PDFVCE

Choose the version that fits your needs	PDF Version	Desktop Test Engine	Online Test Engine
Latest and Up-to-Date exam dumps with real exam questions answers.	✓	✓	✓
Get 12-Months free updates without any extra charges.	✓	✓	✓
Experience same exam environment before appearing in the certification exam.	✗	✓	✓
100% exam passing guarante in the first attempt.	✓	✓	✓
20% discount on more than one license and 30% discount on 5+ license purchases.	✗	✓	✓
100% secure purchase on SSL.	✓	✓	✓
Completely private purchase without sharing your personal info with anyone.	✓	✓	✓

Exam : **ACD201**

Title : Appian Senior Developer

Vendor : Appian

Version : DEMO

NO.1 You need to test a related action that updates record data.
Appropriate users must be able to access the form. You also need to ensure that the data is successfully updated.
Using Appian, which is the best testing method you should use?
A. User Accepting Testing
B. Unit Testing
C. Performance Testing
Answer: A
Explanation:
User Acceptance Testing (UAT) is the best method, as it ensures appropriate users can access the form and verifies that the data update process works as intended in a real-world scenario.

NO.2 You're creating an interface object that displays a report of open cases. This interface will be used as the content of a site page.
Users in the "ACME Employees" group must be able to view the report.
Which two steps should you perform? (Choose two.)
A. Ensure that the site page visibility permits the "ACME Employees" group to view the page.
B. Add the "ACME Employees" group as a viewer on the site object.
C. Include the "ACME Employees" group as a member of the "Report Viewers" system group.
D. Add the "ACME Employees" group as a viewer on the folder containing the interface object.
Answer: A,D
Explanation:
The site page visibility must permit the "ACME Employees" group to ensure they can access the site page with the report.
The "ACME Employees" group must be added as a viewer on the folder containing the interface object to grant them access to the report interface.

NO.3 You need to connect to an external system using OAuth 2.0: SAML Bearer Assertion Flow authentication type, which requests access to an API on behalf of a signed in user.
This standard has the several steps involved with the SAML Bearer Assertion Flow.
Which two steps should you perform? (Choose two.)
A. Add required users and groups to OAuth 2.0 SAML Bearer Assertion Users system group.
B. Enable the checkbox labeled OAuth 2.0: SAML Bearer Assertion Flow in the Admin Console.
C. Create a Bearer Assertion process flow to authenticate the user.
D. Upload Client Certificate to Connected System.
Answer: A,D
Explanation:
Adding required users and groups to the OAuth 2.0 SAML Bearer Assertion Users system group ensures only authorized users can initiate the authentication flow.
Uploading a client certificate to the Connected System is needed for the assertion and secure communication required by the SAML Bearer Assertion Flow.

NO.4 The synced record Task has a self-referential relationship defined in the column parentTaskId.
There is a many-to-one record relationship between the id and parentTaskId called parentTask.
For a given task ID, you need to return the task name and the parent task name.

What should you do?

A. Create a sync-time custom record field on the Task record called parentName. Specify this field to return in the query field selection.

B. Use a!queryRecordType() with a filter on the task id, with fields specified to return recordType!Task.name and recordType!Task.parentTask.name.

C. Use a!queryRecordType() filtered on the task id once to return the task name and parent task ID. Query the record again to return the parent task name.

Answer: B

Explanation:

Using a!queryRecordType() with a filter on the task ID and specifying both recordType!Task.name and recordType!Task.parentTask.name in the fields will efficiently return both the task name and its parent task name in a single query.

NO.5 You're designing a custom component rule!pagingComponent() to page through a list of items. The component must display the current page numbers being shown, as well as the total number of items. Users should be able to click arrows to increase or decrease the pagination.

Review the following:

```
a!localVariables(
local!query: rule!getData(),
{
rule!interface(),
rule!pagingComponent()
}
)
```

Which two patterns should be used when calling the component in this scenario? (Choose two.)

A. Passing into the rule input totalCount of rule!pagingComponent() the value: local!query.totalCount where local!query is returning a dataSubset and returnTotalCount = "True".

B. A rule input ri!pagingInfo on rule!pagingComponent() with a local variable of the same name in the main interface with an initial value. When the arrows are clicked, this updates the value of the local variable in the main interface.

C. A local variable local!totalCount calculated using count(local!query), which is passed into rule!pagingComponent() via a rule input. The data type of local!query is a list of record or Custom Data Type (CDT).

D. A local variable local!pagingInfo inside rule!pagingComponent() with an initial value. When the arrows are clicked, this updates the value of the local variable.

Answer: A,B

Explanation:

Passing local!query.totalCount (from a dataSubset with returnTotalCount: true) into a rule input allows rule!pagingComponent() to accurately display total items for pagination.

Managing paging state (such as ri!pagingInfo) as a local variable in the main interface and updating it when arrows are clicked ensures that pagination is controlled centrally, maintaining state across the UI and allowing the component to function as intended.

NO.6 You need to create an expression rule that will be reused throughout your environment.

What are two reasons why you should include meaningful test cases when creating a new expression rule? (Choose two.)

A. To improve the performance of the environment.

B. To accelerate various types of testing, including: unit, regression, exploratory.

C. To facilitate Test-Driven Development.

D. To enhance the appearance of the code.

Answer: B,C

Explanation:

Meaningful test cases accelerate different testing types by providing clear validation scenarios for the rule.

They support Test-Driven Development by enabling you to define expected outcomes before or during rule creation.

NO.7 You're refining a story for a centralized distribution center that extends the current application by allowing inventory managers to review orders.

These orders are created through a record action that various workshops access from a shared Appian site.

Which consideration best suits this use case?

A. How will you handle errors resulting from a failed integration as a result of the workshops using an external system to place orders?

B. What are the personas that will be utilizing this application so that an Appian group structure can be developed and maintained?

C. What components have already been built relating to this piece of functionality so that you can reuse them and speed up development?

Answer: C

Explanation:

Identifying the personas (such as inventory managers and workshop users) ensures you can design and maintain an appropriate Appian group structure for managing access and permissions, which is critical for a centralized application accessed by multiple roles.

NO.8 Your client wants to configure user authentication using single sign-on (SSO) in their cloud environment.

Which two authentication types should you use to support this requirement? (Choose two.)

A. PIEE user authentication

B. LDAP authentication

C. OpenID Connect authentication

D. Appian authentication

Answer: C,D

Explanation:

OpenID Connect authentication supports single sign-on (SSO) by integrating with external identity providers.

Appian authentication can work in conjunction with SSO configurations as part of a hybrid setup or fallback mechanism.

NO.9 A recent Health Check report of your application indicates an increased risk of large context

size in the interfaces.
What are two possible contributors to this risk? (Choose two.)
A. Data transformation is performed inside the interface rather than delegated to the source system.
B. Multiple local variables hold duplicate data.
C. The load variable has a web service response as JSON.
D. Local variables are set to refresh every five minutes.
Answer: A,B
Explanation:
Performing data transformation inside the interface can increase the amount and size of context data stored, raising context size risk.
Multiple local variables holding duplicate data unnecessarily inflate the interface's memory footprint, contributing to large context size.

NO.10 You need to implement a requirement where a third-party system starts a process in Appian. The third-party system can invoke a service only through Web Services Description Language (WSDL).
What should you do to start the process in Appian?
A. Create a default WSDL URL using process model UUID.
B. Create a custom plug-in.
C. Expose process model as a web service.
Answer: C

NO.11 You're designing a file upload process that integrates with an external document management system.
Part of this process is to develop a start form with a file upload component, so that the user can upload multiple files.
The business has requested that the maximum file size that can be uploaded through this process is 1GB. Additionally, you're aware that the integration's API can accept files with a maximum file size of 3GB, but recommends no larger than 1GB.
Can the process be more performant without straining the application or integration?
A. Yes. Request the business allows for these files to be uploaded as an overnight process by storing large files in a staging table, so that adequate resources are available.
B. No. The file upload component is capable of accepting 1 GB files and the integration can handle this.
C. Yes. Request the business reconsiders its requirements and find manual workarounds as it is not recommended to upload files of this size in both Appian and the integration.
Answer: A
Explanation:
While Appian and the integration technically support 1 GB file uploads, handling such large files during interactive sessions can negatively impact application performance, user experience, and resource utilization.
A better approach is to request that the business allow these files to be uploaded asynchronously (e.g., overnight batch process), by temporarily storing them (e.g., in a staging table or folder) and processing them when system load is lower. This makes the process more performant and reliable without straining Appian or the external system.