

Adobe.AD0-E718.v2023-10-26.q45

試験コード:	AD0-E718
試験名称:	Adobe Commerce Architect Master
認定資格:	Adobe
無料問題数:	45
バージョン:	v2023-10-26
アクセス数:	530
ページビュー数:	450
https://www.jpnpdf.com/Adobe.AD0-E718.v2023-10-26.q45-mondaishu.html	

最新問題: 1

Adobe Commerce Architect が API 機能テスト コードをレビューしています。一部のテストでは、顧客のアドレスが存在しないことを示すエラーが送信されます。

テスト コードは次のことを示します。

```
/**
 * @magentoDataFixture Magento/Customer/_files/customer_one_address.php`
 * ...
 */
public function testMyUseCasTestForCartAddress(): void
```

アーキテクトはテスト エラーを修正するためにどの手順を実行する必要がありますか？

Update the annotation to specify address_id @magentoDataFixture Magento/Customer/_files/customer_one_address.php with:

A. {"address_id": "\$address.id\$"}

Change the annotation to use @magentoApiDataFixture Magento/Customer/_files/customer_one_address.php instead of @magentoDataFixture Magento/Customer/_files/customer_one_address.php

B.

Set the annotation to use @magentoPersistDataFixture Magento/Customer/_files/customer_one_address.php instead of @magentoDataFixture Magento/Customer/_files/customer_one_address.php

C.

Answer: B (メッセージを残す)

テスト エラーは、リクエストで間違った顧客 ID とアドレス ID を使用したことが原因で発生します。正しい顧客 ID とアドレス ID は、顧客とアドレスを作成するための前のリクエストの応答から取得する必要があります。テスト コードでは、ハードコーディングされた値の代わりに \$this->customer->getId() および \$this->address->getId() を使用する必要があります。参考: <https://devdocs.magento.com/guides/v2.4/get-started/web-api-function-testing.html>

最新問題: 2

Adobe Commerce Architect は、まだ稼働していない Adobe Commerce Cloud プロジェクトの問題のトラブルシューティングを行っています。

開発者は、本番環境への移行に備えてステージング データベースを運用環境に移行します。ただし、開発者が製品インポート機能をテストすると、新しい製品はフロントエンドに表示されません。

開発者は、ワニス キャッシュがクリアされていないのではないかと疑っています。ステージングは期待どおりに機能しているようです。データベースの移行前は本番環境が稼働していました。

考えられる原因は何ですか？

A. 運用データベースの Fastly 資格情報が正しくありません。

B. Fastly キャッシュを初期化するには、運用環境でデプロイメントを実行する必要があります。

C. 運用データベース内のサイト URL はステージング インスタンスの URL であるため、更新する必要があります。

Answer: A (メッセージを残す)

運用データベースの Fastly 資格情報が正しくありません。これは、新しい製品が輸入されるときにコマースによって Varnish キャッシュをクリアできないことを意味します。Fastly 認証情報は、実稼働環境と一致するように更新する必要があります。

Adobe Commerce ヘルプセンターの「Fastly 認証情報の設定」を参照してください。参考: [https://](https://experienceleague.adobe.com/docs/commerce-operations/configuration-guide/cache/use-varnish-cache.html?lang=en1)

experienceleague.adobe.com/docs/commerce-operations/configuration-guide/cache/use-varnish-cache.html?lang=en1

<https://support.magento.com/hc/en-us/articles/360006008192-Configure-Fastly-credentials>

最新問題: 3

アーキテクトは、Adobe Commerce オンプレミスインスタンスで Redis の preload.keys を設定しています。

アーキテクトは、各フロントエンド リクエストでキャッシュ キー

eav_entity_types、GLOBAL_PLUGIN_LIST、DB_IS_UP_TO_DATE、SYSTEM_DEFAULT がロードされていることを検出します。

* フロントエンド =>page_cache の id_prefix は 061_ に設定されます。

※frontend =>default:のid_prefixは設定されていません。

* アーキテクトは Redis L2 キャッシュを有効にして構成しました。

preload.keys はどのように設定する必要がありますか？

```
'preload_keys' => [  
  '061_EAV_ENTITY_TYPES:hash',  
  '061_GLOBAL_PLUGIN_LIST:hash',  
  '061_DB_IS_UP_TO_DATE:hash',  
  '061_SYSTEM_DEFAULT:hash',  
],
```

A.

```
'preload_keys' => [  
  'EAV_ENTITY_TYPES:hash',  
  'GLOBAL_PLUGIN_LIST:hash',  
  'DB_IS_UP_TO_DATE:hash',  
  'SYSTEM_DEFAULT:hash',  
],
```

B.

```
'preload_keys' => [
    'EAV_ENTITY_TYPES',
    'GLOBAL_PLUGIN_LIST',
    'DB_IS_UP_TO_DATE',
    'SYSTEM_DEFAULT',
],
```

C.

```
'preload_keys' => [
    '061_EAV_ENTITY_TYPES',
    '061_GLOBAL_PLUGIN_LIST',
    '061_DB_IS_UP_TO_DATE',
    '061_SYSTEM_DEFAULT',
],
```

D.

Answer: D ([メッセージを残す](#))

preload.keys は、オプション D に示すように構成する必要があります。これは、プリロード機能により、Redis が eav_entity_types、GLOBAL_PLUGIN_LIST、DB_IS_UP_TO_DATE、SYSTEM_DEFAULT などのページ間で再利用されるデータをロードできるためです。プリロードキーには、キャッシュバックエンドの id_prefix と、ハッシュされたデータのバージョンを示すサフィックス ':hash' が含まれている必要があります。『Adobe Commerce 開発者ガイド』の「デフォルトのキャッシュに Redis を使用する」を参照してください¹。参考: <https://devdocs.magento.com/guides/v2.3/config-guide/redis/redis-pg-cache.html>¹

最新問題: 4

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 セクションの範囲を絞る必要があります。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。

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。)

A. 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 Index/Index インデックス コントローラ クラスを作成します。

B. controllers_front_send_response_before イベントをリスンし、URL の mm 部分を探し、一致する場合は、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ロードするオブザーバーを作成します。

C. lu^jentoXFraBeworkUppXActien::executed をインターセプトし、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 のキー部分を照合して関連するニュース記事をロードするプラグインを作成します。

D. 標準コントローラ ルートを作成し、内部 URL (news/article/view/id/1 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。

E. 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 ルーターを作成します。次に、URL の日付と URL キー部分を照合してニュース記事を検索して読み込みます。

Answer: A,E ([メッセージを残す](#))

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には、次のオプションを考慮する必要があります。

標準のコントローラ 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 Index/Index インデックス コントローラ クラスを作成します。このオプションは、標準の Magento ルーティング メカニズムを使用して ニュース リクエストを処理するためのシンプルかつ簡単な方法を作成します。Index/Index コントローラ クラスは、リクエストから日付と URL キー パラメーターを受け取り、それらを使用してデータベースからニュース記事モデルをロードします。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読み込みます。このオプションにより、\Magento\Framework\App\RouterInterface を実装するカスタム ルーター クラスを使用してニュース リクエストを処理する、より柔軟でカスタマイズ可能な方法が作成されます。カスタム ルーター クラスは、リクエスト パスがニュースで始まるかどうかを確認し、パスから日付と URL のキー部分を抽出し、それらを使用してデータベースからニュース記事モデルをロードします。

参照 :

<https://belvg.com/blog/how-to-create-custom-router-in-magento-2.html>

<https://devdocs.magento.com/guides/v2.4/extension-dev-guide/routing.html>

最新問題: 5

Adobe Commerce Web サイトの開発が完了しました。Web サイトを実稼働環境に展開する準備ができました。

アーキテクトは、ロード バランサーの背後でリクエストを処理する複数のバックエンド Web サーバーで構成される分散アーキテクチャで実行されるシステムを設計しました。

システムを展開して Web サイトに初めてアクセスした後、ユーザーはログイン後にカスタマー ダッシュボードにアクセスできません。ユーザーが正常にログインしても、Web サイトはユーザーをサインイン ページにリダイレクトし続けます。アーキテクトは、セッションが正常であると判断します。正しく保存されていません。

napp/etc/env.php\ では、セッションは次のように構成されています。

```
'session' => [
    'save' => 'redis',
    'redis' => [
        'host' => '127.0.0.1'
    ]
]
```

この問題を修正するにはアーキテクトは何をすべきでしょうか？

- A. リモートストレージ モジュールを利用してサーバー間のセッションを同期します。
- B. セッション ホスト値を共有 Redis インスタンスに更新します
- C. コマンド `config:set system/security/max_session_size_admin` を使用してセッション サイズを増やします。

Answer: ([解答を表示する](#))

最新問題: 6

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_attribure 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 を介して Adobe Commerce と通信しており、変更されるクエリはすでにキャッシュされています。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？

- Create a new class inheriting from `Magento\Framework\GraphQL\Query\Resolver\IdentityInterface` and returning the value of `my_attribute` from the `getIdentities` function. Then specify a `@cache(cacheIdentity: Path\To\IdentityClass)` directive for each GraphQL query to
- A. include the newly created `IdentityClass` to each query that adds the cache tags for each customer.
- Create a new class inheriting from `Magento\Customer\CustomerData\SectionSourceInterface` and returning the value of `my_attribute` from the `getSectionData` function. Then add this class through DI to the `sectionSourceMap` array of `Magento\Customer\CustomerData\SectionPoolInterface`.
- B.
- Create a new class inheriting from `Magento\GraphQLCache\Model\CacheId\CacheIdFactorProviderInterface` and returning the value of `my_attribute` from the `getFactorValue` function and `my_attribute` from the `getFactorName` function. Then add this class through DI to the `idFactorProviders` array of `Magento\GraphQLCache\Model\CacheId\CacheIdCalculator`.
- C.

Answer: B ([メッセージを残す](#))

最新問題: 7

Adobe Commerce Architect は、オンプレミスの Adobe Commerce プロジェクトの導入と構築ツールをサポートします。このツールは、集中サーバー上でビルドスクリプトを実行し、SSH 接続を使用してプロジェクトサーバーに展開します。クライアントは、ユーザーがインターフェイスロケールを変更するたびにサイトが中断されるため、管理パネルで作業できないと報告しました。

保守性を考慮すると、アーキテクトはどのソリューションを実装する必要がありますか？

- A. プロジェクトの `env.php` ファイルを編集し、`admin_locales_for.build` 値を構成し、必要なロケールをすべて指定します。
- B. ツールのビルドスクリプトを調整し、`setup:static-content:deploy` コマンド中に必要なロケールを指定します。
- C. プロジェクトの `config.php` ファイルを変更し、`admin_locales_for_deploy` 値を構成し、必要なロケールをすべて指定します。

Answer: ([解答を表示する](#))

プロジェクトの `env.php` ファイルを編集し、`admin_locales_for.build` 値を構成し、必要なロケールをすべて指定することが、アーキテクトが実装する必要がある解決策です。これは、この構成により、ツールがビルド段階でさまざまなロケールの静的コンテンツを生成できるため、パフォーマンスが向上し、インターフェイスロケールの変更時にサイトが中断されるのを回避できるためです。Adobe Commerce ヘルプセンターの「静的ビューファイルのデプロイ」を参照してください¹。参照: <https://experienceleague.adobe.com/docs/commerce-operations/configuration-guide/cli/static-view/static-view-file-deployment.html?lang=en1>

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ます。すべてのブランド関連ページは、製品やカテゴリと同じ方法でタグを使用して 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。

```

type Query {
  brands(
    search: String! @doc(description: "Search against brand names (partial matches)")
    pageSize: Int = 20 @doc(description: "Number of brand results to return")
    currentPage: Int = 1 @doc(description: "Page number to return")
  ) : BrandsResult
  @resolver(class: "ClientName\\ProductBrandsGraphQL\\Model\\Resolver\\Brands")
  brand(
    urlKey: String! @doc(description: "Match against brand url key")
  ) : Brand
  @resolver(class: "ClientName\\ProductBrandsGraphQL\\Model\\Resolver\\BrandByUrlKey")
}

```

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. 各 GraphQL クエリのリゾルバー クラスは、MagentoGraphQLCacheModelCacheableQuery を挿入し、リゾルバーの resolve 関数の一部として setCacheValidity (true) を呼び出す必要があります。
- B. 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。
- C. 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\\T\\identityClass) ディレクティブを指定します。

Answer: C (メッセージを残す)

説明

このオプションを使用すると、適切な無効化および区別メカニズムを備えた Varnish または Fastly を使用して、GraphQL クエリ結果をキャッシュできるようになります。

最新問題: 9

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. 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。
- B. 各リダイレクトを .magento/routes.yaml ファイルに追加します。
- C. 管理 UI を介して、各リダイレクトを URL リライトとして追加します。

Answer: A (メッセージを残す)

説明

リダイレクトを構成するための最良のオプションは、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です。これは、大量のリクエストを迅速かつ効率的に処理できるコンテンツ配信ネットワーク (CDN) であり、リダイレクトが迅速かつ確実に処理されるようにします。さらに、VCL スニペットはセットアップが簡単で、他のリダイレクトに再利用できるため、多数のリダイレクトを管理するための効率的でコスト効率の高いソリューションになります。

最新問題: 10

Adobe Commerce Architect は、マーチャントの要件を満たす独自のカスタム価格設定ロジックを備えた新しい製品タイプを含むデータフローを設計します。

実装をレビューする際に有効な 3 つの開発はどれですか？ (3つお選びください。)

- A. etc/product_types.xml ファイルの内容
- B. 新しい製品タイプに属する属性のハイドレーター
- C. 抽象的な製品タイプ モデルから拡張されたカスタム タイプ モデル
- D. 抽象製品モデル クラスを拡張した、カスタム価格設定ロジックを備えた新しいクラス
- E. 新しい品種を登録するためのデータパッチ
- F. \Magento\Catalog\Model\Product\Type\Price を拡張する新しい価格モデル

Answer: A,C,F (メッセージを残す)

独自のカスタム価格設定ロジックを備えた新しい製品タイプを作成するには、次の開発を考慮する必要があります。

etc/product_types.xml ファイルの内容。このファイルは、新しい製品タイプの名前、ラベル、modelInstance、およびpriceModelを定義します。modelInstance は、抽象的な Product Type モデルから拡張されたカスタム タイプ モデルを指定します。PriceModel は、\Magento\Catalog\Model\Product\Type\Price を拡張する新しい価格モデルを指定します。

抽象的な製品タイプ モデルから拡張されたカスタム タイプ モデル。このモデルは、カート用の製品の準備方法、購入リクエストの処理方法、製品オプションの確認方法など、新しい製品タイプのロジックと動作を実装します。

新しい価格モデルは \Magento\Catalog\Model\Product\Type\Price を拡張しました。このモデルは、最終価格、階層価格、最低価格などの計算方法など、新しい製品タイプのカスタム価格設定ロジックを実装します。

参照：

1: <https://meetanshi.com/blog/create-custom-product-type-in-magento-2/>

最新問題: 11

Adobe Commerce ストアのオーナーは、カスタム顧客属性 fny.attribute」 (int 型)を設定します。

アーキテクトは、fny.attribute」が 3 より大きい顧客に対して、ホームページ上の顧客固有のコンテンツを表示する必要があります。Web サイトはフル ページ キャッシュを実行しています。

ベスト プラクティスを使用して、アーキテクトはこれらの要件を実装するためにどの 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. 顧客データ JS ライブラリを使用して fny.attribute」値を取得します
- B. 新しいコンテキスト値 fny.attribute」を Magento\Framework\App\Http\Context に追加します。
- C. カスタム ブロックとコンテンツを含む phtml テンプレートを cmsjindexindex.xml レイアウトに追加します。
- D. 顧客セグメントを作成し、条件に fny.attribute」を使用します。
- E. コンテンツを含むダイナミック ブロックをホームページに追加します。

Answer: B,E (メッセージを残す)

フル ページ キャッシュを有効にしてホームページに顧客固有のコンテンツを表示するには、アーキテクトは新しいコンテキスト値 fny.attribute」を Magento\Framework\App\Http\Context に追加する必要があります。これにより、fny.attribute」の値に基づいてキャッシュを変更できるようになります。次に、アーキテクトはコンテンツを含むダイナミック ブロックをホームページに追加する必要があります。ダイナミック ブロックは、顧客セグメントやその他の条件に基づいてパーソナライズできるコンテンツ ブロックの一種です。参考: <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/cache/page-caching/public-content.html> <https://docs.magento.com/user-guide/marketing /page-builder-add-content-block.html>

最新問題: 12

販売者は、すぐに使える Adobe Commerce アプリケーションを利用しており、顧客向けに新しいポイントカード機能を追加するよう求めています。コードレビュー中に、Adobe Commerce Architect は、この機能用に作成されたreward_card_number 属性設定により、顧客属性がマイアカウント/マイリワードページテンプレートで使用できなくなることに気づきました。

```
seavSetup = $this->customerSetupFactory->create(['setup' => $this->moduleDataSetup]);
seavSetup->addAttribute(
    \Magento\Customer\Model\Customer::ENTITY,
    'reward_card_number',
    [
        'type' => 'varchar',
        'label' => 'Customer Community ID',
        'input' => 'text',
        'user_defined' => true,
        'unique' => false,
        'system' => false,
        'is_used_in_grid' => 1,
        'is_visible_in_grid' => 1,
        'is_filterable_in_grid' => 1,
        'is_searchable_in_grid' => 1,
    ]
);
```

顧客属性を正しく設定するには何を追加する必要がありますか？

- A. グループ プロパティを値 1 で追加する必要があります。
- B. システム プロパティを true の値で追加する必要があります。
- C. スコープ プロパティをグローバルの値で追加する必要があります。

Answer: C ([メッセージを残す](#))

最新問題: 13

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. \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。
- B. 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。
- C. "\Magento\Framework\Config\Data" を実装する Data クラスを作成します。
- D. di.xml に \Magento\Framework\Config\Data」の reader」依存関係を挿入します。
- E. \Magento\Framework\Config\Reader\Filesystem」を実装した Reader クラスを作成します。

Answer: C,E ([メッセージを残す](#))

説明

Web 検索によると、Magento は Data、Reader、Converter などの構成ファイルと対話するためにさまざまなクラスとインターフェイスを使用しているようです。ドキュメント 1 によると、Data はスコープを使用して構成データへのアクセスを提供するクラスです。

Reader は、XML ファイルから設定データを読み取るインターフェイスです。Converter は、XML データを配列表現に変換するインターフェイスです。

これらの定義に基づいて、要件を満たすためにアーキテクトが実行すべき 2 つのステップが考えられます。

* C. "\Magento\Framework\Config\Data" を実装する Data クラスを作成する

* E. "\Magento\Framework\Config\Reader\Filesystem" を実装する Reader クラスを作成する これらの手順により、カスタム 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、PHP でその値を使用できるようになります。クラス。

最新問題: 14

Adobe Commerce Web サイトの開発が完了しました。Web サイトを実稼働環境に展開する準備ができました。

アーキテクトは、ロード バランサーの背後でリクエストを処理する複数のバックエンド Web サーバーで構成される分散アーキテクチャで実行されるシステムを設計しました。

システムを展開して Web サイトに初めてアクセスした後、ユーザーはログイン後にカスタマー ダッシュボードにアクセスできません。ユーザーが正常にログインしても、Web サイトはユーザーをサインイン ページにリダイレクトし続けます。アーキテクトは、セッションが正常であると判断します。正しく保存されていません。

napp/etc/env.php\ では、セッションは次のように構成されています。

```
'session' => [
    'save' => 'redis',
    'redis' => [
        'host' => '127.0.0.1'
    ]
]
```

この問題を修正するにはアーキテクトは何をすべきでしょうか？

- A. リモートストレージ モジュールを利用してサーバー間のセッションを同期します。
- B. セッション ホスト値を共有 Redis インスタンスに更新します
- C. コマンド config:set system/security/max_session_size_admin を使用してセッション サイズを増やします。

Answer: B (メッセージを残す)

ロード バランサーの背後で複数のバックエンド サーバーを使用する場合、セッション データはすべてのサーバーがアクセスできる共有の場所に保存する必要があります。そうしないと、セッション データに一貫性がなくなり、ユーザーが予期せずログアウトされるなどの問題が発生する可能性があります。Redis は、データへの高速かつ信頼性の高いアクセスを提供するため、分散アーキテクチャにセッション データを保存する場合に推奨されるオプションです。env.php ファイル内のセッション ホスト値は、セッション ストレージに使用される Redis インスタンスを指す必要があります。参考:

<https://devdocs.magento.com/guides/v2.4/config-guide/redis/redis-session.html>

最新問題: 15

Adobe Commerce Architect は、月々の分割払い延長のワークフローをカスタマイズする必要があります。この拡張機能は、独自の従来の拡張機能 (非推奨の支払い方法を使用するモジュール) を持つデフォルト Web サイト PSR と契約しているパートナーからのものです。

分割払いパートナーは支払いの初期化のみを管理し、PSP によって実行されるキャプチャを渡します。金額が正常に取得されると、PSP は IPN を通じて Web サイトに通知します。

IPN の目的は、「請求書」を作成し、後で PSP 自体を介した返金リクエストに使用する 「キャプチャ情報」を保存することだけです。

アーキテクトは、要求された動作を副作用なしで取得するための最も単純なソリューションを必要としています。

アーキテクトはどのソリューションを実装する必要がありますか？

A. \$invoice->() の前にプラグインを追加し、その入力を変更して \$payment-> Capture() の呼び出しを防止します。

B. config.xml の支払い方法の can_capture 属性を次のように変更します。

```
<can_capture>0</can_capture>
```

C. di.zm1 の支払い方法 CommandPool に対してタイプ Magento\payment\Gateway\Command\NullCommand のキャプチャ コマンドを宣言します。

Answer: (解答を表示する)

説明

要求された動作を副作用なくキャプチャするために Adobe Commerce Architect が実装する最適なソリューションは、di.xml で支払い方法 CommandPool に対してタイプ Magento\payment\Gateway\Command\NullCommand を使用してキャプチャ コマンドを宣言することです。これにより、パートナーは支払いを初期化し、キャプチャを PSP に引き渡すことができると同時に、Web サイトが \$payment->capture() メソッドを呼び出すことを防ぎます。また、PSP が IPN を通じて Web サイトに通知できるようになり、請求書が作成され、後で PSP 自体を介した払い戻しリクエストに使用される「キャプチャ情報」が保存されます。

最新問題: 16

前回の実稼働展開以降、顧客はチェックアウトを完了できません。エラー ログには次のメッセージが複数回表示されます。

```
main.CRITICAL: Report ID: webapi-61b9fe83f0c3e; Message: Infinite loop detected, review the trace for the looping path
```

アーキテクトは、特定の郵便番号への配信を制限する必要があるデプロイされた機能を見つけます。

アーキテクトは、/webapi_rest\di.xml および etc\frontend\di.xml にデプロイされた次のコードを確認します。

```
<type name="Magento\Shipping\Model\Rate\Result">
    <plugin name="RestrictDeliveryMethods" type="Vendor\RestrictDeliveryMethods\Plugin\Shipping\LimitRates"/>
</type>

LimitRates.php:

public function __construct(
    \Magento\Checkout\Model\Session $session,
    ResultProvider $resultProvider
    ) {
    $this->session = $session;
    $this->resultProvider = $resultProvider;
}

public function afterGetAllRates(\Magento\Shipping\Model\Rate\Result $subject, array $result): array
{
    return $this->resultProvider->getLimitedRates($this->session->getQuote(), $result);
}
```

問題を解決するために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. 挿入された依存関係を置き換えます。

\Magento\Checkout\Model\Session\With\Magento\Framework\Session\SessionManagerInterface

B. \Magento\Quote\API\CartRepositoryInterface のインスタンスを挿入し、\$this->cartRepository->get(\$this->session->getQuoteId()) 経由でカート インスタンスを受け取ります。

C. after プラグインを around プラグインに変更します。この問題は、元のメソッドのコードの後に結果プロバイダー コードを呼び出すことによって発生します。

Answer: C (メッセージを残す)

有効な **AD0-E718** 問題集は GoShiken.com が提供された合格しやすい AD0-E718 試験問題集！ GoShiken.com が最新の **AD0-E718** 試験問題集を提供しています。GoShiken.com AD0-E718 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-E718 問題集をゲットする人はこちら: <https://www.goshiken.com/Adobe/AD0-E718-mondaishu.html> (5030%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

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済みのロジックは、作成済みで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. 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。 `\Magento\testFramework\store\Executionstorecontext` クラスをフィクスチャで 1 回使用し、テストでもう 1 回使用します。
- B. 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。
- C. 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 `@magentoStoreContext 3` アノテーションを使用し、もう一方のメソッドでは `@magentoStoreContext 4` を使用します。

Answer: ([解答を表示する](#))

説明

アーキテクトが要件を満たすための最良のアプローチは、オプション B です。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。クラス レベルで `@magentoExecuteInStoreContext 3` および `@magentoExecuteInStoreContext 4` アノテーションを使用します。これにより、フィクスチャがストア ビュー 3 のコンテキストで実行され、テストされるロジックがストア ビュー 4 のコンテキストで実行されることが保証されます。このアプローチにより、ストア ビューをより詳細に制御できるようになり、テストの複雑さが軽減されます。

最新問題: 18

アーキテクトは、複数のカスタム支払い方法をリファクタリングする新しく開発されたプル リクエストをレビューしているときに、機密データの資格情報を復号化するために `\Magento\Framework\Encryption\EncryptorInterface` に依存する複数のクラスに気づきました。一般的に繰り返されるコードは次のとおりです。

```

namespace Vendor\PaymentModule\Gateway\Config;

class Config extends \Magento\Payment\Gateway\Config\Config
{
    ...
    public function __construct(
        ...
        ScopeConfigInterface $scopeConfig,
        EncryptorInterface $encryptor,
        ...
    ) {
        parent::__construct($scopeConfig, $methodCode, $pathPattern);
        $this->scopeConfig = $scopeConfig;
        $this->encryptor = $encryptor;
    }

    public function getUserSecret(): string
    {
        return $this->encryptor->decrypt(
            $this->scopeConfig->getValue('payment/method_code/user_secret')
        );
    }
}

```

各モジュールで、user_secret 構成は次のように宣言されます。

```

<field id="user_secret" translate="label" type="obscure" sortOrder="20" showInDefault="1" >
    <label>Secret Key</label>
    <backend_model>Magento\Config\Model\Config\Backend\Encrypted</backend_model>
</field>

```

アーキテクトは、メソッド間の冗長な依存関係やコードの重複を避けるために、最適なソリューションを推奨する必要があります。アーキテクトはどのソリューションを推奨する必要がありますか？

A. すべての Vendor\PaymentModule\Gateway\Config\Config クラスを Magento\Payxer.t\Gateway\Config\Config の virtualTyp に置き換え、<user_secret backend_Model="Magento\Config\Model\Config\Backend\Encrypted" を設定します。/> ccnfig.xml の下

B. \$sccpeConfig の getvalue メソッドの後にプラグインを追加し、依存関係から \$encryptor を削除し、構成名が 'user.secret?' の場合はそれをプラグインで使用して値を復号します。

C. Vendor.Payment の下に共通の構成サービス クラス Vendor\Payment\Gateway\config\Config を作成し、それをすべての Vender\EaymentModule\Gateway\Config\Config クラスの親クラスとして使用し、\$sccpeConfig と \$ を削除します。暗号化の依存関係

Answer: ([解答を表示する](#))

メソッド間での冗長な依存関係やコードの重複を避けるには、次の解決策を使用する必要があります。

標準のコントローラ ルートを作成し、内部 URL (news/article/view/id/1 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。このソリューションは、Magento の組み込み URL 書き換え機能を利用してニュース リクエストを処理します。内部 URL を処理し、その ID によってニュース記事をロードする標準コントローラ ルートを作成できます。次に、日付と URL キーに基づいて各ニュース記事の URL リライトを生成し、URL リライト テーブルに保存できます。こうすることで、カスタム ルーターやプラグインの使用を回避し、コードの複雑さを軽減できます。

参照：

<https://devdocs.magento.com/guides/v2.4/extension-dev-guide/routing.html>

<https://devdocs.magento.com/guides/v2.4/extension-dev-guide/url-rewrite.html>

最新問題: 19

Adobe Commerce Architect が API 機能テスト コードをレビューしています。一部のテストでは、顧客のアドレスが存在しないことを示すエラーが送信されます。

テストコードは次のことを示します。

```
/**
 * @magentoDataFixture Magento/Customer/_files/customer_one_address.php
 * ...
 */
public function testMyUseCasTestForCartAddress(): void
{
    // ...
}

Change the annotation to use @magentoApiDataFixture Magento/Customer/_files/customer_one_address.php
instead of @magentoDataFixture Magento/Customer/_files/customer_one_address.php
A.
Update the annotation to specify address_id @magentoDataFixture Magento/Customer/_files/customer_one_address.php with:
B. ("address_id":"$address.id$")
Set the annotation to use @magentoPersistDataFixture Magento/Customer/_files/customer_one_address.php
instead of @magentoDataFixture Magento/Customer/_files/customer_one_address.php
C.
```

Answer: C ([メッセージを残す](#))

最新問題: 20

Adobe Commerce Architect は、キューのコンシューマーが Adobe Commerce Cloud サーバー上で TCP 接続を頻繁に閉じて、メッセージの処理に遅れが生じていることに気付きました。

アーキテクトは、CRON ジョブがコンシューマーを実行しているときに、キュー内の使用可能なメッセージを処理した後にコンシューマーが終了しないことを確認する必要があります。

アーキテクトはこの要件をどのように満たすべきでしょうか？

- A. multiple_process 制限を増やして、各コンシューマーに対してより多くのプロセスを生成します。
- B. デプロイメント段階で CONSUMER_WAIT_FOR_MAX_MESSAGES 変数を true に設定します。
- C. CRON_CONSUMER_RUNNER変数のmax_messagesを10,000から1,000に変更します。

Answer: B ([メッセージを残す](#))

説明

この要件を満たす最善の方法は、展開段階で CONSUMERWAITFORMAXMESSAGES 変数を true に設定することです。この変数は、キューにメッセージがなくなってもコンシューマが終了せず、代わりに新しいメッセージが利用可能になるまで待機することを保証し、接続が途中で閉じられるのを防ぎます。さらに、multiple_processes の制限を増やすと、各コンシューマーに対してより多くのプロセスを生成できるようになり、メッセージをより速く処理できるようになります。

最新問題: 21

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_attribure 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 を介して Adobe Commerce と通信しており、変更されるクエリはすでにキャッシュされています。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)

```
Create a new class inheriting from Magento\Framework\GraphQL\Query\Resolver\IdentityInterface and returning the value of my_attribute from the getIdentities function. Then specify a @cache(cacheIdentity: Path\To\IdentityClass) directive for each GraphQL query to include the newly created IdentityClass to each query that adds the cache tags for each customer.
```

B)

```
Create a new class inheriting from Magento\GraphQLCache\Model\CacheId\CacheIdFactorProviderInterface and returning the value of my_attribute from the getFactorValue function and my_attribute from the getFactorName function. Then add this class through DI to the idFactorProviders array of Magento\GraphQLCache\Model\CacheId\CacheIdCalculator.
```

C)

```
Create a new class inheriting from Magento\Customer\CustomerData\SectionSourceInterface and returning the value of my_attribute from the getSectionData function. Then add this class through DI to the sectionSourceMap array of Magento\Customer\CustomerData\SectionPoolInterface.
```

A. オプション A

B. オプション B

C. オプション C

Answer: ([解答を表示する](#))

オプション B は、Varnish の `vcl_hash` 関数が新しく作成された属性も考慮するための正しい方法です。アーキテクトは、`Magento_Fastly` に依存する新しいモジュールを作成し、`\Magento\Fastly\Model\Config::getVclSnippets()` メソッドのプラグインを宣言する必要があります。プラグインは、タイプ `fev` およびコンテンツ `$set req.http.my_attribute = req.http.X-Magento-Vary;` を持つ新しいスニペットを結果配列に追加する必要があります。これにより、Varnish がキャッシュに使用するハッシュキーに顧客属性の値が追加されます。オプション A は、`Magento_Fastly` モジュールのコア ファイルを変更するため不正解です。これは推奨されず、アップグレード中に問題が発生する可能性があります。オプション C は、スニペットに間違ったタイプとコンテンツを使用するため、間違っています。これは、`vcl_hash` 関数には影響しません。参照:

<https://devdocs.magento.com/cloud/cdn/fastly-vcl-bypass-to-origin.html> Adobe Commerce 開発者ガイド 2 によると、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 `Magento\Customer` のプラグインを使用する必要があります。`\Model\Context::getVaryString` メソッドを実行し、属性値を可変文字列に追加します。これにより、Varnish は顧客の属性値に基づいてさまざまなバージョンの GraphQL クエリをキャッシュするようになります。したがって、選択肢 B が正しいです。選択肢 A は、`vcl_hash` 関数に影響を与えないため、不正解です。オプション C は、`getVaryString` メソッドのプラグインを使用しないため、不正解です。アーキテクトは、`Magento\Customer\Model\Context::getVaryString` メソッドのプラグインを使用し、属性値を可変文字列に追加する必要があります。これにより、Varnish は顧客の属性値に基づいてさまざまなバージョンの GraphQL クエリをキャッシュするようになります。したがって、選択肢 B が正しいです。選択肢 A は、`vcl_hash` 関数に影響を与えないため、不正解です。オプション C は、`getVaryString` メソッドのプラグインを使用しないため、不正解です。

最新問題: 22

アーキテクトは、Adobe Commerce オンプレミスインスタンスで Redis の preload.keys を設定しています。

アーキテクトは、各フロントエンド リクエストでキャッシュ キー


eav_entity_types、GLOBAL_PLUGIN_LIST、DB_IS_UP_TO_DATE、SYSTEM_DEFAULT がロードされていることを検出します。

* フロントエンド =>page_cache の id_prefix は 061_ に設定されます。

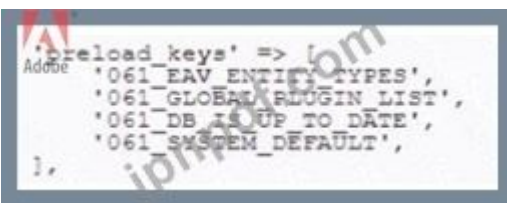
※frontend =>default:のid_prefixは設定されていません。

* アーキテクトは Redis L2 キャッシュを有効にして構成しました。

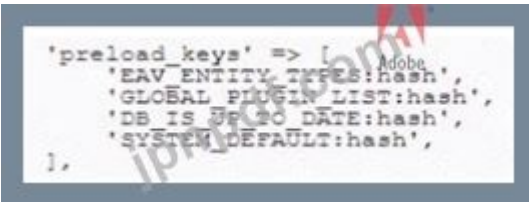
preload.keys はどのように設定する必要がありますか?

A. 

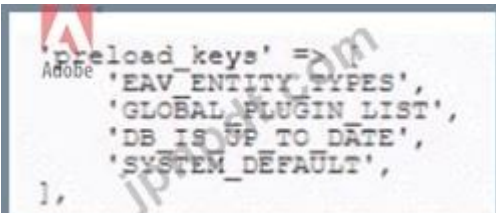
```
'preload_keys' => [  
  '061_EAV_ENTITY_TYPES:hash',  
  '061_GLOBAL_PLUGIN_LIST:hash',  
  '061_DB_IS_UP_TO_DATE:hash',  
  '061_SYSTEM_DEFAULT:hash',  
],
```

B. 

```
'preload_keys' => [  
  'EAV_ENTITY_TYPES',  
  'GLOBAL_PLUGIN_LIST',  
  'DB_IS_UP_TO_DATE',  
  'SYSTEM_DEFAULT',  
],
```

C. 

```
'preload_keys' => [  
  'EAV_ENTITY_TYPES:hash',  
  'GLOBAL_PLUGIN_LIST:hash',  
  'DB_IS_UP_TO_DATE:hash',  
  'SYSTEM_DEFAULT:hash',  
],
```

D. 

```
'preload_keys' => [  
  'EAV_ENTITY_TYPES',  
  'GLOBAL_PLUGIN_LIST',  
  'DB_IS_UP_TO_DATE',  
  'SYSTEM_DEFAULT',  
],
```

Answer: B ([メッセージを残す](#))

最新問題: 23

Adobe Commerce ストアのオーナーは、カスタム顧客属性 fny.attribute」 (int 型) を設定します。

アーキテクトは、fny.attribute」が 3 より大きい顧客に対して、ホームページ上の顧客固有のコンテンツを表示する必要があります。Web サイトはフル ページ キャッシュを実行しています。

ベスト プラクティスを使用して、アーキテクトはこれらの要件を実装するためにどの 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. 顧客データ JS ライブラリを使用して fny.attribute」値を取得します

B. 新しいコンテキスト値 fny.attribute」を Magento\Framework\App\Http\Context に追加します。

- C. カスタム ブロックとコンテンツを含む phtml テンプレートを cmsjindexindex.xml レイアウトに追加します。
- D. 顧客セグメントを作成し、条件に fmy.attribute」を使用します。
- E. コンテンツを含むダイナミック ブロックをホームページに追加します。

Answer: A,C (メッセージを残す)

説明

<https://docs.magento.com/user-guide/v2.3/stores/attributes-customer.html> ベスト プラクティスを使用してキャッシュされたページにカスタム顧客属性を表示するには、次のようないくつかの手順が必要です。

- * 表示するコンテンツを含むカスタム ブロックと phtml テンプレートを作成する
- * カスタム ブロックを表示するページのレイアウト ファイルに追加する
- * カスタム属性の新しいセクションを宣言するためのsection.xmlファイルの作成
- * セクションデータにカスタム属性値を追加するための Magento\Customer\CustomerData\SectionPoolInterface のプラグインを作成する
- * 顧客データ JS ライブラリを使用して、phtml テンプレートのカスタム属性値を取得して表示します

最新問題: 24

アーキテクトは、すべての顧客セッション データがランダムに失われる販売者の Adobe Commerce 運用環境を調査しています。顧客のセッション データは、すべてのキャッシュと同様に、Redis を使用して永続化されるように構成されています (Varnish を介して処理されるフル ページ キャッシュを除く)。

最初のレビューの後、アーキテクトは、Adobe Commerce 管理パネルを使用するか、コマンドラインで bin/iuagento cache: flash を実行して、Magento キャッシュストレージをフラッシュすることで、顧客セッションデータの損失を再現できます。Adobe Commerce 管理パネルですべてのキャッシュを更新するか、コマンドラインで bin/magento cache: clean を実行しても、セッション データは失われません。

次のステップは何でしょうか？

- A. キャッシュ ストレージをフラッシュせず、今後はキャッシュのみを更新するように販売者を教育します。
- B. Adobe Commerce 管理パネルで、「セッションデータを個別に保存」の「ストア/設定」オプションを「はい」に設定します。
- C. app/etc/evn.php を確認し、キャッシュとセッション データの Redis 構成で異なるデータベース番号が使用されていることを確認します。

Answer: (解答を表示する)

説明

これは、Magento キャッシュ ストレージがフラッシュされると、Redis に保存されているすべてのキャッシュとセッション データがクリアされるためです。キャッシュがフラッシュされたときにセッション データが失われるのを防ぐために、販売者はキャッシュとセッション データに対して異なる Redis データベースを構成して、Magento キャッシュ ストレージをフラッシュしてもキャッシュのみがクリアされ、セッション データはクリアされないようにする必要があります。

最新問題: 25

Adobe Commerce Architect は、オンプレミスでの Adobe Commerce の導入プロセス中にダウンタイムがゼロであることを保証する必要があります。アーキテクトはどの 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. bin/magento setup:upgrade --keep-generated を実行してデータベースをアップグレードします
- B. bin/magento setup:upgrade -dry-run=true を実行してデータベースをアップグレードします
- C. リム bin/magento setup:upgrade --convert-old-scripts=true でデータベースをアップグレードします

D. Developer/zere_down_time/enabled で設定フラグを有効にします。

E. デプロイメント/blue_green/enabled で構成フラグを有効にします。

Answer: B,E (メッセージを残す)

1. bin/magento setup:upgrade -dry-run=true を実行すると、データベースに変更を適用せずにアップグレードスクリプトを確認できます。これは、実際のアップグレード前に潜在的な問題を特定するのに役立ちます。E.deployment/blue_green/enabled で設定フラグを有効にすると、Blue-Green デプロイメント戦略を使用できるようになります。これにより、実稼働環境のコピーが作成され、新しいバージョンのテスト後に2つの環境間でトラフィックが切り替えられます。これにより、展開プロセス中にダウンタイムをゼロにすることができます。参考: <https://devdocs.magento.com/guides/v2.4/comp-mgr/cli/cli-upgrade.html#upgrade-cli-dryrun> <https://devdocs.magento.com/cloud/live/stage-prod-merge-prereq.html#blue-green-deployment>

最新問題: 26

ある会社が Adobe Commerce ストアを持っています。fny.attribute」という名前の属性 (タイプ fext) は、複数のシステム間で共有される各製品のグローバル ID を保存するために作成されます。

運用開始から数か月後、fny.attribute」の値はすべて整数になっています。これにより、他のシステムがこのデータを受信するときに問題が発生します。

Adobe Commerce アーキテクトは、fny.attribute」のタイプを text から int に変更するソリューションを推奨する必要があります。これを達成するには、アーキテクトはどの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. テーブル catalog_product_entity_text から catalog_product_entity_int に attribute.id のデータを移行します。

B. [管理] > [ストア] > [属性] > [製品] に移動し、fny.attribute」を編集し、タイプを fext から fint に更新します。

C. \Magento\Eav\Model\Entity\Attribute\Backend\AbstractBackend::afterLoad() のプラグインを作成し、catalog_product_entity_int からデータをロードします

D. データパッチを作成し、fny.attribute」タイプを fext から fint に更新します。

E. コマンド bin/magento indexer:resetcatalog_product_attribute を実行します。

Answer: A,D (メッセージを残す)

オプション A は正解です。属性 ID に基づいてデータをあるテーブルから別のテーブルに移行します。これは属性タイプを変更するときに必要です。オプション D は正解です。これにより、fny.attribute」タイプを fext から fint に更新するデータパッチが作成されます。これは、プログラムによって属性タイプを変更する推奨方法です。

fny.attribute」のタイプを text から int に変更するには、アーキテクトは2つの手順を実行する必要があります。A) テーブル catalog_product_entity_text から catalog_product_entity_int に、attribute_id のデータを移行します。このステップでは、属性の既存の値を text テーブルから新しい型に対応する int テーブルに移動します。D) データパッチを作成し、fny.attribute」タイプを fext から fint に更新します。この手順では、データベース内の属性メタデータが更新され、新しいタイプが反映されます。オプション B は、管理パネルでは既存の属性のタイプを変更できないため、不正解です。Magento\Eav\Model\Entity\Attribute\Backend\AbstractBackend::afterLoad() のプラグインを作成しても属性のタイプは変更されないため、オプション C は不正解です。ただし、別のテーブルからデータをロードするだけです。オプション E は不正解です。コマンド bin/magento indexer:resetcatalog_product_attribute を実行しても属性のタイプは変更されず、インデクサーのステータスがリセットされるだけです。参考: <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/attributes.html>

最新問題: 27

Adobe Commerce Architect は、事前定義された製品セットを割引価格で購入できる新製品をサイト上で紹介する販売キャンペーンに取り組んでいます。セット内の各商品には個別の在庫および税クラスが必要です。

要件の 1 つは、サードパーティ システムを使用して REST API でレポートを作成し、次のデータを取得することです。

* SKU

* 数量

* 元値

※ 販売価格

* 課税額

これらの要件を満たすためにアーキテクトはどのソリューションを使用する必要がありますか？

A. * 単純な製品を収集するための固定バンドル製品を作成します。

* 選択したすべてのオプションの価格を管理します。

* 拡張属性 `original_simple_price` を追加

`\Magento\Sales\Api\Data\OrderItemExtensionInterface` に、単純な製品の価格を値に入力します。

B. * 単純な製品を収集するための動的バンドル製品を作成します。

* コンテンツ ステージングを利用して、キャンペーンに間に合うようにバンドル製品の特別価格を管理します。

* Adobe Commerce Order API を介して必要なデータを公開します。

C. * グループ化された製品を作成し、プラグイン後に作成します。

`\Magento\GroupedProduct\Model\Product\Type\Grouped:preparedForCarrAdvanced` (一括商品注文用)。

* コンテンツ ステージングを利用して、シンプルな製品のキャンペーンに合わせて特別価格を管理します。

* Adobe Commerce Order API を介して必要なデータを公開します。

Answer: B (メッセージを残す)

説明

バンドル製品は、それぞれが単純な製品または仮想製品に基づいた複数のオプションで構成されるカスタマイズ可能な製品です。グループ化された製品は、グループとして提示される単純な製品のコレクションです。

いくつかのチュートリアルによると、Adobe Commerce でバンドル製品を作成するには、次のようないくつかの手順が必要です。

* バンドル製品テンプレートと属性セットの選択

* 名前、SKU、価格、重量などの必要な設定を完了します

* ステータス、可視性、カテゴリなどの基本設定を構成する

* バンドル オプションと関連製品の追加

※ 画像やメタデータなどのオプション商品情報の追加

※ 商品の掲載について

コンテンツ ステージングは、ストアのコンテンツ更新を管理者から直接作成、プレビュー、スケジュールできるようにする機能です。コンテンツ ステージングを使用して、製品、カテゴリ、ページ、ブロック、ウィジェット、価格ルールなどの変更を含むキャンペーンを作成できます。

これらの手順と機能に基づいて、これらの要件を満たすためにアーキテクトが使用すべき考えられるソリューションの 1 つは次のとおりです。

* B. シンプルな製品を収集するための動的バンドル製品を作成します。コンテンツ ステージングを利用して、キャンペーンに合わせてバンドル製品の特別価格を管理します。Adobe Commerce Order API を介して必要なデータを公開します。このソリューションにより、事前に定義された一連の製品を割引価格で購入できる新しい製品を作成できます。セット内の各製品には、個別の在庫および税クラスが設定されます。バンドル製品の特別価格は、コンテンツ ステージングを使用して管理できます。必要なデータは、Adobe Commerce Order API を介して公開される可能性があります。

最新問題: 28

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。ミュレーションでは、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、ミュレーションはこのバリエーションをカートに追加し、null 値を許容しないカート タイプ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、ミュレーションは 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。

```
type Mutation {
  addConfigurableToCart(product_id: Int!): AddToCartOutput!
  @resolver(class: "Vendor\\MyModule\\Model\\Resolver\\AddConfigurableToCart")
}
```

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？

```
interface AddToCartOutput
@typeResolver(class: "Vendor\\MyModule\\Model\\Resolver\\AddToCartOutputTypeResolver") {
}

type ConfigurableProduct implements AddToCartOutput {
}

type Cart implements AddToCartOutput {
}
```

A.

```
union AddToCartOutput
@typeResolver(class: "Vendor\\MyModule\\Model\\Resolver\\AddToCartOutputTypeResolver")
= ConfigurableProduct | Cart
```

B.

```
type AddToCartOutput {
  product: ConfigurableProduct
  cart: Cart
}
```

C.

Answer: B (メッセージを残す)

オプション B は、この突然変異の出力を宣言する正しい方法です。出力タイプは、指定された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る共用体タイプである必要があります。共用体タイプは、キーワード Union とパイプ記号 (|) を使用して宣言し、可能な型を区切る必要があります。オプション A は、共用体タイプの宣言に間違った構文を使用しているため、不正解です。オプション C は、共用体タイプではなくインターフェース型を使用しているため不正解です。つまり、出力型はインターフェースのすべてのフィールドを実装する必要がありますが、Cart 型と ConfigurableProduct 型には当てはまりません。参考: <https://devdocs.magento.com/guides/v2.4/graphql/develop/create-graphqls-file.html>

最新問題: 29

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ます。すべてのブランド関連ページは、製品やカテゴリと同じ方法でタグを使用して 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。

```
type Query {
  brands (
    search: String @doc(description: "Search against brand names (partial matches)")
    pageSize: Int = 20 @doc(description: "Number of brand results to return")
    currentPage: Int = 1 @doc(description: "Page number to return")
  ) : BrandsResult
  @resolver(class: "ClientName\\ProductBrandsGraphQL\\Model\\Resolver\\Brands")
  brand (
    urlKey: String! @doc(description: "Match against brand url key")
  ) : Brand
  @resolver(class: "ClientName\\ProductBrandsGraphQL\\Model\\Resolver\\BrandByUrlKey")
}
```

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. 各 GraphQL クエリのリゾルバー クラスは、MagentoGraphQLCacheModelCacheableQuery を挿入し、リゾルバーの resolve 関数の一部として setCacheValidity (true) を呼び出す必要があります。
- B. 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。
- C. 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\\T\\identityClass) ディレクティブを指定します。

Answer: C (メッセージを残す)

このオプションを使用すると、適切な無効化および区別メカニズムを備えた Varnish または Fastly を使用して、GraphQL クエリ結果をキャッシュできるようになります。

GraphQL クエリ内の情報が古いという問題を解決するには、アーキテクトは各 GraphQL クエリに対して @cache (cacheIdentity: path\\T\\identityClass) ディレクティブを指定する必要があります。これは、関連するブランドおよび関連するキャッシュ タグを追加するクラスに対応します。製品。これにより、ブランドまたは製品データが変更されたときに、Varnish はこれらのクエリのキャッシュを無効にすることができます。オプション A は不正解です。キャッシュ タグなしでクエリをキャッシュ可能にするため、問題は解決されません。オプション B は、これらのクエリのキャッシュを完全に無効にし、パフォーマンスを低下させるため、不正解です。参考: <https://devdocs.magento.com/guides/v2.4/graphql/caching.html>

最新問題: 30

サードパーティ企業は、レポート用の注文データを取得するために Adobe Commerce システムを統合するアプリケーションを作成する必要があります。統合には get /vi/orders エンドポイントへのアクセスが必要です。このエンドポイントは 24 時間ごとに自動的に呼び出されます。販売者は、リソースへのアクセスを制限または拡張したり、管理パネルを使用してアクセスを取り消したりする機能を必要としています。

この統合では、Adobe Commerce で利用可能などのタイプの認証を使用し、サードパーティ システムに実装する必要がありますか？

- A. トークンベースの認証を使用して管理者トークンを取得します。サードパーティ システムは、管理者のユーザー名とパスワードを使用して REST エンドポイントを利用して、承認用のベアラー トークンとして使用される管理者トークンを取得します。

B. OAuth ベースの認証を使用して、システム リソースへのアクセスを提供します。統合は、アクティベーション中の OAuth ハンドシェイクのパネルで販売者によって登録されます。サードパーティ システムは、OAuth プロトコルに従って認証する必要があります。

C. トークンベースの認証を使用して統合トークンを取得します。統合は、デフォルトの統合トークン設定を使用して管理パネルで作成およびアクティブ化され、トークンへのアクセスを取得します。トークンは、承認するベアラー トークンとして使用されます。

Answer: ([解答を表示する](#))

説明

ドキュメント1によると、トークンベースの認証は、統合のアクティブ化時に生成されるアクセス トークンを使用してリソースにアクセスする簡単な方法です。OAuth ベースの認証は、統合が登録および承認されるときに生成されるコンシューマー キー、コンシューマー シークレット、アクセス トークン、およびアクセス トークン シークレットを使用してリソースにアクセスするためのより安全な方法です。

これらの定義に基づくと、この統合のためにサードパーティ システムで使用および実装する必要がある認証のタイプは次のとおりです。

* B. OAuth ベースの認証を使用して、システム リソースへのアクセスを提供します。統合が登録されます

* アクティベーション中にパネル内の販売者による OAuth ハンドシェイク。サードパーティ システムは、OAuth プロトコルに従って認証する必要があります。

最新問題: 31

Adobe Commerce Architect は、キューのコンシューマーが Adobe Commerce Cloud サーバー上で TCP 接続を頻繁に閉じて、メッセージの処理に遅れが生じていることに気付きました。

アーキテクトは、CRON ジョブがコンシューマーを実行しているときに、キュー内の使用可能なメッセージを処理した後にコンシューマーが終了しないことを確認する必要があります。

アーキテクトはこの要件をどのように満たすべきでしょうか？

A. multiple_process 制限を増やして、各コンシューマーに対してより多くのプロセスを生成します。

B. デプロイメント段階で CONSUMER_WAIT_FOR_MAX_MESSAGES 変数を true に設定します。

C. CRON_CONSUMER_RUNNER変数のmax_messagesを10,000から1,000に変更します。

Answer: ([解答を表示する](#))

この要件を満たす最善の方法は、展開段階で CONSUMERWAITFORMAXMESSAGES 変数を true に設定することです。この変数は、キューにメッセージがなくなってもコンシューマーが終了せず、代わりに新しいメッセージが利用可能になるまで待機することを保証し、接続が途中で閉じられるのを防ぎます。さらに、multiple_processes の制限を増やすと、各コンシューマーに対してより多くのプロセスを生成できるようになり、メッセージをより速く処理できるようになります。

デプロイメント段階で CONSUMER_WAIT_FOR_MAX_MESSAGES 変数を true に設定すると、CRON ジョブがこれらのコンシューマーを実行しているときに、キュー内の使用可能なメッセージを処理した後にコンシューマーが終了しないことが保証されます。

これにより、コンシューマーが TCP 接続を頻繁に閉じることがなくなり、パフォーマンスが向上します。Adobe

Commerce ヘルプセンターの「メッセージキューコンシューマーの開始」を参照してください¹。参照: [https://](https://experienceleague.adobe.com/docs/commerce-operations/configuration-guide/message-queues/consumers.html?lang=en2)

experienceleague.adobe.com/docs/commerce-operations/configuration-guide/message-queues/consumers.html?lang=en2

[https:// experienceleague.adobe.com/docs/commerce-operations/configuration -guide/cli/start-message-queues.html1](https://experienceleague.adobe.com/docs/commerce-operations/configuration-guide/cli/start-message-queues.html1)

有効な **AD0-E718** 問題集は GoShiken.com が提供された合格しやすい AD0-E718 試験問題集！ GoShiken.com が最新の **AD0-E718** 試験問題集を提供しています。GoShiken.com AD0-E718 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-E718 問題集をゲットする人はこちら: <https://www.goshiken.com/Adobe/AD0-E718-mondaishu.html> (5030%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: 32

マーケティング キャンペーンにより、Web サイトで非常に多くの同時注文が発生しており、チェックアウトのパフォーマンスに影響を与えています。Web サイトは実稼働展開モードになっています。

チェックアウトのパフォーマンスへの影響を軽減するためにアーキテクトが最適化できる 2 つの Web サイト設定はどれですか? (2つお選びください。)

- A. 非同期インデックス作成管理パネルの設定 (ストア > 設定 > Ccr.figuraticr. > 詳細 > 開発者 > グリッド設定 > 非同期インデックス作成) は、CLI コマンド `tm/magento config:set dev/grid/` を実行することで有効にできます。 `async_ini*xmg 1`
- B. マルチスレッド チェックアウト処理の管理パネル設定 (ストア > 設定 > 構成 > 販売 > チェックアウト > 一般設定 > 非同期) は、チェックアウト専用で使用される PHP スレッドの数を表すより高い値に設定できます。
- C. 非同期メール通知管理パネルの設定 (ストア > 設定 > 構成 > 販売 > 販売メール > 一般設定 > 非同期) を有効にできます。
- D. 次のコマンドを使用して、新しいデータベースを作成し、データベースの分割機能を自動的に構成できます: `bin/Magento setup:db-schema:split-sales-sales --host="<checkout db host or ip >" -dbname="<名前>" -username""<チェックアウト DB ユーザー名>" -password="<パスワード>"`
- E. Web サイトの展開モードは、CLI コマンド `bin/magentodeploy:mode:set siege` を実行することで `si-g-` に設定できます。ただし、同時に発注された注文の数が許容レベルに減少し次第、生産に戻されます。

Answer: A,D (メッセージを残す)

非同期インデックス作成により、Magento はストアフロントのパフォーマンスに影響を与えることなく、バックグラウンドでインデクサーを処理できます。データベースを分割すると、Magento はチェックアウト関連のテーブルに別のデータベースを使用できるようになり、メイン データベースの負荷が軽減され、チェックアウトのパフォーマンスが向上します。参考:

<https://devdocs.magento.com/guides/v2.4/extension-dev-guide/indexing.html#asynchronous-indexing>

<https://devdocs.magento.com/guides/v2.4/config-guide /マルチマスター/マルチマスター.html>

最新問題: 33

Adobe Commerce Architect は、管理者が特定のセグメントに「平均売上高」条件を指定できるように、新しい顧客セグメント条件を作成する必要があります。

アーキテクトはカスタム条件を次のように開発します。

`vendor\Module\Model\Segment\condition\AverageSalesAmount` とそのすべての要件:

```

<?php
declare(strict_types=1);
namespace Vendor\Module\Plugin;
use Magento\CustomerSegment\Model\Segment\Condition\Combine;
class AddNewChildOption
{
    public function afterGetNewChildSelectOptions(
        Combine $subject,
        array $result
    ): array {
        $conditions = [
            [
                'value' => \Vendor\Module\Model\Segment\Condition\AverageSalesAmount::class,
                'label' => __('Average sales amount'),
            ]
        ];
        return array_merge_recursive($result, $conditions);
    }
}

```



テスト中に次のエラーが表示されます。

アーキテクトが問題を解決するには、どの 2 つの手順を完了する必要がありますか? (2つお選びください。)

- A. Set the class to be `Magento\CustomerSegment\Model\Segment\Condition\AverageSalesAmount` for the `$conditions` value attribute
- B. Set the class in the `$conditions` value attribute to be `Segment\Condition\AverageSalesAmount`
- C. Use a virtualType `<virtualType name="Magento\CustomerSegment\Model\Segment\Condition\AverageSalesAmount" type="Vendor\Module\Model\Segment\Condition\AverageSalesAmount"/>`
- D. Use a preference `<preference for="Magento\CustomerSegment\Model\Segment\Condition\AverageSalesAmount" type="Vendor\Module\Model\Segment\Condition\AverageSalesAmount"/>`
- E. Remove the trailing path `Magento\CustomerSegment\Model\Segment\Condition\` from the `$conditions` value attribute

Answer: D,E (メッセージを残す)

最新問題: 34

Adobe Commerce Architect は、オンプレミスの Adobe Commerce プロジェクトの導入と構築ツールをサポートします。このツールは、集中サーバー上でビルドスクリプトを実行し、SSH 接続を使用してプロジェクトサーバーに展開します。クライアントは、ユーザーがインターフェイスロケールを変更するたびにサイトが中断されるため、管理パネルで作業できないと報告しました。

保守性を考慮すると、アーキテクトはどのソリューションを実装する必要がありますか?

- A. プロジェクトの `env.php` ファイルを編集し、`admin_locales_for_build` 値を構成し、必要なロケールをすべて指定します。
- B. ツールのビルドスクリプトを調整し、`setup:static-content:deploy` コマンド中に必要なロケールを指定します。
- C. プロジェクトの `config.php` ファイルを変更し、`admin_locales_for_deploy` 値を構成し、必要なロケールをすべて指定します。

Answer: B (メッセージを残す)

説明

管理パネルが正しく動作し、保守しやすいことを確認するには、アーキテクトはツールのビルドスクリプトを調整して、`setup:static-content:deploy` コマンドの実行時に必要なロケールをすべて指定する必要があります。これにより、サポートされているすべてのロケールに対してプロジェクトが正しく構成され、新しいロケールが追加されるたびにビルドスクリプトを変更する必要がなくなります。

最新問題: 35

Adobe Commerce システムは、以下を含む多層アーキテクチャで実行されるように構成されています。

* Varnish がインストールされたキャッシュ サーバー

* Adobe Commerce がインストールされたバックエンド Web サーバー

※MySQLがインストールされたデータベースサーバー

Adobe Commerce Architect がキャッシュ管理の「Magento キャッシュのフラッシュ」を使用してストア管理者からキャッシュを消去しようとする、フルページキャッシュは消去されません。

フルページキャッシュを適切に機能させるために、アーキテクトはどの 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. [ストア管理ストア] > [構成] > [詳細設定] > [システム] > [フル ページ キャッシュ] > [ワニス構成] > [バックエンド ホスト] で、バックエンド宛先ホストをフロントエンド サーバーのアドレスに設定します。
- B. ストア管理ストア > 構成 > 詳細 > システム > フル ページ キャッシュ > Varnish 構成 > バックエンド ポートで、バックエンド ポートの宛先をフロントエンド サーバーの Varnish ポートに設定します。
- C. ストア管理でキャッシュ タイプを「ワニス キャッシュ」に設定します。
「ストア」> 「構成」> 「詳細」> 「システム」> 「フルページキャッシュ」> 「キャッシュアプリケーション」
- D. Magento キャッシュのフラッシュ」の代わりに「キャッシュストレージのフラッシュ」を使用します。
- E. magento CLI bin/magento setup:config:set --http-cache-hosts<cache_server>:<varnish port> を使用してキャッシュ宛先ホストを設定します。

Answer: C,E (メッセージを残す)

Varnish をページ全体のキャッシュとして使用するには、ストア管理でキャッシュ タイプを「Varnish Caching」に設定する必要があります。これにより、「Flush Magento Cache」ボタンが Varnish にページ リクエストを送信できるようになります。さらに、magento CLI コマンドを使用してキャッシュ宛先ホストを指定し、どの Varnish サーバーをページするかを Magento に指示する必要があります。参考: <https://devdocs.magento.com/guides/v2.4/config-guide/varnish/config-varnish.html>

最新問題: 36

アーキテクトは、オブザーバーを使用して店頭での顧客活動データを記録するカスタム モジュールをレビューしています。クライアントは、Adobe Commerce のステージング プレビュー機能を使用して、今後予定されているキャンペーンに向けて管理者パネルからストアフロント カタログ ページをプレビューしているときにログが記録されたと報告します。

この問題に対処するためにアーキテクトは最初に何を確認する必要があります?

- A. 戻り値を変更する \Magento\Staging\Model\Event\Manager のパブリック メソッド isAllowedObserver() のプラグイン
- B. ロギング オブザーバーは、etc\events.xml から etc\adminhtml\events.xml に属性 disabled="true" でコピーされます。
- C. BandObservers パラメータのロギング オブザーバのリスト
di.xml の \Magento\staging\Model\Event\Managertype

Answer: C (メッセージを残す)

説明

di.xml ファイルの \Magento\staging\Model\Event\Manager タイプの BandObservers パラメーターにログ オブザーバーを追加することで、ステージング プレビュー機能中に実行されるログ オブザーバーを除外できるようになります。

これにより、将来予定されているキャンペーンの管理パネルからストアフロント カタログ ページをプレビューしているときに、顧客のアクティビティ データが記録されなくなります。

最新問題: 37

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. 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。
- B. 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。
- 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: ([解答を表示する](#))

アーキテクトは、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 ページから分離する必要があります。このアプローチでは、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可能な限り削減します。iframe を使用することにより、決済プロバイダーは顧客の機密データをすべて処理し、Magento はカード所有者のデータを保存または処理しません。これにより、PCI の範囲が縮小され、コンプライアンス プロセスが簡素化されます。オプション B は不正解です。Advanced Encryption Standard (AES-256) アルゴリズムを利用して、支払いモジュールからのすべての顧客機密データを暗号化しても、PCI の範囲は縮小されず、むしろ拡大されます。Magento は引き続きカード所有者のデータを保存および処理しますが、これにはより多くの制御と検証が必要です。選択肢 C は不正解です。認証局 (CA) が発行し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することは、PCI の範囲を縮小するものではなく、むしろデータ送信のセキュリティを確保することになります。Magento は引き続きカード所有者のデータを保存および処理しますが、これにはより多くの制御と検証が必要です。参考: <https://devdocs.magento.com/guides/v2.4/payments-integrations/payment-gateway/integration.html>

最新問題: 38

Adobe Commerce Architect は、通関手数料を処理するための追加コストに適用される新しい合計を追加する、関税と呼ばれる新しい機能を作成します。この拡張機能を使用すると、Adobe Commerce 設定システムを介して Web サイトごとに料金値を個別に指定できます。

アーキテクトは、統合テストで新機能をカバーする予定です。1 つのテスト ケースでは、さまざまな Web サイトで合計が正しく計算されるかどうかを確認する必要があります。

アーキテクトは、テスト構成データがベスト プラクティスに従ってテスト メソッドに追加されていることを確認するにはどうすればよいですか？

- A. `setuo()` メソッドをオーバーライドし、`\Magento\TestFramework\App\config` のインスタンスを受け取り、`setValue()` メソッドで値を指定します。
- B. PHPDoc のテスト メソッドに `@magentoconfigFixture` アノテーションを指定します
- C. Adobe Commerce を設定するためのフィクスチャ ファイルを作成し、テスト メソッド PHPDoc で指定します。
`@magentoconfigFixture` アノテーション

Answer: ([解答を表示する](#))

説明

テスト構成データをテストメソッドに確実に追加するためのベストプラクティスは、テストメソッドの PHPDoc で @magentoconfigFixture アノテーションを使用することです。これにより、アーキテクトは Adobe Commerce を設定するフィクスチャ ファイルを指定できるようになり、テストメソッドはこれらの設定値にアクセスできるようになります。さらに、アーキテクトは setUp() メソッドをオーバーライドして \Magento\TestFramework\App\Config のインスタンスを受け取り、setValue() メソッドを介して値を指定することもできます。

最新問題: 39

販売者は、すぐに使える Adobe Commerce アプリケーションを利用しており、顧客向けに新しいポイントカード機能を追加するよう求めています。コードレビュー中に、Adobe Commerce Architect は、この機能用に作成された reward_card_number 属性設定により、顧客属性がマイアカウント/マイリワードページテンプレートで使用できなくなることに気づきました。

```
SeavSetup = $this->customerSetupFactory->create(['setup' => $this->moduleDataSetup]);
SeavSetup->addAttribute(
    \Magento\Customer\Model\Customer::ENTITY,
    'reward_card_number',
    [
        'type' => 'varchar',
        'label' => 'Customer Community ID',
        'input' => 'text',
        'user_defined' => true,
        'unique' => false,
        'system' => false,
        'is_used_in_grid' => 1,
        'is_visible_in_grid' => 1,
        'is_filterable_in_grid' => 1,
        'is_searchable_in_grid' => 1,
    ]
);
```



顧客属性を正しく設定するには何を追加する必要がありますか？

- A. スコープ プロパティをグローバルの値で追加する必要があります。
- B. グループ プロパティには値 1 を追加する必要があります。
- C. システム プロパティを true の値で追加する必要があります。

Answer: ([解答を表示する](#))

グループ プロパティは、顧客属性が属する属性グループ ID を指定します。グループ プロパティを 1 に設定すると、reward_card_number 属性がデフォルトの属性グループに追加され、マイ アカウント/マイ特典ページテンプレートで使用できるようになります。参考: <https://devdocs.magento.com/guides/v2.4/extension-dev-guide/attributes.html#customer-eav-attribute>

最新問題: 40

アーキテクトは、Adobe Commerce ストアを新しい配送業者と統合する必要があります。カート データは配送業者の API に送信され、価格が取得されて顧客に表示されます。この機能がストアに実装されると、「API がクォータに達し、リクエストが多すぎます」というエラーが返されます。配送業者は、同じ内容のリクエストを API に送信しすぎることについてストアに警告します。

キャリアモデルでは、問題を解決するためにアーキテクトは何を変更する必要がありますか？

- A. リクエストが一致した場合にデータを返す _setCachedQuotes() と _getCachedQuotes() を実装します。
- B. _dopaymentRequest() で、API にリクエストを送信する前に canCollectRates() を呼び出します。

C. `getResponse ()` をオーバーライドし、応答を変数に保存し、応答が存在するかどうかを確認してから戻ります。

Answer: A (メッセージを残す)

キャリア モデルに `setCachedQuotes ()` と `getCachedQuotes()` を実装すると、ストアがカート データをキャッシュに保存できるようになり、新しいリクエストを API に送信する代わりに、同じ内容の繰り返しリクエストをキャッシュから取得できるようになります。これにより、リクエストの数が減り、クォータ制限に達するのを回避できます。

最新問題: 41

Adobe Commerce Architect は、複数の外部商品フィードから価格を取得するスキャナーの開発に取り組んでいます。アーキテクトはベンダーのリストを持っており、新しい構成ファイル `marketplacejeeds.xml` を作成することにしました。

アーキテクトは、個々のファイルとマージされたファイルに対して固有の検証ルールを使用して構成ファイルを検証するために実行できる 3 つの手順はどれですか? (3つお選びください。)

- A. `Config Reader` の `Converter` クラスに検証ルールを実装します。
- B. 構成 XML ファイル内の XSD ファイルに統一リソース名を追加します。
- C. マージされたファイルを検証するためのスキーマを提供します。
- D. 個々のファイルを検証するためのスキーマを提供します。
- E. `\Magento\Framework\Config\DataInterface` を実装するクラスを作成します。
- F. `marketplace.schema.xsd` に検証ルールを作成します。

Answer: D,E,F (メッセージを残す)

説明

アーキテクトが個々のファイルとマージされたファイルに固有の検証ルールを使用して構成ファイルを検証するために実行できる 3 つの手順は、D. 個々のファイルを検証するためのスキーマを提供する、F. `Marketplace.schema.xsd` に検証ルールを作成する、および E. を実装するクラスを作成する

`\Magento\Framework\Config\DataInterface`。個々のファイルを検証するためのスキーマを提供し、`marketplace.schema.xsd` で検証ルールを作成し、実装するクラスを作成することによって、

`\Magento\Framework\Config\DataInterface` を使用すると、アーキテクトは、構成ファイルが個別のファイルとマージされたファイルに対して固有の検証ルールで検証されていることを確認できます。

最新問題: 42

販売者は、カテゴリに対して追加のモバイル画像をアップロードできるようにするために、新しいカテゴリ属性を要求します。

販売者は Adobe Commerce のコンテンツのステージングおよびプレビュー機能を利用しており、この新しいモバイル画像フィールドへの変更をスケジュールして確認したいと考えています。

開発者はデータ パッチを介して属性を作成し、それを

`view/adminhtml/ui_component/category_form.xml`。この属性はメイン フォームのカテゴリに対して表示されますが、スケジュールされた更新が行われる場合は追加フォームには表示されません。

新しいカテゴリの更新をスケジュールするときこの属性を変更するには、アーキテクトは開発者にどのような追加アクションを実行するよう依頼する必要がありますか?

- A. データ パッチ ファイル内で、属性の `apply_to` フィールドが「ステージング」に設定されている必要があります。
- B. 属性も追加する必要があります。

`view/adminhtml/ui_component/catalogstaging_category_update_form.xml`。

C. 属性には、`<item name="allow_staging" xsi:type="boolean">true</item>` が設定されている必要があります。
=category_form.xml ファイルの属性設定セクションにあります。

Answer: B (メッセージを残す)

説明

これは、新しいカテゴリの更新をスケジュールするときに属性を変更するには、その属性を
view/adminhtml/ulcomponent/catalogstagingcategoryupdateform.xml ファイルに追加して、更新をスケジュールするときに追加
フォームに表示されるようにする必要があります。この追加フォームは、更新をスケジュールするときにカテゴリ属性の値
を設定するために使用されます。

最新問題: 43

マーケティング キャンペーンにより、Web サイトで非常に多くの同時注文が発生しており、チェックアウトのパフォーマンス
に影響を与えています。Web サイトは実稼働展開モードになっています。

チェックアウトのパフォーマンスへの影響を軽減するためにアーキテクトが最適化できる 2 つの Web サイト設定はどれで
すか? (2つお選びください。)

- A. 非同期インデックス作成管理パネルの設定 (ストア > 設定 > Ccr.figuraticr. > 詳細 > 開発者 > グリッド設定 > 非同期イン
デックス作成) は、CLI コマンド `tm/magento config:set dev/grid/` を実行することで有効にできます。 `async_ini*xmg 1`
- B. マルチスレッド チェックアウト処理の管理パネル設定 (ストア > 設定 > 構成 > 販売 > チェックアウト > 一般設定 > 非同期)
は、チェックアウト専用で使用される PHP スレッドの数を表すより高い値に設定できます。
- C. 非同期メール通知管理パネルの設定 (ストア > 設定 > 構成 > 販売 > 販売メール > 一般設定 > 非同期) を有効にできます。
- D. 次のコマンドを使用して、新しいデータベースを作成し、データベースの分割機能を自動的に構成できます: `bin/Magento
setup:db-schema:split-sales-sales --host="<checkout db host or ip >" -dbname="<名前>" -username""<チェックアウト DB ユー
ザー名>" -password="<パスワード>"`
- E. Web サイトの展開モードは、CLI コマンド `bin/magentodeploy:mode:set siege` を実行することで `si-g-` に設定できます。ただ
し、同時に発注された注文の数が許容レベルに減少し次第、生産に戻されます。

Answer: A,D (メッセージを残す)

非同期インデックス作成により、Magento はストアフロントのパフォーマンスに影響を与えることなく、バックグラウンドでイン
デクサーを処理できます。データベースを分割すると、Magento はチェックアウト関連のテーブルに別のデータベースを使用
できるようになり、メイン データベースの負荷が軽減され、チェックアウトのパフォーマンスが向上します。参考:

<https://devdocs.magento.com/guides/v2.4/extension-dev-guide/indexing.html#asynchronous-indexing>

<https://devdocs.magento.com/guides/v2.4/config-guide/multi-master/multi-master.html> チェックアウトのパフォーマンスへの影
響を軽減するには、アーキテクトは次の 2 つのを行う必要があります。

非同期インデックス作成の管理パネル設定を有効にします。これにより、Magento はチェックアウト中にオンザフライでイン
デックスを更新するのではなく、cron ジョブを使用してインデックスを更新できるようになります。このオプションは、[スト
ア] > [設定] > [構成] > [詳細設定] > [開発者] > [グリッド設定] > [非同期インデックス作成] にあります。次の CLI コマンドを実行
して有効にすることもできます。 `bin/magento config:set dev/grid/async_indexing 1` 新しいデータベースを作成し、データベー
スの分割機能を使用します。これにより、Magento は売上データをメイン データベースから分離し、データベース サーバーの
負荷を軽減できるようになります。このオプションは、次のコマンドで構成できます: `bin/magento setup:db-schema:`

<https://devdocs.magento.com/guides/v2.4/extension-dev-guide/indexing.html#m2devgde-asynchronous-indexing>

<https://devdocs.magento.com/guides/v2.4/config-guide/multi-master/multi-master.html>

最新問題: 44

Adobe Commerce Architect は、月々の分割払い延長のワークフローをカスタマイズする必要があります。この拡張機能は、独自の従来の拡張機能 (非推奨の支払い方法を使用するモジュール) を持つデフォルト Web サイト PSR と契約しているパートナーからのものです。

分割払いパートナーは支払いの初期化のみを管理し、PSP によって実行されるキャプチャを渡します。金額が正常に取得されると、PSP は IPN を通じて Web サイトに通知します。IPN の目的は、「請求書」を作成し、後で PSP 自体を介した返金リクエストに使用する「キャプチャ情報」を保存することだけです。

アーキテクトは、要求された動作を副作用なしで取得するための最も単純なソリューションを必要としています。

アーキテクトはどのソリューションを実装する必要がありますか？

A. `$invoice->()` の前にプラグインを追加し、その入力を変更して `$payment->capture()` の呼び出しを防止します。

B. `config.xml` の下の支払い方法の `can_capture` 属性を `<can_capture>0</can_capture>` に変更します。

C. `di.zm1` の支払い方法 `CommandPool` に対してタイプ `Magento\payment\Gateway\Command\NullCommand` のキャプチャ コマンドを宣言します。

Answer: ([解答を表示する](#))

要求された動作を副作用なくキャプチャするために Adobe Commerce Architect が実装する最適なソリューションは、`di.xml` で支払い方法 `CommandPool` に対してタイプ `Magento\payment\Gateway\Command\NullCommand` を使用してキャプチャ コマンドを宣言することです。これにより、パートナーは支払いを初期化し、キャプチャを PSP に引き渡すことができると同時に、Web サイトが `$payment->capture()` メソッドを呼び出すことを防ぎます。また、PSP が IPN を通じて Web サイトに通知できるようになり、「請求書」が作成され、後で PSP 自体を介した払い戻しリクエストに使用される「キャプチャ情報」が保存されます。

アーキテクトは、`di.xml` 内の支払い方法 `CommandPool` に対してタイプ `Magento\Payment\Gateway\Command\NullCommand` を使用してキャプチャ コマンドを宣言するソリューションを実装する必要があります。このコマンドは、支払い方法でキャプチャメソッドが呼び出されたときは何も行いません。キャプチャは PSP によって処理されるため、これは望ましい動作です。`NullCommand` クラスは `\Magento\Payment\Gateway\CommandInterface` を実装し、`execute()` メソッドをオーバーライドして `null` を返します。オプション A は不正解です。`$invoice->capture()` メソッドの前にプラグインを追加し、その入力を変更しても、`$payment->capture()` メソッドの呼び出しは妨げられず、渡される請求書オブジェクトが変更されます。オプション B は、構成の下で支払い方法の `can_capture` 属性を変更しているため、不正解です。`xml` を `<can_capture>0</can_capture>` に変更しても、キャプチャメソッドの呼び出しは妨げられませんが、管理パネルのキャプチャ オプションが無効になります。参考:

<https://devdocs.magento.com/guides/v2.4/payments-integrations/base-integration/facade-configuration.html>

最新問題: 45

販売者は、単一ストアビューでネイティブ Adobe Commerce B2B および B2C をサポートする統合 Web サイトを使用しています。

販売者は、B2B 企業アカウントの一部であるログイン ユーザーのサイトのヘッダーに、交渉可能な見積もりや与信限度額などの B2B アカウントの機能をすべてのページに表示したいと考えています。

各 B2B 企業には独自の個別の共有カタログと顧客グループがあり、非 B2B 顧客の多くの顧客グループは変化します。販売者は、これを顧客グループに関連付けないよう要求します。

パブリック データとキャッシュを考慮して、アーキテクトが推奨する 2 つのソリューションはどれですか? (2つお選びください。)

- A. ユーザーが B2B 企業に所属している場合にテーマを切り替えるプラグインを作成し、代替テーマで出力をそれに応じて変更できるようにします。
- B. 現在のユーザーがブロック クラス内の B2B 企業の一員であるかどうかを確認し、それに応じて出力を変更します。
- C. ユーザーが B2B 企業の一員であるかどうかを選択できる顧客セグメントの新しいカスタム条件を作成し、このセグメントを使用してそれに応じて出力を変更します。
- D. 現在のユーザーが顧客セッションの B2B 企業の一員であるかどうかを設定し、そのデータを直接使用してそれに応じて出力を変更します。
- E. 新しい HTTP コンテキスト変数を作成して、B2B 企業のユーザー向けに個別のパブリック コンテンツをキャッシュし、出力をそれに応じて変更できるようにします。

Answer: C,E (メッセージを残す)

説明

C では、ユーザーが B2B 企業の一員であるかどうかを選択できるようにする顧客セグメントの新しいカスタム条件を作成し、このセグメントを使用してそれに応じて出力を変更します。E には、B2B 企業のユーザー向けに個別のパブリック コンテンツをキャッシュできるようにする新しい HTTP コンテキスト変数の作成が含まれ、出力をそれに応じて変更できます。

Valid AD0-E718 Dumps shared by GoShiken.com for Helping Passing AD0-E718 Exam! GoShiken.com now offer the **newest AD0-E718 exam dumps**, the GoShiken.com AD0-E718 exam **questions have been updated** and **answers have been corrected** get the **newest** GoShiken.com AD0-E718 dumps with Test Engine here:

<https://www.goshiken.com/Adobe/AD0-E718-mondaishu.html> (50 Q&As Dumps, **30%OFF** Special Discount:

Freepdfdumps)