OutSystems.Architecture-Specialist-11.v2024-03-19.q27

試験コード:	Architecture-Specialist-11
試験名称:	Architecture Specialist (OutSystems 11) Exam
認定資格:	OutSystems
無料問題数:	27
バージョン:	v2024-03-19
アクセス数:	728
ページビュー数:	270
https://www.jpnpdf.com/OutSystems.Architecture-Specialist-11.v2024-03-19.q27-mondaishu.html	

最新問題: **1**
どのアーキテクチャ キャンバス レイヤーで再利用率が高いと予想されますか?
A. エンドユーザー層
B. コア層
C. 基礎層
Answer: C (メーッセージを残す**)**

最新問題: **2**
OAPI は API とは異なります。OAPI の中身は何ですか?
A. サービスアクション
B. REST API
Answer: (解答を表示する**)**

最新問題: **3**
_IS モジュールは
A. テーマ、ルック アンド フィール要素、メニューなど。
B. 同じタイプの操作 (プリンターなど) を実行する、異なるシステムとの複数の統合サービス。特殊な実装 (透過サービス パターンなど) を使用して、同じ API を公開する複数のドライバーを作成できます。
C. 外部サービスを利用および正規化するためのテクニカル ラッパー。
D. 汎用ライブラリモジュール。
E. レイアウトと表示のみのための再利用可能な UI パターン。ビジネス ロジックは含まれません。
Answer: C (メーッセージを残す**)**

最新問題: **4**

Foundation レイヤー モジュールで見つけられるデータ要素は次のうちどれですか?
A. エンティティの CRUD アクション
B. 非コアエンティティ
C. コア エンティティ (読み取り専用として公開)
D. UI 状態を保存するエンティティ
Answer: B (メーッセージを残す**)**

最新問題**: 5**
モバイル アプリケーションの同期のベスト プラクティスではないものは次のうちどれですか?
A. デルタのみを同期します。: 制御タイムスタンプを使用します。
C 単一トランザクションでの長時間の同期を避ける
B. 関連するデータのみを同期します
C. 必要な場合にのみ同期します。p セッション開始時: 概要データをローカル ストレージに同期 プロセス開始時: プロセスの詳細をローカル ストレージに同期 プロセス/トランザクション終了時、サーバーに同期して戻す
D. 小規模で軽量な更新を保証するためにエンティティー括操作を使用しないでください。
E. 同期の粒度を確保します。ネットワークの損失によって作業を繰り返すことなく、継続的な再試行に備えます。チャンク単位でエンティティごとに増分同期
F. 厳密に必要な場合にのみ同期します
E すべての画面またはオンライン イベントで同期しない
Answer: F (メーッセージを残す**)**

最新問題**: 6**
断片化されたシステムまたはマイクロサービスの欠点ではないものは次のうちどれですか?
A. メモリ内の限定されたデータ マッシュアップと API に限定された
B. 複数のトランザクションが個別にコミットされました
C. モニタリングとロギング: 効果的なモニタリングとロギングには、一元化されたサービスが必要です
D. 複雑なデバッグとトラブルシューティング: 根本原因はサービス チェーンの奥深くにある可能性があります
E. フォールト トレランス: 通信エラー、サービスの一貫性
F. プロセス間通信: ネットワーク遅延と中断
G. 上記のすべて
H. セキュリティ: 資格情報とアクセス管理を管理する必要があります
Answer: (解答を表示する**)**

最新問題**: 7**
Foundation には非ビジネス サービスとビジネス サービスが含まれる場合があります

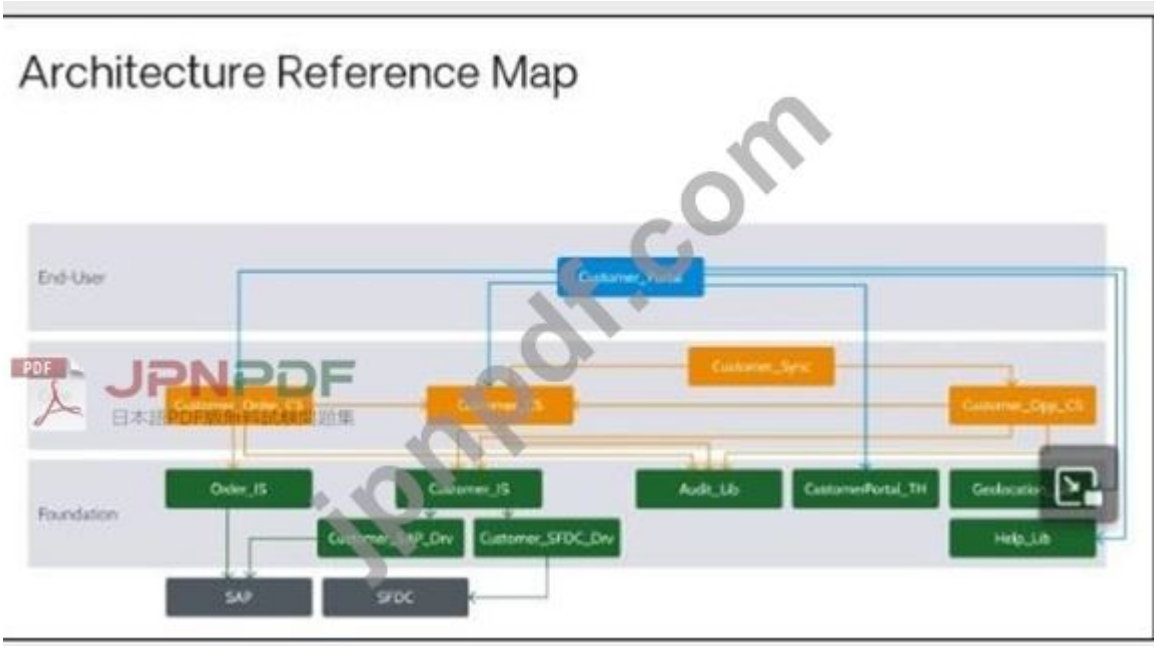
A. 偽
B. TRUE
Answer: A (メーッセージを残す**)**

最新問題**: 8**
アーキテクチャ ダッシュボード (AI メンター) を考慮すると、次の文のうち間違っているものはどれですか?
A. アーキテクチャ ダッシュボードは、パフォーマンス、セキュリティ、アーキテクチャ、保守性に関連するパターンを明らかにするコード分析を実行します。
B. アーキテクチャ ダッシュボードを使用すると、見つかったコード パターンを解決するソリューションを自動的に適用できます。
Answer: (解答を表示する**)**

最新問題**: 9**
以下のうち、コア モジュール パターンに最も一致するものはどれですか - ECS サマリー キャッシュのみのバリエーション
A. ... は 2 つのモジュールを含むパターンです。1 つは入出力構造を持つ外部 API をカプセル化するために使用できるコネクタ モジュール、もう 1 つは正規化された API をコンシューマに公開するためのラッパー モジュールです。
B. ローカル レプリカを使用する ECS と同じですが、API モジュールが提供されます。したがって、外部システムへの変更は OS に通知され、OS は ERP システム (サブスクリプション システム) からアップデートを取得します。
C. 基本 ECS パターンと同じですが、ローカル レプリカがあります。ローカル キャッシュとして機能するデータを保存します。長所: エンティティの使用を活用し、よりシンプルな統合 API。短所: ソースシステムへの影響が少ない
D. ... データが複数の外部システムから送信されている場合に必要です。IS はデータに応じて使用するドライバーを決定します。
E. ... 外部ライブラリ内のコードを公開するロジック、アクション、データを含めるため、または外部データベースを検査してデータ構造をインポートして、内部のエンティティとして使用できるようにするために使用されるラッパー。OS
F. ... 頻繁にリスト化、結合、または検索される概要データのみをキャッシュします。(*) 単一エントリの完全な詳細は、外部システムから直接取得されます。データベース全体が大きすぎるか、同期するにはコストがかかる場合に使用します。詳細は単一のエンティティにのみ必要です (リストではありません)
G. ... エンティティは読み取り専用として公開されており、API を使用してエンティティ ^ の作成/更新のビジネス ロジックを一元化できます
H. ローカル レプリカを使用する ECS と同じですが、同期ロジックが分離されています。長所: コードの独立性。CS のコンシューマは同期の影響を受けません。同期は複数の CS を調整できます

I. ... ローカル キャッシュ エンティティからデータを取得しようとしますが、存在しない場合は、外部 p システムから単一のエントリを取得します。そのレコードのみをキャッシュします (リードスルー キャッシュ) データベース全体が大きすぎるか、同期するにはコストがかかりすぎる場合に使用します。統合はデータベースのほんの一部にのみ影響します。データのリストに事前にアクセスする必要がある場合は避けてください。
J. ... エンティティはOutsystemsではなく、外部ERPシステムにあります。IS は にリモート呼び出しを行うだけです。外部システム/データベース。OS内にデータは保持されません。情報を取得するには 2 つの異なるシステムを横断する必要があるため、データ取得が最適化されない可能性があります。短所: 統合 API はすべてのユースケースをサポートする必要がある
Answer: F (メーッセージを残す**)**

最新問題**: 10**
アーキテクチャ ブループリントとは何ですか?



A. 上記が答えです
B. アーキテクチャ設計プロセスの最終結果です。この時点で、モジュール間に依存関係を追加する必要があります。
Answer: B (メーッセージを残す**)**

最新問題**: 11**
メニューやログインなどの要素は、さまざまなアプリ間で共有できます。メニューとログインはテーマ内で定義されます。これはどのユースケースですか?
A. 独立したアプリ
B. イントラネット
C. ポータル
Answer: C (メーッセージを残す**)**

最新問題: **12**
OutSystemsでは、Foundation、Core、およびEnd-Userモジュールを含むアプリケーションは、
A. 財団アプリケーション。
B. コア アプリケーション。
C. エンドユーザー アプリケーション。
Answer: (解答を表示する**)**

最新問題: **13**
CS モジュールのモバイル バージョンの一般的な命名規則は何ですか?
A. モバイル_CS
B. M_CS
C. m_CS
D. m_cs
E. _CS
Answer: C (メーッセージを残す**)**

最新問題: **14**
軽量モバイル ローカル ストレージを設計するための適切なアドバイスではないものは次のうちどれですか?
A. 関連性があるか無関係であるかに関係なく、すべてのレコードを保存します。
B. ローカル ストレージを非正規化します。
C. グループ関連情報
D. 使用される属性のみを定義します
Answer: A (メーッセージを残す**)**

最新問題: **15**
アーキテクチャ設計プロセスの整理部分では何が行われますか?
A. コンセプトはモジュールに組み立てられます
B. コンセプトはアーキテクチャ キャンバスのレイヤーに追加されます。
C. コンセプトはアーキテクチャ チームに開示されます
Answer: (解答を表示する**)**

最新問題: **16**
Discovery を考慮すると、次の文のうち FALSE はどれですか?
A. Discovery は、モジュールの命名規則に従って、モジュールをアーキテクチャ キャンバス レイヤーに自動的に割り当てます。
B. Discovery はモジュールをドメインに分類できません
C. Discovery はモジュール レベルでのアーキテクチャ ルールのみを検証します。

D. Discovery を使用すると、モジュールごとに、どの要素が他のモジュールによって消費されているかを判断できます。
Answer: (解答を表示する**)**

有効な **Architecture-Specialist-11** 問題集は GoShiken.com が提供された合格しやすい Architecture-Specialist-11 試験問題集！ GoShiken.com が最新の **Architecture-Specialist-11** 試験問題集を提供しています。GoShiken.com Architecture-Specialist-11 試験問題は最新で、解答が正確でございます。最新の GoShiken.com Architecture-Specialist-11 問題集をゲットする人はこちら:
https://www.goshiken.com/OutSystems/Architecture-Specialist-11-mondaishu.html
(**8530%OFF**問題集溶と正解付きで **30%**ｗ特別割引コード: **Freepdfdumps**)

最新問題: **17**
「スパゲッティ アーキテクチャ」は、「密結合アーキテクチャ」または「脆性アーキテクチャ」とも呼ばれます。「スパゲッティアーキテクチャ」が悪い理由にならないのはどれですか?
A. あらゆる変更は保守可能で、デプロイが簡単です。
B. コンポーネント内の 1 つの小さな変更により、他のコンポーネントに一連の連鎖的な影響が引き起こされます。
C. 柔軟性が低下し、管理が困難になるため、将来の変更を妨げます。
Answer: A (メッセージを残す**)**

最新問題: **18**
以下のうち、コア モジュール パターン - コア エンティティ パターンに最も一致するものはどれですか...
A. ... データが複数の外部システムから送信されている場合に必要です。IS はデータに応じて使用するドライバーを決定します。
B. ... エンティティは読み取り専用として公開されており、API を使用してエンティティの作成/更新のビジネス ロジックを一元化できます。
C. ... エンティティはOutsystemsではなく、外部ERPシステムにあります。IS は外部システム/データベースへのリモート呼び出しを行うだけです。OS内にデータは保持されません。情報を取得するには 2 つの異なるシステムを横断する必要があるため、データ取得が最適化されない可能性があります。短所: 統合 API はすべてのユースケースをサポートする必要がある
D. ... 頻繁にリスト化、結合、または検索される概要データのみをキャッシュします。単一エントリの完全な詳細は、外部システムから直接取得されます。データベース全体が大きすぎるか、同期するにはコストがかかる場合に使用します。詳細は単一のエンティティにのみ必要です (リストではありません)

E. ... 外部ライブラリ内のコードを公開したり、外部データベースを検査してデータ構造をインポートして OS 内のエンティティとして使用できるようにするロジック、アクション、データを含めるために使用されるラッパー
F. 2 つのモジュールを含むパターンです。1 つは入出力構造を持つ外部 API をカプセル化するために使用できるコネクタ モジュール、もう 1 つは正規化された API をコンシューマに公開するためのラッパー モジュールです。
G. ローカル レプリカを使用する ECS と同じですが、API モジュールが提供されます。したがって、外部システムへの変更は OS に通知され、OS は ERP システム (サブスクリプション システム) からアップデートを取得します。
H. ローカル レプリカを使用する ECS と同じですが、同期ロジックが分離されています。長所: コードの独立性。CS のコンシューマは同期の影響を受けません。同期は複数の CS を調整できます
I. ... ローカル キャッシュ エンティティからデータをフェッチしようとしますが、そこにない場合は、外部システムから 1 つのエントリを取得します。そのレコードのみをキャッシュします (リードスルー キャッシュ) データベース全体が大きすぎるか、同期するにはコストがかかりすぎる場合に使用します。統合はデータベースのほんの一部にのみ影響します。データのリストに事前にアクセスする必要がある場合は避ける
J. 基本 ECS パターンと同じですが、ローカル レプリカがあります。ローカル キャッシュとして機能するデータを保存します。長所: エンティティの使用を活用し、よりシンプルな統合 API。短所: ソースシステムへの影響が少ない
Answer: (解答を表示する**)**

最新問題**: 19**
このステートメントに最もよく一致するものは次のうちどれですか: 「データが複数の外部システムから送信されている場合に必要です。IS はデータに応じて使用するドライバーを決定します。」
A. 基本 ECS パターン
B. 直接統合された ECS
C. ライブラリ : コネクタ パターン
D. ECS 遅延ロードのバリエーション
E. 分離された同期ロジックを備えた ECS
F. ESB を介したパブリッシュ/サブスクライブを備えた ECS
G. 透明性サービス パターン 1 ウェイ (簡略化)
H. ライブラリ : 拡張パターン
I. 透過性サービス パターン
J. ECS 概要キャッシュのみのバリエーション
Answer: I (メーッセージを残す**)**

最新問題**: 20**
アーキテクチャ キャンバスのどこに API モジュールを配置する必要がありますか?

A. 基礎層
B. コア層
C. エンドユーザー層
Answer: (解答を表示する**)**

最新問題**: 21**
CSモジュールは、
A. BL は、複雑な計算を実行する場合、計算エンジンになります (請求書計算エンジンや保険シミュレーターなど)。通常、エンジンはバージョンの影響を受けます。
B. パブリック エンティティ、アクション、ブロックを含む再利用可能なコア サービス。
C. 複雑さ、構成を管理したり、独自のライフサイクルを持つための、分離されたビジネス ロジック (アクション) またはコア ウィジェット (ブロック)。
D. CS 内のデータを外部システムと同期するロジック。このロジックを分離すると、CS はシステムに完全に依存せず、外部システムの切り離しや置き換えが容易になります。
E. API を外部コンシューマに公開するテクニカル ラッパー。コア サービス システムに依存せず、複数のバージョンの API をサポートします。
Answer: (解答を表示する**)**

最新問題**: 22**
アーキテクチャ ダッシュボードの新しい名前は何ですか?
A. AI メンター
B. アーキテクチャ キャンバス
C. アーキテクチャ フレームワーク
D. アーキテクチャの検証
Answer: (解答を表示する**)**

最新問題**: 23**
アーキテクチャ検証ルールの一部ではないものは次のうちどれですか?
A. 所有者を混合しないでください
B. アプリケーションを正しく階層化する
C. エンティティを正しくレイヤー化する
D. スポンサーを混合しないでください: 事業内容に基づいてアプリを分割します
E. モジュールを正しく階層化する
Answer: C (メーッセージを残す**)**

最新問題**: 24**
モジュールの命名規則を採用する理由にならないものは次のうちどれですか?
A. 各モジュールの性質を明らかにする
B. リファレンス アーキテクチャを適用します。
C. 既知のパターンを正規化します

D. 正しいレイヤーに属していることを確認してください
Answer: D (メーッセージを残す**)**

最新問題**: 25**
SSL ピンニングは、モバイル アプリケーションのセキュリティのベスト プラクティスです。SSL ピンニングに関して誤っているのは次のうちどれですか?
A. SSL ピンニングは、公開キーをコードにハードコーディングすることを意味するものではありません。
B. 上記はすべて true
C. SSL ピン留めまたは HTTP 公開キー ピン留め (HPKP) は、中間者攻撃の軽減に役立ちます
D. クライアント側で動作し、モバイル アプリに事前にバンドルされている公開キーのハッシュに対するサーバー証明書の検証を追加します。
Answer: A (メーッセージを残す**)**

最新問題**: 26**
モバイル アプリケーション アーキテクチャのベスト プラクティスではないものは何ですか。サーバー側の呼び出しを制限しますか?
A. 粒度を高めるために、すべての画面またはオンライン イベントで同期します。
バックエンド システムへの直接呼び出しはありません。Outsystems Core Servicesを呼び出すと、Outsystemsが外部バックエンドシステムとの対話を処理します。
B. サーバー側の呼び出しをオンライン トランザクションまたは同期に制限します。情報をサーバー側に渡す前に、クライアント側で大部分の操作を実行します。
C. セキュリティ、パフォーマンス、保守性の観点から、直接呼び出しは好ましくありません。直接呼び出しには JavaScript コードが必要ですが、JavaScript コードの保守は難しくなります。
Answer: (解答を表示する**)**

最新問題**: 27**
_API モジュールは次のものです
A. API を外部コンシューマーに公開するテクニカル ラッパー。コア サービス システムに依存せず、複数のバージョンの API をサポートします。
B. CS 内のデータを外部システムと同期するロジック。このロジックを分離すると、CS はシステムに完全に依存せず、外部システムの切り離しや置き換えが容易になります。
C. パブリック エンティティ、アクション、ブロックを含む再利用可能なコア サービス。
D. 複雑さ、構成を管理するため、または独自のライフサイクルを持つための、分離されたビジネス ロジック (アクション) またはコア ウィジェット (ブロック)。
E. BL は、複雑な計算を実行する場合、計算エンジンになります (請求書計算エンジンや保険シミュレーターなど)。通常、エンジンはバージョンの影響を受けます。
Answer: (解答を表示する**)**

Valid Architecture-Specialist-11 Dumps shared by GoShiken.com for Helping Passing Architecture-Specialist-11 Exam! GoShiken.com now offer the **newest Architecture-Specialist-11 exam dumps**, the GoShiken.com Architecture-Specialist-11 exam **questions have been updated** and **answers have been corrected** get the **newest** GoShiken.com Architecture-Specialist-11 dumps with Test Engine here: https://www.goshiken.com/OutSystems/Architecture-Specialist-11-mondaishu.html (**85** Q&As Dumps, **30%OFF** **Special Discount:** **Freepdfdumps**)