

Übungsprüfung für Senior System Architekten

Die zertifizierte Pega System Architect Version '24.2 Prüfung besteht aus 60 Fragen. Nachfolgend finden Sie eine Stichprobe.

Frage 1

Ein Entwicklungsteam erstellt eine Pega-Anwendung für ein Finanzdienstleistungsunternehmen. Die Anwendung verarbeitet hochsensible Kundendaten, einschließlich Sozialversicherungsnummern (SSN) und Kreditkartendaten, die sowohl bei der Speicherung in der Datenbank als auch bei der Anzeige in der Zwischenablage geschützt werden müssen. Das Team muss außerdem sicherstellen, dass diese Daten nicht in Protokollen oder Suchindizes offengelegt werden.

Welche der folgenden Verschlüsselungsansätze sollte der System Architekt konfigurieren, um diese Anforderungen zu erfüllen? (Wählen Sie alle zutreffenden aus.)

- A.** Implementierung einer benutzerdefinierten Chiffre und Schlüsselkombination für ein hybrides Verschlüsselungsmodell.
- B.** Verwendung eines irreversiblen Hashing-Prozesses für die sensiblen Eigenschaften.
- C.** Anwendung von Verschlüsselung auf Klassenebene (BLOB), um den gesamten Datensatz im Ruhezustand zu schützen.
- D.** Konfiguration einer PropertyEncrypt Access Control Policy für die spezifischen sensiblen Eigenschaften.
- E.** Aktivierung der automatischen Schlüsselrotation als erweiterte Konfiguration.

Frage 10

Ein System Architekt hat die Aufgabe, eine bestehende Pega-Anwendung zu sichern. Das Datenbankschema der Anwendung wurde für Berichtszwecke optimiert, indem mehrere Spalten offengelegt wurden, die sensible Kundeninformationen enthalten. Diese offengelegten Daten sind jedoch nicht verschlüsselt, was eine Sicherheitslücke darstellt. Das Team muss diese sensiblen Daten verschlüsseln und gleichzeitig ihre Verwendung in Berichten und anderen Pega-Funktionen ermöglichen.

Welche der folgenden Maßnahmen sollte der System Architekt ergreifen, um diese Schwachstelle zu beheben? (Wählen Sie alle zutreffenden aus.)

- A.** Konfiguration der automatischen Schlüsselrotation, um sicherzustellen, dass die offengelegten Daten verschlüsselt werden.
- B.** Implementierung von Verschlüsselung auf Klassenebene (BLOB), die automatisch alle offengelegten Eigenschaften verschlüsselt.
- C.** Verwendung von Verschlüsselung auf Eigenschaftsebene, da sie Eigenschaften außerhalb des BLOB verschlüsseln kann, einschließlich offengelegter Spalten.
- D.** Konfiguration einer PropertyEncrypt Access Control Policy für die spezifischen Eigenschaften, um sie in der Datenbank, Zwischenablage und in Berichten zu verschlüsseln.
- E.** Änderung des Eigenschaftstyps in TextEncrypt, was die Standardpraxis für die Verschlüsselung sensibler Daten ist.

Frage 20

Eine Organisation muss eine Reihe von Berichten erstellen, um die Effizienz ihres Kreditantragsverfahrens zu überwachen. Das Unternehmen interessiert sich für zwei Hauptbereiche:

- Ermittlung der durchschnittlichen Zeit für die Genehmigung eines Kreditantrags.
- Verfolgung der Anzahl der Kreditanträge, die für einen bestimmten Kredittyp eingereicht werden.

Welche zwei Arten von Metriken und welche zwei Klassen würde ein Pega System Architekt verwenden, um diese Anforderungen zu erfüllen? (Wählen Sie alle zutreffenden aus.)

- A.** Business Metrics für die Anzahl der Kreditanträge.
- B.** Process Metrics für die durchschnittliche Zeit zur Genehmigung eines Kreditantrags.
- C.** Die History-Klasse zur Verfolgung der Anzahl der Kreditanträge.
- D.** Die Work-Klasse für die Anzahl der Anträge und die History-Klasse für die Genehmigungszeit.
- E.** Die Assign-Worklist-Klasse für die Anzahl der Anträge und die History-Klasse für die Genehmigungszeit.

Frage 30

Ein Pega System Architekt hat die Aufgabe, eine neue Claims-Anwendung mit einem benutzerdefinierten, nicht-nativen Front-End zu integrieren. Das Unternehmen verlangt, dass die Integration hochperformant ist, mit minimalem Netzwerkverkehr und einer umfangreichen Daten-Payload für eine einzelne Anfrage. Das Team erwägt die Verwendung der Pega DX API.

Welche zwei der folgenden sind Vorteile der Verwendung der Constellation DX API gegenüber der Traditional DX API in diesem Szenario? (Wählen Sie zwei aus.)

- A.** Die Constellation DX API ermöglicht die Verwendung traditioneller Pega-Harnesses und Sections zum Aufbau der Benutzeroberfläche.
- B.** Die Constellation DX API hat eine siebenmal kleinere Netzwerk-Payload und 30% weniger Anfragen beim ersten Laden.
- C.** Die Constellation DX API-Antworten sind weniger robust und erfordern zusätzliche API-Aufrufe für Validierung und Aktionen.
- D.** Die Constellation DX API bietet eine Reihe von modellgesteuerten API-Endpunkten, die sowohl von nativen als auch von nicht-nativen UIs verwendet werden können.
- E.** Die Constellation DX API basiert auf einer völlig anderen JSON-Struktur, die Daten vom View-Layout trennt.

Frage 40

Ein Pega System Architekt behebt einen fehlgeschlagenen Connector, der mit einem Drittanbieter-Versanddienst integriert ist. Der Fehler tritt auf, wenn der Dienst ein ungültiges Antwortformat zurückgibt, das der Pega-Connector nicht analysieren kann. Der Architekt muss einen robusten Fehlerbehandlungsmechanismus implementieren, um dieses Problem zu beheben.

Welche zwei der folgenden sind Best Practices für die Behandlung dieser Art von Integrationsfehler? (Wählen Sie zwei aus.)

- A.** Verwendung des pxErrorHandlingTemplate Data Transform zur Erstellung eines wiederverwendbaren Data Transform für die Fehlererkennung in der Antwort.
- B.** Der ConnectionProblem-Flow sollte so konfiguriert werden, dass das Arbeitsobjekt an einen Workbasket weitergeleitet und ein Operator über das Problem benachrichtigt wird.
- C.** Schreiben der Fehlerdetails, einschließlich der Connector-Anfrage und -Antwort, in die Protokolldatei zur Unterstützung der Fehlerbehebung.
- D.** Ausschließliches Verlassen auf den Error Handling Flow zur Erkennung und Behandlung aller Arten von Integrationsfehlern, da er immer aktiviert ist.
- E.** Konfiguration einer When-Bedingung im Response Data Transform zur Überprüfung auf spezifische Fehlermeldungen und anschließende Anwendung des wiederverwendbaren Fehlerbehandlungs-Data Transform.

Frage 50

Ein Pega-Entwickler verwendet eine Activity zur Automatisierung eines komplexen Geschäftsprozesses. Die Activity muss eine andere Activity aufrufen und dann die Steuerung an die ursprüngliche aufrufende Regel zurückgeben, nachdem die zweite Activity abgeschlossen ist. Welche Schrittanweisung sollte der Entwickler in der ersten Activity verwenden, um dieses Verhalten zu erreichen?

- A.** Branch
- B.** Call
- C.** Jump
- D.** Fork
- E.** Queue

Frage 60

Eine Pega-Anwendung erfordert einen Batch-Prozess, der eine große Anzahl von Arbeitsobjekten aus der Datenbank liest, ihren Status aktualisiert und einige Geschäftslogik ausführt. Dieser Prozess ist zeitaufwendig und muss robust genug sein, um Systemneustarts zu bewältigen, ohne Daten zu beschädigen. Was sind die Best Practices für die Implementierung dieses Hintergrundprozesses? (Wählen Sie alle zutreffenden aus.)

- A.** Aufteilung großer Arbeitseinheiten in kleinere, einzeln verarbeitete Arbeitseinheiten.
- B.** Verwendung eines Queue Processors, da er für regelmäßig wiederkehrende Aufgaben am besten geeignet ist.
- C.** Implementierung von Checkpoints durch Speichern des Status von Jobs in persistentem Speicher oder einer Nachrichtenwarteschlange.
- D.** Verwendung eines Job Schedulers und einer einzelnen Activity zur Ausführung der gesamten Logik.
- E.** Verwendung der Obj-Save- und Commit-Methoden innerhalb einer Schleife, um jeden Datensatz bei der Verarbeitung zu speichern.

LÖSUNGSSCHLÜSSEL

Frage 1

Richtige Antworten: D und C

- A.** Falsch. Obwohl Pega eine Chiffre und einen Schlüssel verwendet, fragt die Frage nach dem korrekten Pega Platform-Ansatz, nicht nach einer benutzerdefinierten Lösung. Die integrierten Verschlüsselungsmechanismen von Pega sind der richtige Weg.
- B.** Falsch. Hashing ist ein irreversibler Prozess, der hauptsächlich für Passwörter verwendet wird, nicht für Daten, die von autorisierten Benutzern entschlüsselt und eingesehen werden müssen.
- C.** Richtig. Verschlüsselung auf Klassenebene (BLOB) ist eine effiziente Methode zur Verschlüsselung eines gesamten Case- oder Datendatensatzes, der als Binary Large Object in der Datenbank gespeichert ist. Sie schützt die Daten im Ruhezustand, was eine Hauptanforderung ist.
- D.** Richtig. Verschlüsselung auf Eigenschaftsebene wird durch Konfiguration einer PropertyEncrypt Access Control Policy implementiert. Dieser Ansatz ist speziell dafür ausgelegt, Eigenschaften innerhalb und außerhalb der Datenbank zu verschlüsseln, einschließlich Zwischenablage, Protokollen, Suchindizes und Berichten. Dies erfüllt direkt die Anforderung zum Schutz von Daten an diesen spezifischen Orten.
- E.** Falsch. Automatische Schlüsselrotation ist eine Best Practice für Sicherheit, aber eine erweiterte Konfiguration und bestimmt nicht allein den Verschlüsselungsansatz.

Frage 10

Richtige Antworten: C und D

- A.** Falsch. Automatische Schlüsselrotation ist eine Sicherheits-Best Practice, bestimmt aber nicht die Verschlüsselungsmethode selbst oder ob offengelegte Spalten verschlüsselt werden.
- B.** Falsch. Verschlüsselung auf Klassenebene (BLOB) verschlüsselt speziell keine Eigenschaften, die als Spalten für Berichtszwecke offengelegt sind. Dieser Ansatz würde das Problem nicht lösen.
- C.** Richtig. Verschlüsselung auf Eigenschaftsebene ist der richtige Ansatz, da sie Eigenschaften sowohl innerhalb als auch außerhalb der Datenbank verschlüsselt, einschließlich Eigenschaften, die für Berichtszwecke optimiert sind.
- D.** Richtig. Verschlüsselung auf Eigenschaftsebene wird durch Konfiguration einer PropertyEncrypt Access Control Policy implementiert. Diese Policy verschlüsselt die Daten an verschiedenen Orten, einschließlich Datenbank, Zwischenablage, Protokollen und Suchindizes.
- E.** Falsch. Der TextEncrypt-Eigenschaftstyp ist veraltet, und der richtige Ansatz besteht darin, eine PropertyEncrypt Access Control Policy zu konfigurieren.

Frage 20

Richtige Antworten: B und D

- A.** Falsch. Business Metrics würden für die Anzahl der Kreditanträge verwendet, was ein Maß für den Erfolg oder Misserfolg eines Geschäftsprozesses ist, aber diese Antwort liefert nur die Hälfte der korrekten Information. Die Frage verlangt sowohl nach Metriktypen als auch nach den zugehörigen Klassen.
- B.** Richtig. Process Metrics messen "wie Arbeit durchgeführt wird". Die Zeit, die für die Erledigung einer Aufgabe benötigt wird, ist eine Prozessmetrik. Die durchschnittliche Zeit zur Genehmigung eines Kreditantrags ist eine Prozessmetrik.
- C.** Falsch. Die History-Klasse wird verwendet, um Leistung und Leerlaufzeit zu messen. Die Work-Klasse speichert den Case selbst und ist der richtige Ort, um die Anzahl der Anträge zu finden.
- D.** Richtig. Die Work-Klasse enthält die Case-Daten, sodass ein Bericht über diese Klasse die Anzahl der erstellten Kreditantrags-Cases zählen kann. Die History-Klasse enthält Leistungsdaten, wie z.B. für Aufgaben aufgewendete Zeit, die zur Berechnung der durchschnittlichen Genehmigungszeit verwendet würde.
- E.** Falsch. Assign-Worklist wird für Aufgaben eines bestimmten Benutzers verwendet. Die Anzahl der Anträge würde in der Work-Klasse gefunden.

Frage 30

Richtige Antworten: B, D und E

- A.** Falsch. Die Constellation-Architektur, zu der die DX API gehört, ist eine Abkehr von traditionellen Harnesses und Sections hin zu einem modellgesteuerten Ansatz.
- B.** Richtig. Die Constellation DX API "hat den Netzwerkverkehr drastisch reduziert mit 30% weniger Anfragen beim ersten Laden und einer siebenmal kleineren Netzwerk-Payload", was direkt die Anforderung an eine hochperformante Integration mit minimalem Netzwerkverkehr erfüllt.
- C.** Falsch. Das Dokument besagt, dass die "DX API-Antworten viel robuster sind und alle notwendigen Informationen enthalten, um Views in jedem Front-End-Framework zu reproduzieren, einschließlich Layouts, Feldern, Validierung, bedingter Sichtbarkeit und Aktionen."
- D.** Richtig. Die Constellation DX API "unterstützt sowohl native Pega-UIs als auch nicht-native UIs", was eine Hauptanforderung für das Szenario der Integration mit einem benutzerdefinierten Front-End ist. Die "gleichen API-Endpunkte ermöglichen sowohl die Out-of-the-Box-Benutzererfahrungen von Pega als auch benutzerdefinierte Front-End-Erfahrungen."
- E.** Richtig. Das Dokument besagt ausdrücklich, dass die Constellation DX API "eine völlig neu

entwickelte Erfahrung" ist und "eine völlig andere JSON-Struktur hat, die Daten deutlich vom View-Layout trennt".

Frage 40

Richtige Antworten: C und E

- A.** Falsch. Das `pxErrorHandlingTemplate` ist ein wiederverwendbarer Data Transform, wird aber typischerweise von einer anderen Regel aufgerufen (wie einem Response Data Transform), nicht zur direkten Fehlererkennung verwendet. Die Fehlererkennungslogik wird in der aufrufenden Regel konfiguriert.
- B.** Falsch. Der `ConnectionProblem`-Flow ist der Standard-Fehlerbehandlungs-Flow, aber ein Flow wird im Allgemeinen für Fehler aufgerufen, die nicht durch einen Data Transform erkannt werden. In diesem Szenario ist das Problem ein ungültiges Antwortformat, das typischerweise in einem Data Transform behandelt wird. Ein spezifischerer, benutzerdefinierter Flow könnte erforderlich sein, aber der Standard-Flow wäre für Verbindungsprobleme.
- C.** Richtig. Das Schreiben detaillierter Fehlerinformationen, einschließlich Anfrage und Antwort, in die Protokolldatei ist eine kritische Best Practice für die Fehlerbehebung bei Integrationsproblemen. Dies hilft Entwicklern, Muster und die Grundursache des Fehlers zu identifizieren.
- D.** Falsch. Obwohl der Error Handling Flow immer aktiviert ist, ist es keine Best Practice, sich für alle Fehler darauf zu verlassen. Für Fehler, die früh erkannt werden können (z.B. in einem Response Data Transform), bietet deren dortige Behandlung eine bessere Benutzererfahrung und vermeidet den Overhead eines Flows.
- E.** Richtig. Die Konfiguration einer When-Bedingung im Response Data Transform zur Überprüfung auf spezifische Fehlermeldungen oder eine fehlerhafte Antwort und anschließendes Aufrufen eines wiederverwendbaren Fehlerbehandlungs-Data Transform ist eine Standard-Best Practice für die Behandlung von Integrationsfehlern.

Frage 50

Richtige Antwort: B

- A.** Branch - Dies ist falsch. Branch führt die angegebene Activity aus, gibt aber die Steuerung an die Regel zurück, die ursprünglich die erste Activity aufgerufen hat. Die erste Activity wird beendet.
- B.** Call - Dies ist die richtige Antwort. Die Call-Schrittanweisung führt die angegebene Activity aus und gibt dann die Steuerung nach Abschluss an die aufrufende Activity zurück, sodass die erste Activity ihre eigene Verarbeitung fortsetzen kann.
- C.** Jump - Dies ist falsch. Jump ist eine Schrittanweisung, die die Steuerung auf einen anderen Schritt innerhalb derselben Activity überträgt, nicht auf eine andere Activity.

D. Fork - Dies ist falsch. Fork ist keine standardmäßige Pega-Activity-Methode.

E. Queue - Dies ist falsch. Queue ist eine Methode, die Teil der Queue-For-Processing-Anweisung ist und für asynchrone Verarbeitung verwendet wird. Sie gibt die Steuerung nicht synchron an die aufrufende Activity zurück.

Frage 60

Richtige Antworten: A, C

A. Aufteilung großer Arbeitseinheiten in kleinere... - Dies ist eine richtige Antwort. Für einen robusten Hintergrundprozess ist es Best Practice bei Pega, große Aufgaben in kleinere, besser handhabbare aufzuteilen. Dies verhindert, dass ein einzelner Fehler den gesamten Prozess beeinträchtigt, und erleichtert die Wiederherstellung.

B. Verwendung eines Queue Processors... - Dies ist falsch. Ein Queue Processor dient zur Verarbeitung einzelner Aufgaben, die in einer Warteschlange stehen. Für einen regelmäßig geplanten Batch-Job ist ein Job Scheduler das geeignete Werkzeug.

C. Implementierung von Checkpoints durch Speichern des Status von Jobs... - Dies ist eine richtige Antwort. Checkpointing ist eine entscheidende Best Practice für lang laufende Hintergrundprozesse. Durch Speichern des Job-Status kann der Prozess Neustarts oder Fehler ordnungsgemäß behandeln, ohne Daten zu beschädigen oder von vorne zu beginnen.

D. Verwendung eines Job Schedulers und einer einzelnen Activity... - Dies ist falsch. Während ein Job Scheduler das richtige Werkzeug ist, ist die Verwendung einer einzelnen Activity für die gesamte Logik eine schlechte Praxis. Best Practice ist die Trennung von Geschäftslogik und Transaktionslogik für bessere Wartbarkeit und Fehlerbehandlung.

E. Verwendung der Obj-Save- und Commit-Methoden innerhalb einer Schleife... - Dies ist falsch. Die wiederholte Verwendung von Obj-Save und Commit in einer Schleife ist ein Performance-Anti-Pattern. Es kann zu Datenbank-Sperrproblemen führen und ist ineffizient. Die Hintergrundverarbeitungsfunktionen sind so konzipiert, dass sie Transaktionen effektiver handhaben.