The consultant can also use custom style to apply different borders for each case status

NEW QUESTION: 99

A consultant needs to ensure an OmniScript does not allow users to proceed to the next step until all required fields are completed and valid. Which built-in OmniScript capability supports this?

A. Custom Apex validation only

B. Field-level "Required" property combined with OmniScript's native validation on Next/Submit actions

C. Manual JavaScript validation is always required

D. This is not supported natively and requires Flow

Answer: ([SHOW ANSWER](#))

OmniScript natively supports marking fields as Required and validates them automatically when the user attempts to navigate to the next step or submit, without requiring custom code for basic validation.

NEW QUESTION: 100

A company has designed a process that extracts a large amount of data from a Salesforce object. Due to the volume of data that will be retrieved, it is important that Governor limits are not exceeded.

What solution should the consultant recommend?

- A. DataRaptor Extract with Chaining
- B. DataRaptor Turbo Extract
- C. Calculation Procedure and DataRaptor Turbo Extract
- D. Integration Procedure and DataRaptor Turbo Extract

Answer: ([SHOW ANSWER](#))

The solution that the consultant should recommend is DataRaptor Turbo Extract.

DataRaptor Turbo Extract allows the designer to retrieve a large amount of data from a Salesforce object in batch mode without hitting governor limits. DataRaptor Extract is not suitable for this scenario, as it can only retrieve data from a single record or a small set of records. Calculation Procedure and DataRaptor Turbo Extract are not needed for this scenario, as there is no calculation involved. Integration Procedure and DataRaptor Turbo Extract are not needed for this scenario, as there is no data integration involved.

Valid Plat-Con-201 Dumps shared by TrainingQuiz.com for Helping Passing Plat-Con-201 Exam! TrainingQuiz.com now offer the **newest Plat-Con-201 exam dumps**, the TrainingQuiz.com Plat-Con-201 exam **questions have been updated** and **answers have been corrected** get the **newest** TrainingQuiz.com Plat-Con-201 dumps with Test Engine here: <https://www.trainingquiz.com/Plat-Con-201-practice-quiz.html> (222 Q&As Dumps, **40%OFF** Special Discount: **Exam-Tests**)