DUMPSBOSS.

Lead System Architect (LSA) Pega Architecture Exam 86V2

Pegasystems PEGAPCLSA86V2

Version Demo

Total Demo Questions: 20

Total Premium Questions: 200

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

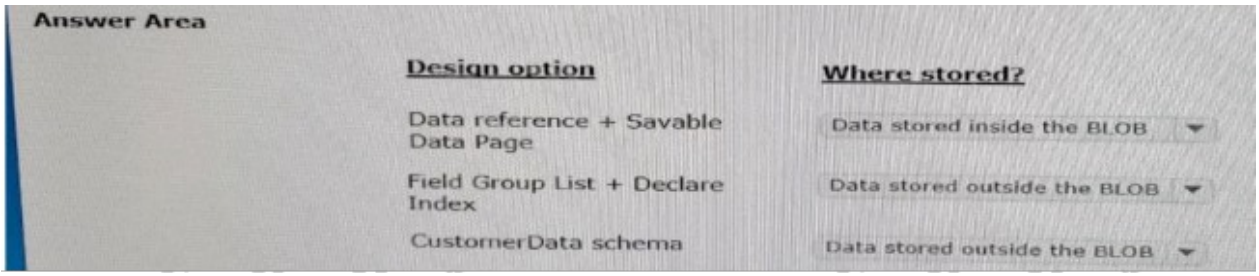
support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Application Design	65
Topic 2, Data Modeling	18
Topic 3, Data Integration	18
Topic 4, Security	37
Topic 5, Performance	48
Topic 6, Deployment	14
Total	200

QUESTION NO: 1 - (SIMULATION)

In the Answer Area, determine where data is stored for each design option.



ANSWER: See the explanation for the answer

Explanation:

Set the storage locations as follows:

Data Reference + Savable Data Page — Data stored inside the BLOB
Field Group List + Declare Index — Data stored inside the BLOB
CustomerData schema — Data stored outside the BLOB

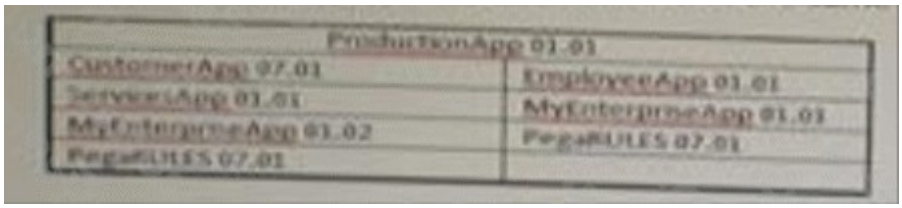
A CustomerData schema uses a separate database schema, whereas the first two design options retain the case data in the Pega instance BLOB.

QUESTION NO: 2 - (DRAG DROP)

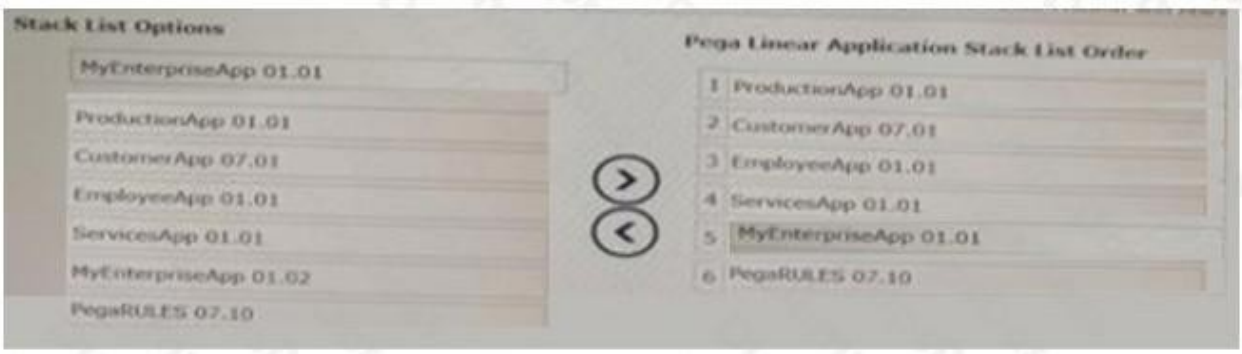
The following figure depicts a hierarchy of applications that are built on other applications.

Specifically , the productionapp application is built on applications customerapp and

employeeapp. Each of these applications has additional built-on applications, including the duplicated MYEnterpriseApp application, which has two different versions in use. All applications are built on the same version of PegaRUUS.



ANSWER:



Explanation:

A Pega application stack is the ordered rule-resolution context created from the current application and its built-on applications. The current application, **ProductionApp 01.01**, is at the top because rules defined there have the highest application-level precedence. Its directly built-on applications are then included in the order presented by the application hierarchy: **CustomerApp 07.01** followed by **EmployeeApp 01.01**. Directly built-on applications are placed ahead of applications that are reached farther down the hierarchy, which preserves the intended layering of the production application.

After those directly built-on applications, **ServiceApp 01.01** is added from the CustomerApp branch. The hierarchy reaches **MyEnterpriseApp 01.01** through the EmployeeApp branch before it reaches the alternate MyEnterpriseApp version through the ServiceApp branch. A linear application stack contains an application only once. Consequently, the first encountered MyEnterpriseApp entry is retained, and there is no second MyEnterpriseApp entry in the resolved stack. This is why the version placed in the stack is **MyEnterpriseApp 01.01**.

Finally, **PegaRULES 07.10** is the shared foundation for every application in the hierarchy, so it is placed at the bottom of the stack. This gives the expected rule-resolution order: production rules first, then the framework and shared application layers, and Pega platform rules last. Pega learning materials describe how applications are layered through built-on application relationships in the Pega Academy, and the platform documentation is available through Pega Documentation.

QUESTION NO: 3

Consider the following requirement:

case worker must be able to add a work party when creating a case

select the configuration option that fulfills this requirement

A. add a data transform to the pyCaseManagementDefault work parties rule

B. Ensure the addparty user action is available as a local action

C. add the addworkObjectparty activity to the assigments

D. Select the *Allow users to add work parties* option on the pyCaseManagementDefault work parties rule.

ANSWER: D

Explanation:

Selecting the *Allow users to add work parties* option in the *pyCaseManagementDefault* work parties rule enables case workers to add an appropriate work party while they are creating a case. A work party represents a person, organization, or operator associated with the case in a defined role, such as a customer, contact, or interested party. The work parties rule is the case-management configuration artifact that determines which party roles are available and whether users can supply party information during case entry.

When this setting is enabled, the case creation experience can present the party-entry capability to the case worker, allowing the worker to associate the relevant party record at the point the case is initialized. This supports correct case context from the start and makes the party available to downstream processing, correspondence, routing, and data relationships. The configuration is therefore applied at the work-party definition level rather than implemented through custom processing.

For background on configuring case behavior and participants, see Pega Academy: Case management and Pega Documentation.

QUESTION NO: 4

you are configuring authentication for a pega web mashup implementation.

how do you ensure the host system origin is trusted?

A. in the authentication service JNDI binding parameters, specify the protocol, host, and port.

B. in the authentication service spaecify the IACAuthverification activity

C. in the application definition, specify the protocol, host, and port

D. in the content security policy record specify the website as allowed

ANSWER: C

Explanation:

In a Pega Web Mashup implementation, the embedded Pega content is loaded from a host system that is distinct from the Pega application. The application must explicitly trust that host origin before it can participate in the mashup authentication and browser-based interaction flow. Configure the application definition with the host system's protocol, host name, and port. These values identify the complete browser origin, because an origin is defined by the scheme, host, and port together. For example, `https://portal.example.com:443` is treated as a specific origin and is not interchangeable with a different scheme, subdomain, or port.

This configuration lets Pega validate that requests associated with the mashup come from the approved embedding site. It is particularly important when a web application uses the Pega Web Mashup JavaScript integration to render Pega UI within the host page and establish the required authentication context. Maintain separate, precise origin entries for development, test, and production hosts rather than relying on broad or implied trust. See Pega's Pega Web Mashup documentation and the Pega Academy Web Mashup topic for implementation guidance.

QUESTION NO: 5

A partner application invokes a Pega REST service to create service-request cases. The partner must authenticate using OAuth 2.0, and the authenticated partner must be authorized to invoke only the intended service operation.

Which two configurations support the requirement? (Choose Two)

A. Configure an OAuth 2.0 authentication service for the partner integration.

B. Associate the authenticated operator with an access group that grants only the required access role and privilege.

C. Configure the REST service as unauthenticated and validate the partner name in a data transform.

D. Grant the partner operator the standard administrator access role.

ANSWER: A B

Explanation:

Configure an OAuth 2.0 authentication service for the partner integration. The authentication service validates the partner's access token and establishes an authenticated operator context before the REST service operation is processed.

Associate the authenticated operator with an access group that grants only the required access role and privilege for the service operation. Access roles and privileges enforce the authorized capabilities of the authenticated partner, ensuring that successful authentication alone does not provide access to unrelated case types or service operations. See Pega documentation on OAuth 2.0 authentication and authorizing users with access roles.

QUESTION NO: 6

You add database connection information to prconfig.xml and want to encrypt the password in the connection information how do you encrypt the password?

A. create an encrypted keyring password and replace the unencypted password in prconfig.xml

B. enter a password in an application ID instance and reference the external system

C. create an encrypted password with PR cipherGenerator and reference the external system

D. Create an encrypted password with passGen and replace the unencrypted password in prconfig.xml.

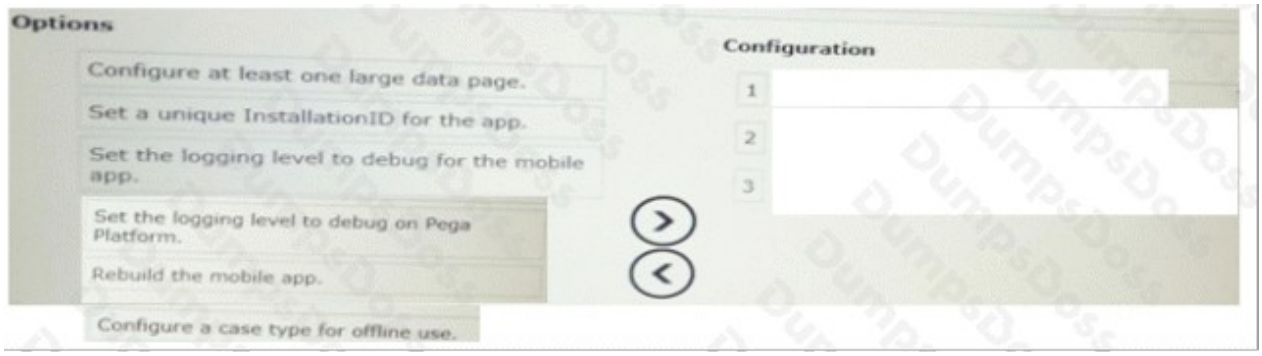
ANSWER: D

Explanation:

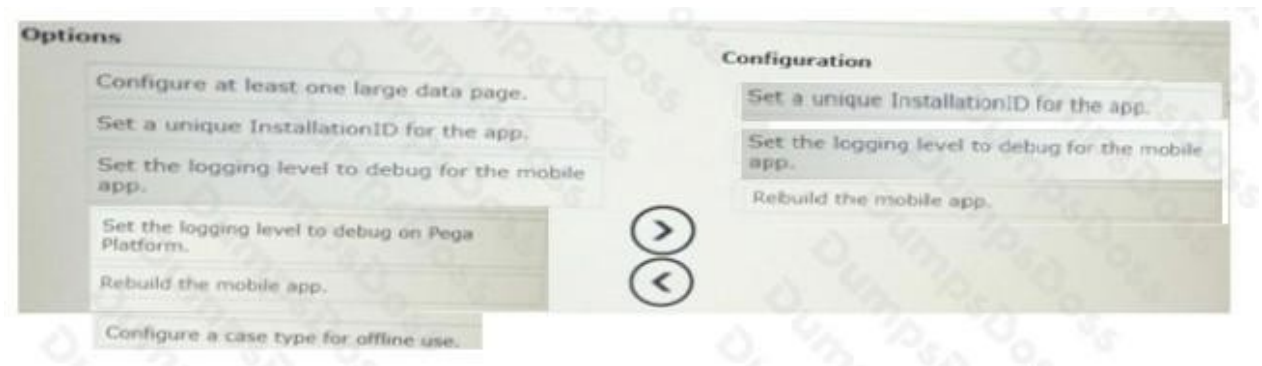
Create an encrypted password with passGen and replace the unencrypted password in prconfig.xml. The `passGen` utility is the Pega-supported mechanism for producing an encrypted value suitable for use in Pega configuration properties, including credentials used by database connection settings in `prconfig.xml`. Run the utility in the Pega installation environment and supply the database password as its input. It returns an encrypted representation that is pasted into the applicable password property in place of the plaintext value. At runtime, Pega uses its configured encryption facilities to resolve the protected value when it establishes the database connection. This protects the credential from being directly readable in the configuration file while preserving the expected configuration format and allowing the platform to connect normally after restart. Credential encryption should be performed before the configuration file is distributed or stored in source-control and deployment systems, and access to both the configuration and the Pega environment's cryptographic material should remain restricted. See the Pega Platform documentation portal for platform configuration and security guidance at Pega Documentation, and Pega Academy's official learning resources at Pega Academy.

QUESTION NO: 7 - (DRAG DROP)

Select and move the three option that are required to enable to access to the client clipboard in a mobile app to the configuration column and place them in the correct order



ANSWER:



Explanation:

Access to the client clipboard is enabled through the mobile application's diagnostic configuration. First, assign a unique InstallationID to the app. The InstallationID uniquely identifies the installed mobile app instance and lets the platform associate client-side diagnostic information with the intended application installation. This identifier is important when working with a particular mobile client rather than a generic server-side requestor session.

Next, set the logging level to debug for the mobile app. Client clipboard information is produced by the mobile client, so the relevant diagnostic setting must be enabled in the mobile app configuration. Debug logging provides the level of client-side detail needed when inspecting information generated and maintained by the mobile runtime.

Finally, rebuild the mobile app. Rebuilding is required because the InstallationID and mobile logging settings are packaged into the generated mobile application. An already-built application does not automatically receive these updated settings; the regenerated app must be installed or distributed for the mobile client to run with the required diagnostic configuration.

This sequence follows the standard approach of identifying the app instance, enabling the mobile-side diagnostics, and packaging those settings into a fresh build. Pega's mobile capabilities and configuration concepts are documented in the Pega Academy Mobile topic and the Pega Documentation portal.

QUESTION NO: 8

An attribute-based access control configuration evaluate the attributes of the operator creating the

case. A data page contains the value used in the evaluation.

A. Use the data page as a restriction method in the access control policy.

B. Use the data page as the policy condition value in the access control policy condition.

C. Use the data page as a the permit access if the value in the access control policy

D. Use the data page as a the policy condition column source in the access control policy condition

ANSWER: B

Explanation:

Use the data page as the policy condition value in the access control policy condition. In Pega attribute-based access control, an access control policy evaluates a policy condition against the relevant context, such as the operator who is creating a case. The condition identifies what is being evaluated and compares it with an appropriate value. When that comparison value must be obtained dynamically from a data page, the data page belongs in the policy condition's value configuration. This lets the policy retrieve the current value supplied by the data page at evaluation time and use it in the comparison that determines whether access is permitted. This approach keeps authorization logic declarative and centralized in the access control policy while allowing the compared value to be sourced from application data. Pega access control policies are designed to apply authorization decisions based on conditions and contextual attributes, including operator-related information. For additional details, see Pega documentation on access control policy rules and the Pega Academy material on attribute-based access control.

QUESTION NO: 9

Identify three rule types that are used in defining Authentication Service data instances. (Choose Three)

A. Data pages

B. Activities

C. Connectors

D. Reports

E. Data transforms

ANSWER: A B C

Explanation:

Authentication Service data instances can use **Data pages**, **Activities**, and **Connectors** to implement authentication behavior and integrate that behavior with Pega's security framework. An activity is the rule type used when custom authentication logic is needed, such as evaluating supplied credentials and establishing the authenticated operator context. A data page can provide a reusable, declarative way to obtain authentication-related information, including data sourced from an external system or a Pega data source. Connectors enable the authentication implementation to communicate with external identity systems and services, allowing credentials or identity attributes to be validated outside the Pega Platform application. Together, these rule types support configurable authentication flows while allowing organizations to centralize reusable data access and external-system integration. Pega's authentication configuration supports both built-in mechanisms and custom authentication processing that can invoke such rules as part of the authentication implementation. See the Pega Platform security documentation at Pega Platform security overview and the Pega Academy security learning resources at Pega Academy: Security.

QUESTION NO: 10

your application connects to two REST services that list details about bank offices.

One REST service uses a postal code as a GET parameter and returns a detailed list of bank officers in that postal code.

the other REST service uses a city name as a GET parameter and returns a datailed list of back offices in cities with that name. The application uses data pages to cache details about bank offices.

select two options to configure a data page to accept either a postal code or a city name as a parameter and call the appropriate REST connector.(choose TWO)

A. sourcing data from both rest connectors and using the response data transform to select the correct data

B. sourcing data from a single REST connector that is circumstanced based on the input parameter

C. sourcing data from each REST connector separately and using a when rule to select the appropriate data source at run time

D. sourcing data from an activity that evaluates the parameter to select the appropriate REST connector to call at run time

ANSWER: C D

Explanation:

A data page can define more than one data source and use a *when* rule on each source to determine which one is applicable for the current parameter values. Therefore, sourcing data from each REST connector separately and using a when rule to select the appropriate data source at run time is appropriate. The data page receives the city or postal-code parameter, and the applicable source invokes only the corresponding connector. This keeps the conditional source-selection behavior declarative and allows the data page to retain its normal caching behavior for the parameterized results.

Sourcing data from an activity that evaluates the parameter to select the appropriate REST connector to call at run time is also valid. An activity can inspect which parameter was supplied, set the required request parameter, and invoke the appropriate connector rule. The activity is then configured as the data page's source. This approach is useful when the selection or request preparation requires procedural logic beyond a simple declarative condition. In either configuration, the data page provides a single parameterized interface to callers while directing the request to the service that accepts the supplied search criterion. See Pega guidance on data pages and connectors.

QUESTION NO: 11

Which two tools and/or methods support continuous integration practices? (Choose Two)

A. Performing UI regression testing

B. Leveraging release toggles

C. Using an automation server to invoke unit test suites

D. Configuring pre and post import steps

ANSWER: B C

Explanation:

Leveraging release toggles supports continuous integration by separating deployment from feature exposure. Teams can merge and deploy code regularly while keeping incomplete, higher-risk, or selectively released functionality disabled until it is ready. This reduces the need for long-lived branches and enables smaller, safer integrations into a shared code line. In Pega applications, release toggles can be used to control whether a feature is available at run time, helping teams adopt an incremental delivery approach.

Using an automation server to invoke unit test suites is also a core continuous-integration practice. A CI server can detect or receive notification of changes, build or package the application, execute automated PegaUnit test suites, and report failures promptly. Fast automated feedback lets developers identify integration defects close to the change that introduced them, preserving the health of the main branch and making frequent integration practical. Pega's DevOps guidance describes automating quality checks and tests as part of CI/CD pipelines, while its PegaUnit capability provides automated unit testing for application rules. See Pega documentation on continuous integration and PegaUnit testing documentation.

QUESTION NO: 12

Which three actions do you recommend to address their performance issue and ensure optimal application performance in the future? (Choose Three.)

A. Dedicated a node to background processing.

B. Review application environment sizing.

C. Remove embedded dashboard reports from the manager portal.

D. Minimize the usage of robotic automations.

E. Review selection criteria of agent activities.

ANSWER: A B E

Explanation:

"Dedicated a node to background processing," "Review application environment sizing," and "Review selection criteria of agent activities" are the appropriate recommendations. The time-based, worsening nature of the issue indicates that scheduled or accumulating background workload should be investigated through AES. Reserving a node for background processing isolates agent and queue-processor work from the user-requestor nodes. This helps preserve responsive interactive processing when background demand rises during the afternoon and prevents that work from competing directly with end-user sessions for node resources.

Reviewing agent activity selection criteria is essential because poorly scoped criteria can select an excessive number of records, repeatedly select the same records, or create processing spikes at scheduled times. Correct criteria ensures that each background run processes the intended work efficiently and predictably. Reviewing environment sizing is also necessary before another division begins using the application. AES trend data can reveal sustained CPU, memory, thread, database, and node-utilization pressure, allowing the team to validate whether the current three-node topology has sufficient capacity and to plan additional capacity where needed. Pega's node-type guidance and performance-monitoring resources support separating workloads and using monitored operational data to make these decisions: Pega Platform documentation and Pega Academy.

QUESTION NO: 13

In order to produce a complete view of a customer, a customer service(CS) application requires reference data from multiple external systems. The customer data resides on a customerinfo page in a parent case type. The data on the CustomerInfo Page is also used by some of its subcases. Sometimes, the connectors that populate the data page are slow.

Which two of the following approaches for handling the required reference data from the external systems uses the least system resources? (Choose Two.)

A. Use Case Designer data propagation to copy the data to the subcases to populate the page contents.

B. Use the Load-DataPage method to initiate the population of the page contents.

C. Use the system of Record (SOR) data access pattern to populate the page contents.

D. Use the Snapshot data access pattern to populate the page contents.

ANSWER: B C

Explanation:

Use the Load-DataPage method to initiate the population of the page contents. is appropriate because it explicitly loads a configured data page when the application determines that the reference data is needed. The data page mechanism manages the connector invocation and then makes the resulting clipboard page available for reuse according to its configured scope, refresh strategy, and parameters. This avoids building separate integration logic and repeatedly invoking slow connectors for each use of the same reference data.

Use the system of Record (SOR) data access pattern to populate the page contents. is also appropriate for externally owned customer reference data. In the SOR pattern, Pega accesses the authoritative external source through a data page and can cache the retrieved result at the appropriate scope. This provides a shared, on-demand representation of the customer information for the parent case and its processing, minimizing redundant retrieval and avoiding unnecessary persistence or duplication of reference data. Proper data-page configuration, particularly scope and reload conditions, allows the application to balance connector latency with the required data freshness. Pega documents data pages as the standard abstraction for sourcing and reusing data from systems of record: Pega Platform documentation: Data pages. Additional data-integration guidance is available at Pega Academy: Data integration.

QUESTION NO: 14

You are audit a property named. Generic Code. The label on the form is named suggested generic substitute. when the field value changes, you want the history details to show as the name of the label.

you also want to record the name of the property

How do you ensure that the entry shows both the label and the property name?

A. On the pyTracksecurtiyChanges data tranform, in the source field, enter "suggested generic substitue (. GenricCode)"

B. on the pyTracksecurtiyChanges declare trigger activity, enter GenericCode as a parameter name and "suggested generic substitute" as a perameter description

C. on the work management landing page> Filed level Auditing tab in the source filed enter GenericCode. in the Description filed enter "suggested generic substitute"

D. on the work management landing page > Filed level Auditing tab,, in the source filed enter "suggested generic substituite(. GeneriCode)"

ANSWER: C

Explanation:

Use the Work Management landing page's field-level auditing configuration to define the audited property and its user-friendly history description. Entering `.GenericCode` in the Source field identifies the clipboard property whose changes Pega must audit. Entering `Suggested generic substitute` in the Description field supplies the readable label used in the audit-history entry. Pega associates that description with the configured source property, so the history can present the

business-facing field name while retaining the property identity, such as *Suggested generic substitute (.GenericCode)*. This is the appropriate configuration because field-level auditing is maintained centrally through the Work Management landing page rather than by modifying the internal security-change tracking data transform or declare-trigger implementation. The configuration applies when Pega records changes to the specified property in case history, supporting meaningful audit details for users and auditors without requiring custom logic. See Pega's documentation on configuring field-level auditing and the use of properties in Pega Platform.

QUESTION NO: 15

An insurance company wants to extend a native mobile app to allow customers to create claims using a claims management application implemented on the pega platfrom. As a claim is processed, update are sent to the mobile app as push notifications.

How do you satisfy this requirement ?

A. Package the claims management application as a SDK mobile app

B. Configure a service to creat claim cases when called from the native mobile app

C. configure the native mobile app to creat a claim case using the page API

D. Embed the claims management application in the native mobile app using a pega web Mashup gadget

ANSWER: A

Explanation:

Package the claims management application as a SDK mobile app is correct because Pega Mobile SDK is intended for extending a native iOS or Android application with Pega Platform application capabilities. The SDK provides the integration layer through which the native application can securely interact with the Pega claims application, allowing a customer to initiate and work with claim-related case processing from the mobile experience. It also supports the mobile application patterns required to receive notifications generated as work progresses in Pega, so customers can be informed of claim updates without having to continually open or refresh the application.

This approach keeps the claims lifecycle, business rules, case processing, and notification decisions in the Pega application while providing an appropriately native mobile client experience. In particular, it addresses both aspects of the requirement together: integration of a pre-existing native app with the Pega claims application and delivery of claim-status updates to the device. Pega documents its mobile capabilities and SDK-based approach in the Pega Mobile SDK documentation and provides mobile implementation guidance in the Pega Academy mobile topic.

QUESTION NO: 16

you want to expose a set of services for your application. Each service should be exposed as a separate WSDL. How do you accomplish this?

A. create a separate service package for each WSDL.

B. create a service listener for each WSDL.

C. Run the service wizard for each WSDL.

D. place the service rules in separate classes, one class per WSDL

ANSWER: A

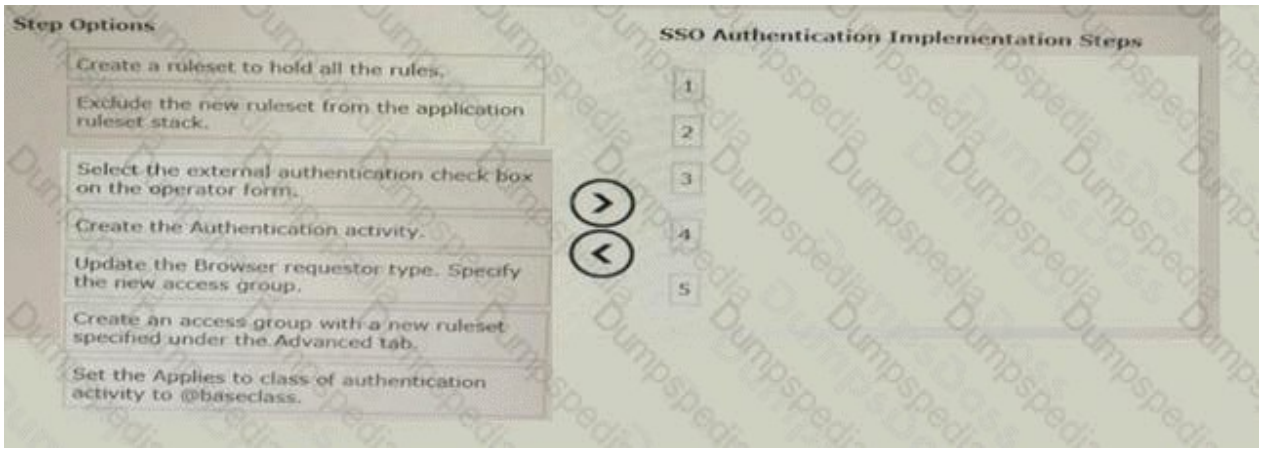
Explanation:

To expose each service through a separate WSDL, create a separate service package for each WSDL. In Pega, a service package is the SOAP exposure and deployment boundary: it identifies the group of service rules available through a particular endpoint and controls package-level settings such as authentication, requestor configuration, and access requirements. Pega generates the SOAP WSDL for the service package, so placing a service in its own package gives that

service its own distinct WSDL URL and contract. This design is useful when integrations need independently versioned or separately secured SOAP interfaces, while still allowing the implementation to use the appropriate Pega service rules underneath. The package name is part of the SOAP service endpoint/WSDL addressing, enabling external consumers to retrieve the contract for that package directly. Configure the service package first, associate the intended SOAP service rule or rules with it, and provide consumers the generated WSDL URL for that package. See Pega's documentation on creating service packages and SOAP service rules.

QUESTION NO: 17 - (SIMULATION)

Select and move the five steps required to implement single sign-on (SSO) authentication in a pega application to the SSO authentication implementation steps column. (choose five)



ANSWER: See the explanation for the answer

Explanation:

Move these five options to the **SSO Authentication Implementation Steps** column, in this order:

1. Create a ruleset to hold all the rules.
2. Create an access group with the new ruleset specified under the **Advanced** tab.
3. Update the **Browser** requestor type and specify the new access group.
4. Create the Authentication activity.
5. Set the **Applies To** class of the authentication activity to `@baseclass`.

Do not select *Exclude the new ruleset from the application ruleset stack* or *Select the external authentication check box on the operator form*. The Browser requestor type must use the dedicated access group so that the authentication activity is available before the user is authenticated; the authentication activity is defined on `@baseclass`.

QUESTION NO: 18

The clipboard contains a list of details for all companies on the NASDAQ stock Exchange

select the requirement that indicates you use a page list instead of a page group.

A. A DATA transform must iterate over the list.

B. company details must be referenced using the company stock triker.

C. the list must be sorted by company stock ticker.

D. Each element in the list of companies must contain a list of accounts.

ANSWER: C

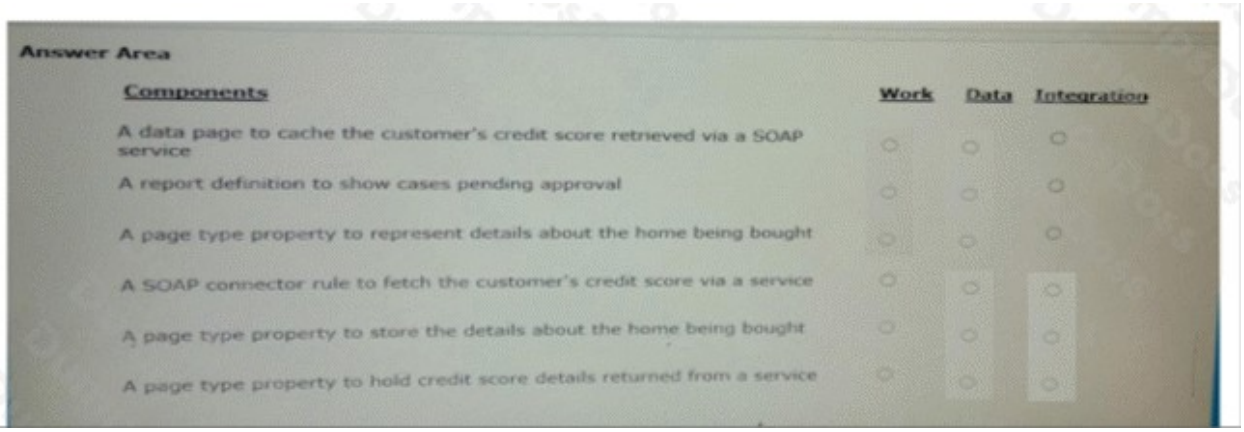
Explanation:

The requirement that *the list must be sorted by company stock ticker* indicates use of a Page List. A Page List represents an ordered collection of pages, with each page occupying a sequential position. This ordered structure makes it appropriate when the application must establish, preserve, or process a defined ordering of company records, such as alphabetical or ticker-symbol order. The application can populate the collection, sort it by the ticker property, and then access or iterate through the companies in that resulting sequence. This aligns with the fundamental Pega data-modeling distinction that a Page List is used for an ordered set of embedded pages. The individual company pages can still contain all relevant company and account details; the important requirement here is that the collection itself has an explicit sortable order. Pega's data-modeling learning materials describe Page Lists as collections used to maintain pages in a sequence and support processing records in that sequence. See the Pega Academy data modeling topic and the Pega Platform documentation for guidance on choosing property modes for embedded data.

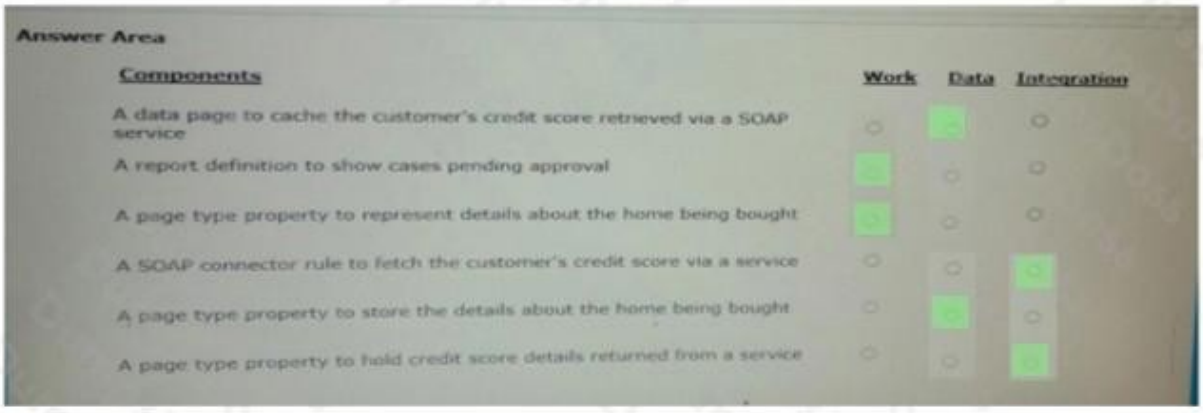
QUESTION NO: 19 - (HOTSPOT)

A case is used to process home loans for approval.

in the answer area, area, select the enterprice class structure (ECS) layer in which to bulid each of the following components. each layer of the ECS contains two components.



ANSWER:



Explanation:

The selected ECS-layer assignments correctly separate case processing, reusable business data, and external-service implementation. The home-loan approval case is the central business process, so artifacts that directly support the lifecycle and visibility of those loan cases are built in the Work layer. That includes the report definition for pending approvals, because it reports on work objects, and the page property representing the home being purchased, because those details are part of the information processed by a specific home-loan case.

The Data layer is the appropriate place for reusable, business-oriented data resources that are not themselves the mechanics of connecting to an outside system. The cached credit-score data page belongs there because it makes credit-score information available for use by the application as a data asset. Likewise, the page property that stores home details is

correctly classified as Data when it represents the reusable business-data structure for a home. This keeps the home data model independent of a particular case implementation and promotes reuse.

The Integration layer contains the implementation details that communicate with systems of record and represent their service interactions. The SOAP connector rule belongs in this layer because it invokes the external service that supplies the credit score. The page property that holds credit-score details returned by that service also belongs with the integration implementation, where the response can be represented and mapped before the application uses it. This division supports Pega's layered architecture by maintaining clear boundaries between case behavior, data assets, and external-system connectivity. Pega's documentation on application layering and class structure provides the architectural rationale for this separation: Pega application layering and Pega class structure.

QUESTION NO: 20

An application include two case types: expense report and purchase requests which two steps are required to enable support for both case type for offline user on a mobile device ?

A. Configure the application record to enable offline access

B. Configure the user's access group to enable offline access.

C. configure both case type to enable offline access

D. Configure the application to build a custom mobile app

ANSWER: B C

Explanation:

Configure the user's access group to enable offline access. is required because offline capability is authorized at the access-group level. The access group used by mobile users must permit offline operation so that the mobile client can download the relevant application rules and data for use when a network connection is unavailable. This setting also ensures that the access group is included appropriately in the offline mobile application configuration.

configure both case type to enable offline access is also required because offline support is configured for the individual case types that users must be able to create, open, and process while disconnected. Since the application contains both Expense Report and Purchase Request case types, each must be explicitly made available offline. Pega then packages the required case views, data pages, and supporting rules for those offline-enabled workflows into the mobile app's offline cache. The access-group authorization and the case-level offline configuration work together: one grants the users offline capability, while the other identifies the work that the device can handle offline. See the Pega Documentation portal and Pega Academy for Pega Mobile and offline-capability configuration guidance.