ACD200 Q&As

Appian Certified Senior Developer

Pass Appian ACD200 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/acd200.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Appian Official Exam Center

QUESTION 1

You create an interface, but it fails to load. When you open the design errors log, you see a memory circuit breaker error.

What are two possible root causes of this error? (Choose two.)

A. A database query is taking too long to evaluate.

B. The interface contains some special characters.

C. The interface is storing too much data in local variables.

D. The interface component is looping over too many items.

Correct Answer: CD

A memory circuit breaker error occurs when an interface consumes too much memory on the server. This can happen when the interface is storing too much data in local variables or when the interface component is looping over too many items. Both of these scenarios can cause excessive memory allocation and garbage collection, which can degrade the performance of the interface and the server. Therefore, the possible root causes of this error are C and D. References: Memory Circuit Breaker Performance Best Practices

QUESTION 2

You need to create a plug-in to perform a job in the background. The plug-in should not be available under an expression rule, connected system, or the process model.

What type of plug-in should you create? (Choose the best answer.)

A. Servlet

B. Function

C. Connected systems

D. Smart service

Correct Answer: A

The type of plug-in that should be created to perform a job in the background and not be available under an expression rule, connected system, or the process model is a servlet plug-in. A servlet plug-in is a plug-in that allows you to create custom servlets that can be accessed through a URL. Servlets are Java classes that run on a web server and handle requests and responses. Servlet plug-ins can be used to perform background tasks, such as sending notifications, logging events, or integrating with external systems, without exposing them to the end users or developers. References: Servlet Plug-ins, Appian Suite Plug-ins

QUESTION 3

Your organization is considering the adoption of Behavior-Driven Development (BDD) and automated testing as part of application development.

Which three testing tools have packages preconfigured to work with Appian applications and are available via the AppMarket? (Choose three.)

A. Cucumber

B. Jenkins

C. Selenium API

D. Git

E. FitNesse

Correct Answer: ACE

The question is about the testing tools that have packages preconfigured to work with Appian applications and are available via the AppMarket. The following are three testing tools that meet these criteria:

Cucumber: This is a tool that supports Behavior-Driven Development (BDD) and automated testing by allowing users to write test scenarios in plain language and execute them using step definitions. Cucumber has a package called Appian

Cucumber Test Framework that provides a set of predefined step definitions and helper methods for testing Appian applications.

Selenium API: This is a tool that allows users to automate web browser interactions and test web applications. Selenium API has a package called Appian Selenium Framework that provides a Java library and a set of helper methods for

testing Appian applications.

FitNesse: This is a tool that allows users to create and run acceptance tests using wiki pages and tables. FitNesse has a package called Appian FitNesse Testing Framework that provides a set of fixtures and helper methods for testing

Appian applications.

The following are not testing tools that have packages preconfigured to work with Appian applications and are available via the AppMarket:

Jenkins: This is a tool that allows users to automate software development tasks such as building, testing, and deploying applications. Jenkins does not have a package specifically for Appian applications, but it can be integrated with other

testing tools such as Cucumber or Selenium API.

Git: This is a tool that allows users to manage version control and collaboration for software projects. Git does not have a package specifically for Appian applications, but it can be used to store and share test scripts or scenarios created by other testing tools. References: AppMarket Cucumber Selenium API FitNesse

QUESTION 4

You need to show joined data from 5 tables. Each table contains a large number of rows and could generate a large result set after executing the Joins.

The business is not expecting live data, and a 2-hour refresh is acceptable. Performance is a top priority.

What should you use? (Choose the best answer.)

A. Table

B. View

C. Stored procedure

D. Materialized view

Correct Answer: D

A materialized view is the best option to show joined data from 5 tables that contain a large number of rows and could generate a large result set after executing the joins. A materialized view is a physical table that holds the results of the SQL that a view would normally be constructed from and can be generated periodically. A materialized view can improve performance by reducing the execution time of complex queries that involve multiple joins, aggregations, or calculations. A materialized view can also reduce the load on the database server by storing the query results in advance. A materialized view can be refreshed at regular intervals or on demand to reflect the changes in the underlying tables. References: [Materialized Views], [View Performance]

QUESTION 5

Your organization is considering automating the running of expression rule test cases to provide unit tests for your Appian applications.

Which three methods could be used to launch a test run when required? (Choose three.)

A. Via the DevOps section of the Administration Console.

B. A process model invoked via an API.

C. A process model exposed to users as an action.

D. A web hook from a content versioning system (CVS).

E. A SAIL interface embedded in a report.

Correct Answer: BCE

Three methods that could be used to launch a test run for expression rule test cases when required are: A process model invoked via an API. A process model can be designed to run test cases for expression rules using the a!testRule() function or the Test Rule smart service. This process model can be exposed as a web API with an HTTP method such as POST or PUT, allowing external systems or applications to invoke it through an HTTP request. A process model exposed to users as an action. A process model can also be designed to run test cases for expression rules using the same function or smart service as above. This process model can be exposed to users as an action on an interface, such as a button or a link, allowing users to trigger it manually when needed. A SAIL interface embedded in a report. A SAIL interface can be created to run test cases for expression rules using the a!testRule() function. This interface can be embedded in a report, such as a grid or a chart, allowing users to view the test results interactively on an interface. References: Automated Testing for Expression Rules, a!testRule() Function, Test Rule Smart Service, Web APIs, SAIL Interfaces