

# Adobe.AD0-E722.v2024-03-14.q17

|                                                                                                                                                         |                                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| 試験コード:                                                                                                                                                  | AD0-E722                        |
| 試験名称:                                                                                                                                                   | Adobe Commerce Architect Master |
| 認定資格:                                                                                                                                                   | Adobe                           |
| 無料問題数:                                                                                                                                                  | 17                              |
| バージョン:                                                                                                                                                  | v2024-03-14                     |
| アクセス数:                                                                                                                                                  | 202                             |
| ページビュー数:                                                                                                                                                | 170                             |
| <a href="https://www.jpnpdf.com/Adobe.AD0-E722.v2024-03-14.q17-mondaishu.html">https://www.jpnpdf.com/Adobe.AD0-E722.v2024-03-14.q17-mondaishu.html</a> |                                 |

## 最新問題: 1

Adobe Commerce Architect は、販売者の Adobe Commerce ストアフロント用にオーダーメイドのニュース セクションをスコープする必要があります。販売者の SEO 代理店は、次の URL 構造を要求します。

news/{date}/{article\_url\_key}。ここで、{date} は記事の公開日、{article\_url\_key} は記事の URL キーです。

アーキテクトは、ニュース エンティティ タイプが作成される範囲を指定します。日付と URL キーのデータは各レコードに対して保存され、保存時に自動生成されます。値は手動で上書きできるようになります。

- A. アーキテクトは、この機能のルーティングを管理し、ベスト プラクティスに従う必要があります。これらの要件を満たすためにアーキテクトが考慮すべき 2 つのオプションはどれですか? (2 つお選びください。)
- B. 標準コントローラ ルートを作成し、内部 URL (news/article/view/id/i など) をリライトにマッピングします。リライトは保存時に生成され、URL リライト テーブルに保存されます。
- C. 標準ルーターの前に実行され、URL のニュース部分と一致するカスタム ルーターを作成し、URL の日付と URL キー部分を一致させてニュース記事を検索してロードします。
- D. Magento\Framework\App\Action: :() をインターセプトするプラグインを作成し、URL のニュース部分を検索し、一致する場合は、URL の日付と URL キーを照合して関連するニュース記事をロードします。部品。
- E. 標準コントローラ ルートと、URL の日付と URL のキー部分を照合して関連するニュース記事を読み込むインデックス/インデックス コントローラ クラスを作成します。

**Answer: B,C (メッセージを残す)**

## 説明

これら 2 つのオプションはどちらも、オーダーメイドのニュース セクションのルーティングを管理し、ベスト プラクティスに従うための有効な方法です。オプション B では、Adobe Commerce の既存の URL 書き換え機能を活用し、任意のエンティティ タイプのカスタム URL を作成して

データベースに保存できます。このオプションでは、news/article/view/id/i などのニュース エンティティ タイプの標準コントローラ ルートを作成する必要があります。i はニュース記事 ID です。

次に、各ニュース記事を保存すると、内部 URL を必要な SEO フレンドリーな URL (news/{date}/{article\_url\_key} など) にマップする書き換えルールが生成されます。書き換えルールは url\_rewrite テーブルに保存され、標準ルーターが要求を照合してリダイレクトするために使用します。

オプション C では、\Magento\Framework\App\RouterInterface を実装し、ルーティング プロセスで標準ルーターの前に実行するカスタム ルーター クラスを作成します。カスタム ルーター クラスは、URL のニュース部分と照合し、そこから日付と URL のキー部分を抽出できます。次に、モデルまたはリポジトリ クラスを使用して、それらの値に一致するニュース記事を検索してロードできます。一致するものが見つかった場合は、それに応じてリクエスト パラメータを設定し、ニュース記事ページをレンダリングするコントローラ アクションにリクエストをディスパッチできます。

参考文献:

ルーティング | Adobe Commerce 開発者ガイド

URL の書き換え | Adobe Commerce 開発者ガイド

カスタムルーター | Adobe Commerce 開発者ガイド

## 最新問題: 2

Adobe Commerce Architect は、stopwordsjtJT.csv という名前のイタリア語ロケールのストップワードを作成し、ストップワード ディレクトリを次の場所に変更します:

<magento\_root>/app/code/Customvendor/Elasticsearch/etc/stopwords/ ストップワード ディレクトリを変更する正しい方法は何ですか?カスタムモジュールの中では?

- A. \Magento\Elasticsearch\SearchAdapter\Query\Preprocessor\Stopwords ClfSS の stopwordsDirectory にストップワードを追加し、ストップワード sModule パラメータに CustomerVendor\_Elasticsearch を追加します。di.xml 経由
- B. \Magento\Framework\Setup\Patch\PatchInterface を実装する新しい ClaSS を追加して、core.conf ig\_data テーブルの elasticsearch\customer\stopwordspath のデフォルト値を変更します。
- C. stopwords/it.xml 経由で \Hagento\Elasticsearch\Model\Adapter\Document\DirectoryBuilder ClaSS の stopwordsDirectory パラメータにストップワードを追加すると、Adobe Commerce が現在のモジュールを自動的に検出します。

**Answer: A** ([メッセージを残す](#))

説明

Adobe Commerce のドキュメントによると、カスタム モジュール内のストップワード ディレクトリを変更する正しい方法は、依存関係の注入を使用して、\Magento\Elasticsearch\SearchAdapter\Query\Preprocessor\Stopwords クラス。stopwordsDirectory パラメータはモジュール ディレクトリからのストップワード ディレクトリの相対パスを指定し、stopwordsModule パラメータはストップワード ディレクトリを含むモ

ジュールの名前を指定します。これらのパラメーターをカスタム モジュールの di.xml ファイルに追加することで、アーキテクトはコア コードやデータベースを変更せずにストップワード ファイルの場所を変更できます。

参考文献:

モジュールからディレクトリを変更するには  
Elasticsearch ストップワードを構成する

### 最新問題: 3

クライアントは Adobe Commerce Cloud に移行中で、実装する必要がある既存のリダイレクトが約 800 あります。すべてのリダイレクトは固有であり、どのパターンにも一致しないため、リダイレクトの数を減らすことはできません。

パフォーマンスを確保するにはリダイレクトをどのように構成する必要がありますか?

- A. 各リダイレクトを magento/routes.yaml ファイルに追加します。
- B. VCL スニペットを使用してリダイレクトを Fastly にオフロードします。
- C. 管理 UI を介して、各リダイレクトを URL 書き換えとして追加します。

**Answer: B (メッセージを残す)**

説明

選択肢 B は正解です。VCL スニペットを使用してリダイレクトを Fastly にオフロードすることが、リダイレクトを構成しパフォーマンスを確保する最良の方法であるためです。VCL スニペットは、Fastly 構成に追加してキャッシュ サービスの動作を変更できるカスタム コード セグメントです。VCL スニペットを使用すると、Magento アプリケーションやデータベースに到達することなく、エッジサーバー レベルでリダイレクトを処理できます。これにより、サーバーの負荷が軽減され、リダイレクトされたリクエストの応答時間が短縮されます<sup>1</sup>。

magento/routes.yaml ファイルに各リダイレクトを追加することは、リダイレクトの構成に推奨される方法ではないため、オプション A は不正解です。magento/routes.yaml ファイルは、ドメインまたはサブドメインを環境やサービスにマッピングするなど、Magento Cloud プロジェクトのカスタム ルートを定義するために使用されます。このファイルにリダイレクトを追加すると、既存のルートと競合が発生し、プロジェクトのルーティング ロジックに影響を与える可能性があります<sup>2</sup>。

オプション C は不正解です。管理 UI を介して URL リライトとして各リダイレクトを追加することは、リダイレクトを構成する最適な方法ではありません。Magento の URL 書き換え機能を使用すると、製品、カテゴリ、CMS ページのカスタム URL を作成し、それらを正規 URL にリダイレクトできます。ただし、多数の URL 書き換えを追加すると、データベースのサイズが増加し、Magento アプリケーションのパフォーマンスに影響を与える可能性があります。さらに、このタスクに管理 UI を使用するのには面倒でエラーが発生しやすい可能性があります<sup>3</sup>。

参考文献:

- 1: カスタム VCL スニペット | Adobe Commerce 開発者ガイド
- 2: ルートを構成する | Adobe Commerce 開発者ガイド
- 3: URL の書き換え | Adobe Commerce ユーザーガイド

#### 最新問題: 4

Adobe Commerce Architect は、構成可能な製品をカートに追加するプロセスを変更するための新しい GraphQL API 変異を作成しています。ミューテーションでは、構成可能な製品 ID を受け入れます。指定された製品にバリエーションが 1 つだけある場合、ミューテーションはこのバリエーションをカートに追加し、null 非許容の Carttype を返す必要があります。構成可能な製品にさらに多くのバリエーションがある場合、ミューテーションは null 非許容の ConfigurableProduct タイプを返す必要があります。

突然変異の宣言は次のようになります。

Adobe Commerce Architect はこの突然変異の出力をどのように宣言する必要がありますか？

A.

B.

C.

**Answer:** ([解答を表示する](#))

説明

Adobe Commerce のドキュメントによると、GraphQL ミューテーションの出力は、ミューテーションによって返されるデータのタイプを指定することによって宣言されます。型は、スカラー型 (String、Int、Boolean など)、オブジェクト型 (Cart、Product、Customer など)、または共用体型 (SearchResult など) のいずれかになります。製品またはカテゴリ)。共用体型は、入力または突然変異のロジックに応じて、突然変異が複数の可能なデータ型を返す可能性がある場合に使用されます。この場合、変更は、構成可能な製品のバリエーションの数に応じて、Cart タイプまたは ConfigurableProduct タイプのいずれかを返すことができます。したがって、突然変異の出力は、Cart 型と ConfigurableProduct 型の両方を含む共用体型として宣言する必要があります。オプション B は、パイプ記号 (|) を使用して可能な型を区切って共用体型を正しく宣言する唯一のオプションです。オプション A とオプション C は、それぞれリスト型と入力オブジェクト型の宣言に使用される括弧 ([ ]) と中括弧 ({ }) を使用しているため、正しくありません。

参考文献:

GraphQL API - Adobe Inc.

GraphQL を使用したスキーマ言語 | アドビコマース

#### 最新問題: 5

アーキテクトは、次のことを行う統合テストを作成したいと考えています。

- \* データフィクスチャを使用して製品を追加します
- \* 製品に対して `$this->someLogic->execute($product)` を実行します
- \* 結果が true かどうかを確認します。

`$this->someLogic` には、`setup()` メソッドで正しいオブジェクトが割り当てられています。

製品の作成とテストされたロジックは、作成されテストに使用できる ID が 3 と 4 の 2 つの異なるストア ビューのコンテキストで実行する必要があります。

アーキテクトはこれらの要件をどのように満たすべきでしょうか？

A. それぞれ 1 つのテスト メソッドを持つ 2 つのテスト クラスを作成します。@magentoExecuteInStoreContext 3 を使用し、\$MagentoExecuteInStoreContext クラス レベルの 4 アノテーション。

B. 2 つのテスト メソッドを含む 1 つのテスト クラスを作成します。一方のメソッドでは emagentstorecontext 3 アノテーションを使用し、もう一方のメソッドでは amagentstorecontext 4 を使用します。

C. 1 つのテスト メソッドを持つ 1 つのテスト クラスを作成します。使用 \Magento\TestFramework\store\ExecuteInStoreContext クラスをフィクスチャで 1 回、テストで もう 1 回使用します。

**Answer:** ([解答を表示する](#))

説明

さまざまなストア ビューでさまざまなロジックを実行する統合テストを作成するには、アーキテククトは次の手順を実行する必要があります。

\Magento\TestFramework\TestCase\AbstractController を拡張するテスト クラスを 1 つ作成するか、

\Magento\TestFramework\TestCase\AbstractBackendController (テスト対象のコントローラーの種類に応じて)1。

@magentoDataFixture アノテーションを使用して、product2 を作成するデータ フィクスチャ ファイルを指定するテスト メソッドを 1 つ作成します。

\Magento\TestFramework\Store\ExecuteInStoreContext クラスを使用して、フィクスチャとテストされたロジックをさまざまなストア ビューで実行します。このクラスに

は、executeInStoreContext というメソッドがあり、ストア ID と呼び出し可能な関数という 2 つのパラメータを取ります。呼び出し可能な関数は、指定されたストア ID のコンテキストで実行され、元のストア ID が復元されます3。例えば：

PHPAI で生成されたコード。慎重に確認して使用してください。詳細については FAQ をご覧ください。

パブリック関数 testSomeLogic()

```
{
// フィクスチャから製品を取得します
$product = $this->getProduct();
// オブジェクト マネージャーから ExecuteInStoreContext インスタンスを取得します
$executeInStoreContext
$this->_objectManager->get(\Magento\TestFramework\Store\ExecuteInStoreContext::class);
// ストアビュー 3 でフィクスチャを実行します
$executeInStoreContext->executeInStoreContext(3, function () use ($product) {
// ストア ビュー 3 で商品に対していくつかの操作を実行します
});
// テストされたロジックをストア ビュー 4 で実行します。
$result = $executeInStoreContext->executeInStoreContext(4, function () use ($product) {
// ストア ビュー 4 の製品でテスト済みのロジックを呼び出します
```

```
$this->someLogic->execute($product) を返します。  
});  
// 結果が true であることをアサートします  
$this->assertTrue($result);  
}
```

参考文献:

統合テスト | Magento 2 開発者向けドキュメント

データフィクスチャ | Magento 2 開発者向けドキュメント

Magento\TestFramework\Store\ExecuteInStoreContext | Magento 2 開発者向けドキュメント

## 最新問題: 6

すべてのキャッシュタイプを有効にして開発者モードで Web サイトの新機能を開発する際、Adobe Commerce 開発者は、`\Magento\Sales\Model\Service\InvoiceService` `sinvoiceService` を新しい依存関係として `Vendor\CustomModule\` の既存のページアクションコントローラーに追加する必要があります。コントローラー `\インデックス\インデックス`。これは次のようにして実現されます。

`full_page` キャッシュを消去してページを再ロードした後、開発者は次の例外に遭遇します。

回復可能なエラー: `Vendor\CustomModule\Controller\Index\Index::__construct()` に渡される引数 2 は、次のインスタンスである必要があります。

`\Magento\Sales\Model\Service\InvoiceService [...]`

このエラーを修正するために、アーキテクトは開発者にどのアクションを推奨する必要がありますか？

- A. `full_page` キャッシュとともに `block_html` キャッシュを消去します。
- B. 新しい `\Magento\Sales\Model\Service\InvoiceService` `sinvoiceservice` 依存関係をコンストラクター署名の最後に追加します。
- C. 生成された子クラスを `generated/code/Vendor/CustomModule/Controller/Index/Index` から削除します。

**Answer: C (メッセージを残す)**

説明

このエラーは、生成された子クラスが新しい依存関係で更新されていないことが原因で発生します。生成された子クラスを削除すると、システムは正しい依存関係を持つ新しい子クラスを生成できるようになります。生成された子クラスは、元のコントローラー クラスを拡張し、コンストラクターをオーバーライドして、オブジェクト マネージャーを使用して依存関係を注入するプロキシクラスです。生成された子クラスは、パフォーマンスの問題を回避するために、キャッシュが有効になっている開発者モードでシステムが実行されるときに作成されます。ただし、新しい依存関係が元のコントローラー クラスに追加されると、生成された子クラスには変更が反映されず、コンストラクター引数に不一致が発生します。したがって、生成された子クラスを `generated/code` ディレクトリから削除すると、問題は解決します。

参考文献:

生成されたコード | Adobe Commerce 開発者ガイド

### 最新問題: 7

Adobe Commerce Architect は、既存の Adobe Commerce Web サイトに新しい複合製品タイプを導入する機能を設計および実装します。新しい製品タイプの視覚的なデモンストレーションに加えて、変更には価格指数の調整が含まれます。

この Web サイトでは、多次元インデクサー機能を利用して価格インデックスを保存しています。アーキテクトは、統合テストでこれをカバーすることにしました。1 つのテストを作成して実行した後、アーキテクトはデータベース ストレージが完全にクリーンアップされていないことを発見します。

テスト メソッドには次のアノテーション宣言があります。

この問題を解決するには、アーキテクトはどの調整を行う必要がありますか？

A. メソッド PHPDoc にアノテーション `@magentoAppIsolation` を追加します。

B. メソッド PHPDoc を変更し、アノテーション `@magentoDbIsolation` を有効に変更します。

C. フィクスチャごとに `Customer_ProductType: :Test/_files/{fixture_name}_rollback.php` を作成します。

**Answer: B** ([メッセージを残す](#))

#### 説明

ここでの問題は、テストの実行後にデータベース ストレージが完全にクリーンアップされていないことです。解決策は、メソッド PHPDoc を変更し、アノテーション `@magentoDbIsolation` を有効に変更することです。これにより、テストの実行後にデータベース ストレージが完全にクリーンアップされます。参考文献:

<https://developer.adobe.com/commerce/testing/guide/integration/#database-isolation1>

### 最新問題: 8

既存の Adobe Commerce Web サイトはヘッドレス実装に移行しています。

既存の Web サイトには、「すべてのブランド」ページと各ブランドの個別ページがあり、すべてのブランド関連ページは、製品やカテゴリと同じようにタグを使用して Varnish にキャッシュされます。

この情報を新しいヘッドレス実装のフロントエンドで利用できるようにするために、2 つの新しい GraphQL クエリが作成されました。

テスト中に、クエリによって古い情報が返されることがあります。パフォーマンスを維持しながらこの問題をどのように解決すべきでしょうか？

A. GraphQL クエリごとに `@cache(cacheable: false)` ディレクティブを指定し、返されたデータがキャッシュされておらず、最新であることを確認します。

B. 関連するブランドおよび関連製品のキャッシュ タグを追加するクラスに対応する、GraphQL クエリごとに `$cache(cacheidentity: Path\To\identityclass)` ディレクティブを指定します。

C. 各 GraphQL クエリのリゾルバー クラスは、リゾルバーの解決関数の一部として \Magento\GraphQLcache\Model\cacheableQuery を挿入し、それに対して setcachevalidity(true) を呼び出す必要があります。

**Answer:** ([解答を表示する](#))

説明

このソリューションは、パフォーマンスを維持しながら、GraphQL クエリによって返されるデータが最新であることを保証します。GraphQL クエリごとに \$cache(cacheidentity: Path\To\identityclass) ディレクティブを指定すると、関連するブランドと関連製品がキャッシュ タグとして追加されます。

**最新問題: 9**

Adobe Commerce Architect は、製品価格インデックスの実行に時間がかかりすぎることに気づきました。ストアは複数の Web サイトと数十の顧客グループで構成されています。

アーキテクトが全価格インデックスの実行時間を短縮できる 2 つの方法はどれですか? (2つお選びください。)

- A. `mage_indexer_threads_COUNT` 環境変数を設定して並列モードを有効にします
- B. `catalog_Price_index` インデクサーを別のカスタム インデクサー グループに移動します
- C. 階層価格のない製品の価格インデックス顧客グループの結合を有効にする
- D. 顧客共有顧客アカウント オプションをグローバルに設定します
- E. 顧客グループを編集して、使用していない Web サイトを除外します

**Answer:** A,C ([メッセージを残す](#))

説明

製品価格指数は、パラレル モードと顧客グループの結合を使用して最適化できます。パラレルモードでは、インデクサーが複数のスレッドを同時に実行できるため、インデックス作成プロセスが高速化されます。顧客グループの結合では、同じ製品価格を持つ顧客グループを結合することにより、価格インデックス テーブルの行数が削減されます。これにより、価格インデックス クエリのパフォーマンスが向上し、インデックス サイズが削減されます。参考資料: インデックスの最適化、価格インデックスの顧客グループの統合

**最新問題: 10**

ある企業は、自国の顧客に製品を販売するために Adobe Commerce Web サイトを構築したいと考えています。

各国の税金は非常に複雑なので、Adobe Commerce へのカスタマイズが必要です。アーキテクトは、Adobe Commerce のすべての注文の税金の計算を処理するカスタム税金計算ツールを作成することで、この問題を解決しようとしています。

ベスト プラクティスに従って、アーキテクトはすべての注文に税金をどのように追加する必要がありますか?

- A. イベント `sales.quote.collectTotals.before` に新しいオブザーバーを追加し、カスタム税を見積もりに追加します

B. 前プラグインを `\Magento\Quote\Model\QuoteManagement::placeOrder()` に書き込み、カスタム税を見積もりに追加します

C. カスタム モジュールの `etc/sales.xml` で新しい合計コレクターを宣言します。

**Answer:** ([解答を表示する](#))

説明

Adobe Commerce のドキュメントによると、すべての注文にカスタム税計算を追加する最良の方法は、カスタム モジュールの `etc/sales.xml` ファイルで新しい合計コレクターを宣言することです。このようにして、カスタム税金ロジックを拡張する別のクラスに実装できます。

`\Magento\Quote\Model\Quote\Address\Total\AbstractTotal` クラスであり、`collect()` メソッドと `fetch()` メソッドをオーバーライドします。`collect()` メソッドは税額を計算して見積アドレスに追加する役割を果たし、一方 `fetch()` メソッドはカートとチェックアウト ページに税額を表示する役割を果たします。新しい合計コレクターは、小計、送料、総計の前後など、注文合計の任意の領域に割り当てることができます。

参考文献:

注文合計のカスタマイズ

Magento 2 で注文合計にカスタム料金または割引を追加する方法

最新問題: 11

アーキテクトは、シンボリックリンク ディレクトリ `/var/www/current` で動作するように構成されているサーバーに関する展開の問題を調査しています。これにより、アプリケーションの最新リリース バージョンが発生します。

展開プロセスでは次の手順が実行されます。

前回の導入後、顧客のメイン ファイル データをエクスポートする Adobe Commerce インポート/エクスポート機能が機能しないと販売者から報告がありました。アーキテクトは、生成されたファイルのリストにエクスポート ファイルが表示されていないことを発見しました。

この問題を解決するには、展開プロセスに対してどの変更を実行する必要がありますか？

A. デプロイ中にコンシューマ プロセスを再起動して、エクスポート ファイル用の新しいアプリケーション バージョンのディレクトリを使用します。

B. コマンド `config:set export/customr/files_directory /var/releases/{release_number}` を実行して、新しいエクスポート パスを設定します。

C. デプロイ前に Crontab を実行し、デプロイ後に再起動できます。

**Answer: B** ([メッセージを残す](#))

説明

問題は、エクスポート ファイルが生成されたファイルのリストに表示されないことです。これは、エクスポート パスが正しく設定されていないことが原因です。解決策は、コマンド `config:set export/customr/files_directory` を実行することです。

`/var/releases/{release_number}` を使用して、新しいエクスポート パスを設定します。これにより、エクスポート ファイルが正しいディレクトリに保存され、管理パネルからアクセスできるようになります。参考文献:

[https:// experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/develop/deploy/staging-production](https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/develop/deploy/staging-production)

### 最新問題: 12

Adobe Commerce Architect は、テスト チームに渡される前に、開発者がパフォーマンスではなく機能を特にテストするために使用されるオンプレミス プロジェクト用の開発環境をセットアップしています。

Magento アプリケーションは次の要件で実行する必要があります。

1. エラーはログに記録され、ユーザーから隠される必要があります。
  2. キャッシュ モードはコマンド ラインからのみ変更できます。
  3. 静的ファイルは動的に作成してキャッシュする必要があります。
- これを実現するにはどのアプリケーション モードが必要ですか？

- A. デフォルトモード
- B. 本番モード
- C. 開発者モード

**Answer:** ([解答を表示する](#))

#### 説明

開発者モードは、テスト チームに渡される前に、パフォーマンスではなく機能をテストするための開発環境をセットアップするための最良のオプションです。開発者モードの場合:

エラーはログに記録され、ユーザーには表示されません。これにより、ユーザーには捕捉されなかった例外やデバッグ情報が表示されなくなりますが、開発者はログ ファイルからそれらにアクセスできます。

キャッシュ モードはコマンド ラインからのみ変更できます。これにより、管理パネルやその他のソースからキャッシュ設定が誤って変更されたり、不正に変更されたりすることがなくなります。静的ファイルは動的に作成され、キャッシュされます。これにより、開発者は静的コンテンツ展開 コマンドを毎回実行しなくても、静的ファイルに対する最新の変更を確認できるようになります。静的ファイルもキャッシュされ、読み込みが高速化されます。

参考文献:

[https:// experienceleague.adobe.com/docs/commerce-operations/configuration-guide/setup/application-modes.htm](https://experienceleague.adobe.com/docs/commerce-operations/configuration-guide/setup/application-modes.htm)

### 最新問題: 13

中小企業の代表者は、サードパーティの支払いソリューションのカスタム統合を設計するために Adobe Commerce Architect を必要としています。彼らは、既存の Magento アプリケーションの PCI 準拠を達成するために、自己評価アンケートで特定されたコントロールのリストをできる限り減らしたいと考えています。

ビジネス ニーズを満たすアプローチはどれですか？

- A. Advanced Encryption 標準 (aes-256) アルゴリズムを利用して、支払いモジュールからの顧客の機密データをすべて暗号化します。

B. 決済プロバイダーの iframe システムを利用して、埋め込みフレームのコンテンツを親 Web ページから分離します。

C. 認証局 (CA) によって発行された信頼できる署名付き証明書を利用して、https 経由の支払いソリューション プロトコルによって行われる各接続を保護します。

**Answer: B (メッセージを残す)**

説明

支払い統合に iframe システムを使用すると、加盟店の Web サイトやサーバーに触れることなく、支払いデータが iframe 内の決済サービス プロバイダー (PSP) によって収集および処理されるため、加盟店の PCI 範囲とコンプライアンスの負担を軽減できます。このようにして、加盟店は PSP の PCI 認証を活用し、カード所有者の機密データを自社のシステムに保存したり送信したりすることを回避できます。また、iframe はホスト Web ページと読み込まれたページの間に安全なバリアを提供し、悪意のある行為者による支払いデータへのアクセスや操作を防ぎます。このアプローチを実装するには、販売者は iframe 要素を使用してチェックアウト ページに PSP の支払いフォームを埋め込み、JavaScript123 を使用して iframe とホスト ページ間の通信を構成する必要があります。

最新問題: 14

カスタム モジュールで、アーキテクトは新しい XML 構成ファイルを定義したいと考えています。モジュールは、システムで宣言されたすべての XML 構成ファイルを読み取り、それらをマージし、その値を PHP クラスで使用できる必要があります。

この要件を満たすためにアーキテクトは次の 2 つの手順を実行する必要がありますか? (2つお選びください。)

A. di.xml に `Magento\Framework\Config\Data` の「リーダー」依存関係を挿入します。

B. `Magento\Framework\Config\Data::get()` のプラグインを作成し、カスタム XML ファイルを読み取ります

C. `"Magento\Framework\Config\Data"` を実装する Data クラスを作成します。

D. di.xml の `Magento\Config\Model\Config\Structure\Reader` にカスタム XML ファイル名を追加します。

E. `Magento\Framework\Config\Reader\Filesystem` を実装する Reader クラスを作成します。

**Answer: C,E (メッセージを残す)**

説明

Adobe Commerce のドキュメントによると、新しい XML 構成ファイルを作成するには、アーキテクトはカスタム モジュールの Data クラスと Reader クラスを作成する必要があります。Data クラスは、キャッシュまたは Reader クラスからの構成データの保存と取得を担当します。Data クラスは以下を実装する必要があります。

`Magento\Framework\Config\Data` インターフェイス、または `Magento\Framework\Config\Data` クラスを拡張します。Reader クラスは、ファイル システムから XML 構成ファイルを読み取り、検証する役割を果たします。Reader クラスは、`Magento\Framework\Config\Reader\Filesystem` インターフェイスを実装するか、

"\Magento\Framework\Config\Reader\Filesystem" クラス。アーキテクトは、カスタム モジュールの di.xml ファイルで Data クラスと Reader クラスを宣言し、XML 構成ファイル、コンバーターの名前を指定する必要もあります。クラス、および Reader クラスのスキーマ ロケータ クラス。

参考文献:

構成の種類 | Adobe Commerce - エクスペリエンス リーグ

構成タイプを作成する | Adobe Commerce - エクスペリエンス リーグ

最新問題: 15

ヘッドレス Adobe Commerce プロジェクトに取り組むアーキテクトは、my\_attribute という名前の新しい顧客属性を作成します。顧客の属性値に基づいて、GraphQL クエリの結果がプラグインを使用して変更されます。フロントエンドアプリケーションは、Varnish by Fastly を通じて Adobe Commerce と通信します。これは、変更されるクエリをすでにキャッシュしています。Adobe Commerce Fastly 拡張機能がインストールされており、アプリケーションにはその他の変更は加えられません。

Varnish の vcl\_hash 関数が新しく作成された属性も考慮するようにするには、アーキテクトはどの手順を実行する必要がありますか?

**A.** を継承して新しい Class を作成します。

\Magento\GraphQLCache\Model\Cached\CachedFactorProviderInterface を実行

し、getFactorValue 関数から my\_attribute の値を返し、getFactorName 関数から my\_attribute の値を返します。次に、このクラスを DI を介して \Magento\GraphQLCache\Model\Cached\CachedCalculator の idFactorProviders 配列に追加します。

**B.** \Magento\Framework\GraphQL\Query\Resolver\IdentityInterface を継承し、getIdentities 関数から my\_attribute の値を返す新しいクラスを作成します。

次に、各 GraphQL クエリに ecache(cacheidentity: Path\To\IdentityClass) ディレクティブを指定して、各顧客のキャッシュ タグを追加する各クエリに新しく作成された IdentityClass を含めます。

**C.** \Magento\customer\customerData\SectionSourceInterface を継承し、getSectionData 関数から my\_attribute の値を返す新しいクラスを作成します。次に、\Magento\Customer\CustomerData\SectionPoolInterface の SectionSourceMap 配列を通じて thisClass を追加します。

**Answer: A** ([メッセージを残す](#))

説明

Varnish の vcl\_hash 関数が新しく作成された属性を考慮するようにするには、アーキテクトは次の手順を実行する必要があります。

を実装する新しいクラスを作成します。

\Magento\GraphQLCache\Model\Cached\CachedFactorProviderInterface インターフェイス。このインターフェイスは、getFactorName と getFactorValue という 2 つのメソッドを定義します。getFactorName メソッドは属性の名前 (この場合は my\_attribute) を返す必要があります。

ず。getFactorValue メソッドは、現在の顧客の属性の値を返す必要があります。この属性の値は、顧客セッションまたは顧客リポジトリ 1 から取得できます。

このクラスを idFactorProviders 配列に追加します。

依存関係注入による Magento\GraphQLCache\Model\Cached

\CachedCalculator。CachedCalculator は、idFactorProviders によって提供される要素に基づいて、各 GraphQL リクエストのキャッシュ ID を生成します。この配列に新しいクラスを追加することで、アーキテクトはキャッシュ ID に my\_attribute1 の値が含まれるようにします。

次に、キャッシュ ID は Varnish によって使用され、各リクエストのキャッシュされた応答をハッシュして検索します。キャッシュ ID に my\_attribute を含めることで、アーキテクトは Varnish が customer2 の属性値に基づいて異なる応答を提供することを保証します。

Magento\_GraphQLCache モジュール | Magento 2 開発者向けドキュメント

ワニスのキャッシュ | Adobe Commerce 2.4 ユーザーガイド - Magento

### 最新問題: 16

単一の Adobe Commerce Cloud インスタンスは、異なるドメインを持つ 2 つの Web サイト (それぞれ単一のストアビューを持つ) でセットアップされます。

\* デフォルトの Web サイトは website\_one で、ストア ビューは store\_one、ドメインは storeone です。コム。

\* 2 番目の Web サイトは website\_two で、ストア ビューが store\_two、ドメインが storetwo です。コム。

magento-vars.php ファイルは、各リクエストがどの Web サイトに対して実行されるかを決定するために次のように設定されます。

新しい GraphQL 統合をテストすると、ドメインに関係なく、すべてのリクエストでデフォルトの Web サイトに関連するデータが返されました。この問題の原因は何ですか?

A. magento-vars.php ファイルは GraphQL リクエストに対して処理されないため、デフォルトの Web サイトが常に処理されます。

B. \$\_server["mage\_run\_cooe") をストアに設定する必要があり、代わりに \* \$\_SERVER["MAGE\_RUN\_TYPE"] をストア コードに設定する必要があります。

C. ストア ヘッダーまたはストア Cookie が提供されない限り、GraphQL リクエストは常にデフォルトのストア ビューに対して実行されます。

**Answer: C (メッセージを残す)**

説明

magento-vars.php ファイルは、HTTP ホストに基づいて Web サイトまたはストア ビューを設定するために使用されますが、GraphQL リクエストには影響しません。GraphQL リクエストは、magento-vars.php ファイルを使用しない別のコントローラーによって処理されます。代わりに、GraphQL リクエストは、リクエストにストア ヘッダーまたはストア Cookie が指定されていない限り、デフォルト Web サイトのデフォルトストア ビューを使用します。ストア ヘッダーまたは Cookie には、目的のストア ビューのストア コードが含まれている必要があります。たとえ

ば、website\_two からデータをクエリするには、リクエストに store:store\_two のようなヘッダー、または store=store\_two12 のような Cookie を含める必要があります。  
GraphQL の概要 | Adobe Commerce 2.4 ユーザーガイド - Magento  
Magento 2 で複数の Web サイトをセットアップする方法 - Mageplaza

有効な **AD0-E722** 問題集は GoShiken.com が提供された合格しやすい AD0-E722 試験問題集！ GoShiken.com が最新の **AD0-E722** 試験問題集を提供しています。GoShiken.com AD0-E722 試験問題は最新で、解答が正確でございます。最新の GoShiken.com AD0-E722 問題集をゲットする人はこちら: <https://www.goshiken.com/Adobe/AD0-E722-mondaishu.html>  
(**5230%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

いくつかのレポートのデータを収集するために、カスタム cron ジョブが Adobe Commerce システムに追加されました。それは crontab です。XML の構成は次のとおりです。

このジョブは大量のデータを処理し、毎晩 20 ~ 30 分間実行されます。

導入から数日以内に、サイトのサイトマップが変更されていることがわかります。新しいジョブが追加されてから、xml ファイルは更新されていません。

この問題を解決するにはどうすればよいでしょうか？

**A.** siten.aP\_generate cron ジョブのスケジュールを 30 0 \* \* \* に変更して、gather\_reporting\_data ジョブが完了した後に実行されるようにします。

**B.** 次のように指定して、レポート ジョブ用の新しい cron グループを作成します。

`<use_セパレート_プロセス>i</use_セパレート_プロセス>`

**C.** データ収集ジョブをいくつかの小さなジョブに分割し、個々のジョブが最大 5 分間実行されるようにします。

**Answer: B** ([メッセージを残す](#))

#### 説明

ここでの問題は、gather\_reporting\_data ジョブが長時間実行されており、sitemap\_generate ジョブの実行をブロックしていることです。解決策は、レポート ジョブ用の新しい cron グループを作成し、レポート ジョブが別のプロセスで実行され、sitemap\_generate ジョブをブロックしないように `<use_dependent_process>i</use_Separate_process>` を指定することです。参考文献:

[https:// experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.htm](https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.htm)

**Valid AD0-E722 Dumps** shared by GoShiken.com for Helping Passing AD0-E722 Exam!  
GoShiken.com now offer the **newest AD0-E722 exam dumps**, the GoShiken.com AD0-E722 exam questions have been updated and answers have been corrected get the newest

GoShiken.com AD0-E722 dumps with Test Engine here:

<https://www.goshiken.com/Adobe/AD0-E722-mondaishu.html> (52 Q&As Dumps, **30%OFF**

**Special Discount: Freepdfdumps**)