

## Adobe.AD0-E716.v2025-04-28.q70

試験コード:	AD0-E716
試験名称:	Adobe Commerce Developer with Cloud Add-on
認定資格:	Adobe
無料問題数:	70
バージョン:	v2025-04-28
アクセス数:	269
ページビュー数:	700
<a href="https://www.jpnpdf.com/Adobe.AD0-E716.v2025-04-28.q70-mondaishu.html">https://www.jpnpdf.com/Adobe.AD0-E716.v2025-04-28.q70-mondaishu.html</a>	

### 最新問題: 1

Adobe Commerce 開発者は、見積もりに関する追加情報を提供するように求められました。複数の見積もりを取得すると拡張属性が返されますが、単一の見積もりを取得すると返されません。拡張属性が欠落している理由の 1 つは何ですか？

- A. 開発者は、etc/extension\_attributes.xml ファイルの属性に、collection="true" を追加するのを忘れました。O attribute code="my\_attributes" type="MyVendor\MyModule\Api\Data\AttributeInterface[]" collection="true" />
- B. 開発者は、Magento\Quote\Api\CartRepositoryInterface::get でプラグインを提供しませんでした。
- C. 開発者は、collection\_load\_after イベントにオブザーバーを実装することを怠りました。

**Answer: A (メッセージを残す)**

collection="true" 属性が etc/extension\_attributes.xml ファイルに設定されていないため、拡張属性がありません。この属性は、見積もりが取得されたときに拡張属性が返される必要があることを Magento に指示します。

この問題を修正するには、開発者は、my\_attributes 拡張属性に collection="true" 属性を追加する必要があります。

<attribute code="my\_attributes" type="MyVendor\MyModule\Api\Data\AttributeInterface[]" collection="true" /> collection="true" 属性が設定されると、見積もりが取得されるときに拡張属性が返されます。

### 最新問題: 2

Adobe Commerce 開発者は、サードパーティ ベンダーのモジュールをインストールしました。このモジュールは、third\_party\_event\_after というカスタム イベントを発生させ、そのイベントをリッスンする third\_party\_event\_after\_observer というオブザーバーも定義します。開発者は、独自のモジュールでこのカスタム イベントをリッスンし、third\_party\_event\_after\_observer オブザーバーの実行が完了した後にオブザーバーのロジックを実行したいと考えています。

開発者は、サードパーティ モジュールによって定義されたオブザーバーの後にオブザーバーが実行されるようにするにはどうすればよいでしょうか。

- A. サードパーティ モジュールが開発者の module.xml ファイルの <sequence> ノードにリストされていることを確認します。
- B. 新しいオブザーバーの並べ替え順序を、サードパーティ ベンダーのオブザーバーの並べ替え順序よりも低くなるように設定します。
- C. 同じイベントをリッスンしているオブザーバーは任意の順序で呼び出される可能性があるため、これは不可能です。

**Answer: A (メッセージを残す)**

Adobe Commerce では、module.xml ファイルでモジュールの依存関係を指定することにより、同じイベントに対してオブザーバーが実行される順序を制御できます。モジュールの module.xml ファイルの <sequence> ノードにサードパーティ モジュールをリストすることにより、同じイベントに対して、モジュールのオブザーバーがサードパーティ モジュールのオブザーバーの後に読み込まれるようになります。

この設定により、オブザーバーがサードパーティ モジュールのオブザーバーの後に実行されるようになります。Adobe Commerce は、イベントの sortOrder 属性ではなく、module.xml ファイルで定義されたモジュール シーケンスに基づいて、オブザーバーの実行順序を解決します。

module.xml の例:

```
<?xml バージョン="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:
magento:framework:モジュール/etc/module.xsd">
<モジュール名="ベンダーモジュール" セットアップバージョン="1.0.0">
<シーケンス>
<モジュール名="サードパーティモジュール"/>
</シーケンス>
</モジュール>
</config>
```

追加リソース:

Adobe Commerce 開発者ガイド: イベント オブザーバー  
モジュール依存関係シーケンス

### 最新問題: 3

Adobe Commerce 開発者が cron ログを確認すると、次のジョブが毎日発生していることがわかります: main。

INFO: Cron Dob inventory\_cleanup\_reservations が正常に終了しました。ただし、データベースの inventory\_reservation テーブルは空になっていません。なぜ inventory\_reservation テーブルにレコードが残っているのでしょうか?

- A. キャンセルされた注文に一致する予約のみが cron ジョブによって削除されます。
- B. 不要になった予約のみが cron ジョブによって削除されます。
- C. マルチソース インベントリの「自動クリーンアップ」機能が構成で無効になっています。

**Answer: B** ([メッセージを残す](#))

inventory\_reservation テーブルにレコードが残っているのは、不要になった予約のみが cron ジョブによって削除されるためです。inventory\_reservation テーブルは、各注文の各製品の数量を追跡し、注文が発注、出荷、キャンセル、または返金されたときに各製品の予約を作成します。最初の予約の数量値は負で、後続の予約の数量値は正です。注文が完了すると、製品のすべての予約の合計はゼロになります。cron ジョブは、合計がゼロの予約のみをテーブルから削除し、未完了の注文に必要な予約はそのまま残します。検証済みの参照: [Magento 2.4 DevDocs] [Magento Stack Exchange]

### 最新問題: 4

配送が遅れる可能性がある製品を識別するために、カート ページの製品タイトルの下に表示されるメッセージを制御するカスタム製品属性を作成するタスクがあります。

新しい EAV 属性 is\_delayed がブール値として作成され、管理パネルと製品ページで正しく機能しています。

\$block->getProduct()->getIsDelayed() などの .phtml カート ページで is\_delayed EAV 属性を使用できるようにする次の実装は何でしょうか?

A)

新しいファイル etc/catalog\_attributes.xmi を作成します。

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Catalog:etc/catalog_attributes.xsd">
  <group name="quote_item">
    <attribute name="is_delayed" />
  </group>
</config>
```

B)

新しいファイル etc/extension attribute.xmi を作成します。

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:Api/etc/extension_attributes.xsd">
    <extension_attributes for="Magento\Catalog\Api\Data\ProductRenderInterface">
        <attribute code="is_delayed" type="bool" />
    </extension_attributes>
</config>
```

新しいファイル etc/eav attribute.xmi を作成します。

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Eav/etc/eav_attributes.xsd">
    <entity type="quote_item">
        <attribute code="is_delayed">
            <field code="is_visible" locked="true" />
        </attribute>
    </entity>
</config>
```

A. オプションA

B. オプションB

C. オプションC

**Answer: A (メッセージを残す)**

is\_delayed EAV 属性を .phtml カート ページで使用できるようにするには、開発者は etc/catalog\_attributes.xmi という新しいファイルを作成する必要があります。このファイルには、is\_delayed 属性の定義が含まれます。

次のコードは、etc/catalog\_attributes.xmi ファイルを作成する方法を示しています。

テキスト

```
<?xml バージョン="1.0"?>
```

```
<カタログ属性>
```

```
<属性コード="is_delayed" タイプ="int">
```

```
<label>遅延しています</label>
```

```
<note>この属性は、製品が遅延しているかどうかを示します。</note>
```

```
<ソート順序>10</ソート順序>
```

```
<必須>偽</必須>
```

```
</属性>
```

```
</カタログ属性>
```

etc/catalog\_attributes.xmi ファイルが作成されると、is\_delayed 属性が .phtml カート ページで使用できるようになります。この属性には、Product クラスの getIsDelayed() メソッドを使用してアクセスできます。

PHP の

```
$block から製品を取得します。
```

```
$isDelayed = $product->getIsDelayed();
```

isDelayed 変数には、is\_delayed 属性の値が含まれます。属性の値が 1 の場合、製品は遅延しています。属性の値が 0 の場合、製品は遅延していません。

**最新問題: 5**

Adobe Commerce 開発者は、PageBuilder スライダー コンテンツ タイプを変更して、新しいカスタム コンテンツ タイプ (スライド以外) を子として割り当てられるようにするよう依頼されました。開発者は、モジュール内に improved\_slide という新しいコンテンツ タイプを既に作成しています。新しいコンテンツ タイプをスライダー コンテンツ タイプの子として割り当てられるように、モジュール内に新しい view/adminhtml/pagebuilder/content\_type/slider.xml ファイルを作成する必要があります。

これを実現するための正しい XML は何ですか？

```
<type name="slider">
  <children>
    <child name="improved_slide" policy="allow"/>
  </children>
</type>
```

A.

```
<type name="slider">
  <allowed_descendants>
    <descendant name="improved_slide" />
  </allowed_descendants>
</type>
```

B.

```
<type name="slider">
  <arguments>
    <argument name="allowed_children" xsi:type="array">
      <item name="improved_slide" xsi:type="string">improved_slide</item>
    </argument>
  </arguments>
</type>
```

C.

Answer: B (メッセージを残す)

次のXMLにより、新しいコンテンツタイプをスライダーコンテンツタイプの子にすることができます。

<ページビルダーコンテンツタイプ>

<type>スライダー</type>

<子供達>

<type>改善されたスライド</type>

</子供>

</ページビルダー\_コンテンツ\_タイプ>

コードは慎重に使用してください。

このXMLは、スライダーコンテンツタイプが、子として improved\_slide コンテンツタイプを持つことができることを Magento に伝えます。

最新問題: 6

Adobe Commerce 開発者はカスタム モジュールを開発しています。実装の一環として、Custom\Module\Model\Example クラスのすべてのインスタスが Magento\Filesystem\Adapter\Local の新しいインスタンスを受け取ることを決定しました。

開発者は di.xml を使用してこれをどのように実現するのでしょうか？

```
<type name="Custom\Module\Model\Example">
  <arguments>
    <argument name="adapter" xsi:type="object" shared="false">Magento\Filesystem\Adapter\Local</argument>
  </arguments>
</type>
```

A.

```
<type name="Custom\Module\Model\Example">
  <arguments>
    <argument name="adapter" xsi:type="object" singleton="false">Magento\Filesystem\Adapter\Local</argument>
  </arguments>
</type>
```

B.

```
<type name="Custom\Module\Model\Example">
  <arguments>
    <argument name="adapter" xsi:type="object" transient="true">Magento\Filesystem\Adapter\Local</argument>
  </arguments>
</type>
```

C.

**Answer: B (メッセージを残す)**

開発者は、di.xml ファイルに次の構成を追加することでこれを実現できます。

テキスト

<設定>

<コンポーネント名="Custom\Module\Model\Example" ファクトリー="Custom\Module\Model\ExampleFactory">

<引数>

<引数名="filesystemAdapter" タイプ="Magento\Filesystem\Adapter\Local" />

</引数>

</コンポーネント>

</config>

この構成により、Custom\Module\Model\Example クラスのすべてのインスタンスが Magento\Filesystem\Adapter\Local クラスの新しいインスタンスを受け取るようになります。

最新問題: 7

販売者は、Adobe Commerce Cloud Pro プランの統合環境でパフォーマンスの問題が発生しており、拡張統合環境にアップグレードしたいと考えています。

拡張統合環境にアップグレードするために再デプロイする前に必要な手順は何ですか？

- A. 1. 統合ブランチの数を2つに制限する
- 2. アップグレードを要求するサポートチケットを送信する
- B. 1. 統合ブランチの数を3つに制限する
- 2. .magento.env.yamlのENV.ENVIRONMENTをENHANCEDINTEGRATIONに設定する
- C. 1. 統合ブランチの数を4つに制限する
- 2. クラウドGUIで統合環境を設定し、拡張スイッチをオンに設定します。

**Answer: A (メッセージを残す)**

Adobe Commerce Cloud の拡張統合環境にアップグレードするには、統合ブランチの制限の管理や Adobe サポートとの調整など、アップグレードに向けて環境が準備されていることを確認するための特定の手順が必要です。

\* 統合ブランチの制限:

\* Adobe Commerce Cloud の拡張統合環境では、アクティブな統合ブランチが最大 2 つまでしかサポートされません。アップグレード前にこの要件を満たす必要があります。

\* サポートチケットの送信:

\* これにはインフラストラクチャの変更が伴うため、アップグレードをリクエストし、Adobe のサポート チームから支援を受けるには、サポート チケットを送信する必要があります。

\* オプションAが正しい理由:

\* このオプションでは、2 つのブランチの制限やサポート チケットのプロセスなど、適切な手順が概説されています。オプション B と C では、.magento.env.yaml を変更するか、GUI を介して構成することが提案されていますが、アップグレードには Adobe からの直接サポートが必要なため、このシナリオではこれらは適用されません。

\* 参考文献:

\* 拡張統合環境に関する Adobe Commerce Cloud ドキュメント

最新問題: 8

Adobe Commerce 開発者が顧客グリッドに新しい列を正常に追加しました。この列の内容をグリッドに表示する前に、データをフォーマットする必要があります。

ベストプラクティスによれば、開発者は列をレンダリングするためのカスタムロジックをどのように追加するのでしょうか？

- A. 1. プラグインforMagento\Ui\Component\Listing\Columns\Column::prepareColumn() を作成します。
- 2. afterPreparecolumn メソッド内にカスタム ロジックを追加します。
- B. 1. Magento\Ui\Component\Listing\Columns\Column を拡張するカスタム クラスを作成します。
- 2. prepareDataSource メソッド内にカスタム ロジックを追加します。
- 3. モジュールの customer\_listing.xml 内の列ノードに属性クラスを追加します。
- C. 1. 設定を使用して Magento\Customer\Ui\Component\DataProvider クラスをオーバーライドします。
- 2. getData() メソッドをオーバーライドし、行ごとにカスタム ロジックを追加します。

**Answer: B (メッセージを残す)**

グリッド列のデータフォーマットにカスタムロジックを追加するベストプラクティスは、新しいクラスを作成して拡張することです。

Magento\Ui\Component\Listing\Columns\Column。prepareDataSource メソッドは、UI コンポーネントに表示される前にデータを処理およびフォーマットするために設計されています。

\* カスタム列クラスで prepareDataSource を使用する:

\* Magento\Ui\Component\Listing\Columns\Column を拡張すると、prepareDataSource メソッドにアクセスでき、必要に応じてデータを操作できるようになります。

\* カスタム クラスを追加すると、再利用性とモジュール化コードが可能になり、Magento のアーキテクチャに沿ったものになります。

\* オプション B が正しい理由:

\* このオプションは Magento の UI コンポーネント構造を適切に使用し、グリッド データ操作のクラスとメソッドに重点を置いています。オプション A には不要なプラグインが含まれており、DataProvider 設定のオプション C は、単純な UI 変更には一般的に推奨されません。

\* 実装手順:

\* Column クラスを拡張し、prepareDataSource メソッドにロジックを追加します。

\* 次に、customer\_listing.xml で、class 属性を使用して <column> ノード内でこのクラスを参照します。

\* 参考文献:

\* Magento UI コンポーネントのカスタム列作成ガイド

\* Magento\Ui\Component\Listing\Columns\Column の Adobe Commerce ドキュメント

## 最新問題: 9

Adobe Commerce開発者は、フロントエンドコントローラーにデータを送信するカスタムフォームを作成するという課題を抱えています。彼らはアクションを作成することを決定し、Magento\Framework\App\Action\HttpPostActionInterface クラスを使用していますが、データベースにデータが保存されておらず、送信後にフロントエンドにエラー メッセージが表示されます。

デバッグしてデータ永続ロジックが正しいことを確認した後、この問題の原因と解決策は何でしょうか？

A. Magento はフロントエンド コントローラーへの POST リクエストを許可しないため、送信機能を API エンドポイントとして書き直す必要があります。

B. 開発者はアクションにvalidatePostDataQメソッドを実装し忘れました。このメソッドを実装する必要があります。検証されていないすべてのPOSTデータはリクエストから削除され、エラーがスローされます。

C. フォーム キーの検証はすべての非 AJAX POST リクエストで実行されるため、開発者はリクエストに for\_key を追加する必要があります。

**Answer: C (メッセージを残す)**

Magento Stack Exchange の回答によると、フォーム キー検証は、リクエスト内のフォーム キーが Magento によって生成されたものと一致するかどうかを確認することで CSRF 攻撃を防ぐセキュリティ機能です。開発者がカスタム フォームに form\_key を含めない場合、検証は失敗し、エラーが表示されます。したがって、開発者は <?> を使用してリクエストに form\_key を追加する必要があります。

テンプレート ファイルに \$block->getBlockHtml ('formkey') ?> を追加します。検証済みの参照:

<https://magento.stackexchange.com/questions/95171/magento-2-form-validation>

#### 最新問題: 10

インデクサーに関する問題を調査しているときに、Adobe Commerce 開発者がログに「一時テーブルに割り当てられたメモリ サイズが innodb\_buffer\_pool\_size の 20% を超えています」のようなエラーを確認しています。クライアントで innodb\_buffer\_pool\_size を更新するか、バッチ サイズの値を減らすことをお勧めします。

バッチ サイズの値を小さくするとパフォーマンスが向上するのはなぜですか？

- A. これにより、一時テーブルのメモリ使用量が削減されます。
- B. これにより、バッチプロセスごとに長いタイムアウトが可能になります。
- C. これにより、プロセス中にさらに多くの PHP スレッドを利用できるようになります。

**Answer: A** ([メッセージを残す](#))

バッチ サイズの値を小さくすると、一時テーブルのメモリ使用量が減り、パフォーマンスが向上します。

バッチ サイズの値は、インデクサーが一度に処理するデータ行数を決定します。バッチ サイズの値が大きすぎると、一時テーブルに割り当てられたメモリ サイズが innodb\_buffer\_pool\_size の 20% を超え、エラーが発生してインデックス作成プロセスが遅くなる可能性があります。バッチ サイズの値を小さくすると、インデクサーはデータをより効率的に処理し、メモリの問題を回避できます。検証済みの参照: [Magento 2.4 DevDocs] [Magento Stack Exchange]

#### 最新問題: 11

国際的な販売業者は、商品全体をインポートした後、変更がフロントエンドに反映されるまでに時間がかかりすぎると不満を述べています。

Adobe Commerce 開発者は、データベースの問題である可能性があると考え、次のエンティティ数を収集します。

- \* カテゴリ: 900
- \* 製品数: 30万
- \* 顧客数: 70万人
- \* 顧客グループ: 106
- \* 注文数: 160万
- \* 請求書: 500k
- \* クレジットメモ: 50k
- \* ウェブサイト: 15
- \* 店舗数: 45

考えられる原因は何でしょうか？

- A. 製品、カテゴリ、店舗の数の組み合わせが大きすぎます。これにより、フラット カタログ インデックスに大量の値が格納され、通常で処理するには大きすぎます。
- B. 注文、顧客、請求書、クレジットメモの数の組み合わせが大きすぎます。これにより、顧客グリッド インデックスに大量の値が格納され、通常で処理するには大きすぎます。
- C. 製品、顧客グループ、Web サイトの数の組み合わせが大きすぎます。これにより、価格インデックスに保存される値が膨大になり、通常で処理できなくなります。

**Answer: (解答を表示する)**

完全な製品インポート後にフロントエンドで変更が反映されるまでに時間がかかる原因として考えられるのは、製品、顧客グループ、ウェブサイトの数の組み合わせです。これにより、価格インデックスに膨大な量の値が格納され、通常で処理できないほど大きくなります。価格インデックスは、税金、割引、カタログ価格ルールなどのさまざまな要素を考慮して、顧客グループとウェブサイトごとに各製品の最終価格を計算します。製品、顧客グループ、ウェブサイトの数が多い場合、価格インデックスは非常に複雑になり、更新に時間がかかります。検証済みの参照: [Magento 2.4 DevDocs] [Magento Stack Exchange]

#### 最新問題: 12

Adobe Commerce 開発者は、B2C および B2B Web サイトを含む Magento 2 インスタンスで作業しています。各 Web サイトには、英語、ウェールズ語、フランス語のユーザー向けの 3 つの異なるストア ビューが含まれています。開発者は、サイト全体で使用される汎用リンク テンプレートを使用して、B2C Web サイトと B2B Web サイトの間にリンクを追加することを任されていますが、ストア ビューに関係なく、これらのリンクを英語で表示したいと考えています。

開発者は、レンダリング前にこのテンプレートで使用するカスタム ブロックを作成し、次のコードを使用して翻訳ロケールを設定し、環境エミュレーションを開始します。

```
/** @var $this->_translate \Magento\Framework\TranslateInterface */
$this->_translate->setLocale($newLocaleCode);

/** @var $this->_emulation \Magento\Store\Model\App\Emulation */
$this->_emulation->startEnvironmentEmulation($storeId, \Magento\Framework\App\Area::AREA_FRONTEND);
```

テンプレートのテキストが各ストアの言語にまだ翻訳中であることがわかりました。なぜこのようなことが起こるのでしょうか？

**A.** startEnvironmentEmulation() は、setLocale() オプションの 2 番目の \$lock パラメータ、つまり setLocale(\$newLocaleCode, true) を使用してロケールを設定およびロックし、エミュレートされたストアのロケールをオーバーライドしてロックします。これが最初に設定およびロックされている場合、環境エミュレーションはこれをオーバーライドできません。

**B.** startEnvironmentEmulation() は、エミュレートされたストアの 1 つに翻訳ロケールをリセットします。これは、上記のように setLocale と startEnvironmentEmulation の順序が使用されるときに開発者が設定したロケールをオーバーライドします。

**C.** setLocale() は、翻訳ロケールを最初に設定した後には変更しません。

\$this->\_translate->emulate(\$newLocaleCode) メソッドは、新しいロケールを現在の emulatedLocales スタックの先頭にプッシュすることで、これを一時的に変更するために存在します。

**Answer: B** ([メッセージを残す](#))

startEnvironmentEmulation() メソッドは、エミュレートされたストアの 1 つに翻訳ロケールをリセットします。これにより、上記のように setLocale() と startEnvironmentEmulation() の順序が使用されるときに、開発者が設定したロケールがオーバーライドされます。

望ましい結果を得るための正しい方法は、emulate() メソッドを使用して翻訳ロケールを一時的に変更することです。次のコードは、その方法を示しています。

PHP の

```
$this->_translate->emulate('en_US');
```

```
// テンプレートをレンダリングする
```

```
$this->_translate->revert();
```

このコードは、テンプレートをレンダリングする前に翻訳ロケールを英語に設定し、テンプレートがレンダリングされた後にロケールをデフォルト値に戻します。

startEnvironmentEmulation() メソッドは、異なるストア ビューまたは Web サイトをエミュレートするために使用されます。これは、テスト目的や、さまざまな環境で動作する必要がある機能の開発に役立ちます。

emulate() メソッドは、翻訳ロケールを一時的に変更するために使用されます。これは、特定の言語でテンプレートをレンダリングする場合や、異なる言語で動作する必要がある機能をテストする場合に役立ちます。

#### 最新問題: 13

開発者は Adobe Commerce Cloud ステージング環境に新しいリリースを展開したいと考えていますが、まずは実稼働環境から最新のコードを入手する必要があります。

開発者はステージング環境を更新するために何をすべきでしょうか？

**A.** 1. プロジェクト Web インターフェイスにログインします。

2. ステージング環境を選択し、同期をクリックします。

**B.** 1. プロジェクト Web インターフェイスにログインします。

2. ステージング環境を選択し、マージをクリックします。

**C.** 1. 本番環境へのチェックアウト

2. magento-cloud synchronize <environment-ID> コマンドを使用する

**Answer: A** ([メッセージを残す](#))

#### 最新問題: 14

モジュールの di.xml ファイルは、クラス Action の 2 つのプラグインを添付します。

```
<config>
  <type name="Magento\Framework\App\Action\Action">
    <plugin name="vendor_module_plugina" type="Vendor\Module\Plugin\PluginA" sortOrder="10" />
    <plugin name="vendor_module_pluginb" type="Vendor\Module\Plugin\PluginB" sortOrder="20" />
  </type>
</config>
```

around プラグインのコードは次のとおりです。

```
class PluginA
{
    public function aroundDispatch(\Magento\Framework\App\Action\Action $subject, $next, $request)
    {
        // custom code
        return $request;
    }
}
```

プラグインの実行順序はどうなりますか？

- PluginA::beforeDispatch()
- PluginA::aroundDispatch()
- PluginB::beforeDispatch()
- Action::dispatch()
- PluginB: afterDispatch()
- A. PluginA::aroundDispatch()
- PluginA::beforeDispatch()
- PluginA::aroundDispatch()
- PluginA: afterDispatch()
- PluginA::beforeDispatch()
- PluginA::aroundDispatch()
- B. PluginB::beforeDispatch()

- C. Action::dispatch()
- PluginB: afterDispatch()
- PluginA::aroundDispatch()
- PluginA::afterDispatch()

**Answer: C** (メッセージを残す)

プラグインの実行順序は次のとおりです。

プラグインA::beforeDispatch()  
プラグインB::beforeDispatch()  
PluginA::aroundDispatch()

aroundプラグインのコード  
プラグインB::afterDispatch()  
プラグインA::afterDispatch()

aroundDispatch() メソッドは別のスコープで実行されるため、around プラグインのコードは、両方のプラグインの beforeDispatch() メソッドの後、両方のプラグインの afterDispatch() メソッドの前に実行されます。

プラグインの実行順序を示す図を以下に示します。

プラグインA  
ディスパッチ前()  
ディスパッチ()  
afterDispatch()  
プラグインB  
ディスパッチ前()  
afterDispatch()

#### 最新問題: 15

Adobe Commerce Cloud プロジェクトでは、2つの新しい支払いモジュールをインストールした拡張統合環境を使用しています。

開発者は Cloud CLI for Commerce ツールを使用しています。

開発者は統合環境でこの新しい機能をテストするために何をすべきでしょうか？

- A. 1. 統合環境ブランチの1つを複製します。
- 2. 統合から新しいアクティブ ブランチを作成し、モジュールをインストールします。
- 3. 変更をプッシュします。
- B. 1. 統合から新しいブランチを作成し、モジュールをインストールします。
- 2. 変更をプッシュします。
- 3. ブランチのアクティブステータスのチェックは必要ありません。
- C. 1. アクティブな統合環境ブランチの1つを非アクティブ化します。
- 2. 統合から新しいアクティブ ブランチを作成し、モジュールをインストールします。
- 3. 変更をプッシュします。

**Answer:** [\(解答を表示する\)](#)

開発者は、アクティブな統合環境ブランチの1つを非アクティブ化し、統合から新しいアクティブなブランチを作成してモジュールをインストールし、変更をプッシュすることで、統合環境で新機能をテストできます。これは、拡張統合環境では一度にアクティブなブランチが4つに制限されており、各ブランチには専用のデータベースとサービスがあるためです。開発者は、Cloud CLI for Commerce ツールを使用してブランチを管理し、コードの変更をデプロイできます。検証済みリファレンス: [Magento 2.4 DevDocs] 1

#### 最新問題: 16

Adobe Commerce 開発者が、特定の注文を外部の会計システムにエクスポートするプロセスを作成しました。このプロセスは、Magento CLI を使用してコマンド `php bin/magento my_module:order:process --order_id=<order_id>` で起動する必要があります。

例: `php bin/magento my_module:order:process --order_id=1245`。

コマンドを設定する正しい方法は何かですか？

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    parent::configure();
}

protected function values()
{
    return [new InputValue('order_id', InputValue::REQUIRED, 'Order ID')];
}
```



A.

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addOption('order_id', null, InputOption::VALUE_REQUIRED, 'Order ID');
    parent::configure();
}
```



B.

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addOption('order_id', null, InputOption::VALUE_REQUIRED, 'Order ID');

    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addArgument('order_id', InputArgument::REQUIRED, 'Order ID');
    parent::configure();
}
```



C.

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addArgument('order_id', InputArgument::REQUIRED, 'Order ID');
    parent::configure();
}
```

D.

Answer: C (メッセージを残す)

Magento コマンドライン インターフェース (CLI) ガイドの使用法によると、Magento CLI を設定して実行するには、開発者は bin/magento ファイルを実行可能にして、それを使用してコマンドを実行する必要があります。カスタム コマンドを作成するには、開発者は \Symfony\Component\Console\Command\Command を実装するクラスを作成し、その名前、説明、引数、オプション、およびロジックを定義する必要があります。開発者は、Magento\Framework\Console\CommandList 引数を使用して、モジュールの di.xml ファイルにコマンドを登録する必要もあります。したがって、オプション C は、カスタム コマンドを設定するための正しいクラスと di.xml コードを示しているため、正解です。検証済みの参照: <https://www.a2hosting.com/kb/installable-applications/optimization-and-configuration/magento1/using-the-magento-command-line-interface-cli/>

有効な **AD0-E716** 問題集は GoShiken.com が提供された合格しやすい AD0-E716 試験問題集！ GoShiken.com が最新の **AD0-E716** 試験問題集を提供しています。GoShiken.com AD0-E716 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-E716 問題集をゲットする人はこちら: <https://www.goshiken.com/Adobe/AD0-E716-mondaishu.html> (7230%OFF問題集溶と正解付きで 30%w特別割引コード: **Freepdfdumps**)

Adobe Commerce 開発者は、管理パネルのカスタム フォームにファイル フィールドを追加するタスクを負っています。このフィールドでは、サイズが 2 MB 以下の .PDF ファイルのみを受け入れる必要があります。これまでに、開発者は、フォーム コンポーネント xml ファイルの fieldset ノード内に次のコードを追加しました。

```
<field name="pdf_file" formElement="fileUploader">
  <formElements>
    <fileUploader>
      <settings>
        <uploaderConfig>
          <param xsi:type="string" name="url">myvendor_mymodule/customForm/uploadPdf</param>
        </uploaderConfig>
      </settings>
    </fileUploader>
  </formElements>
</field>
```

開発者は検証をどのように実装するのでしょうか？

A)

HyVendor\MyModule\Controller\Adminhtml\CustomEntity\UploadPdf コントローラー内に検証を追加する

```
public function execute()
{
    $file = $this->fileUploaderFactory->create($this->getRequest()->getPdfFile());
    if($file->getExtension() == 'pdf') {
        throw new InvalidFileException(__('The file must be PDF.'));
    }
    if($file->getSize() >= '2048000') {
        throw new InvalidFileException(__('The file size must be less or equal than 2MB'));
    }
    return $this->resultFactory->create(ResultFactory::TYPE_PAGE);
}
```

B)

モジュールの di.xml 内に、コンストラクターの allowedExtensions と maxFileSize を指定する仮想型 forMyvendor\MyModule\Model\customPdfuploader を追加します。

```
<type name="MyVendor\MyModule\Model\CustomPdfUploader">
  <arguments>
    <argument name="allowedExtensions" xsi:type="string">pdf</argument>
    <argument name="maxFileSize" xsi:type="number">2048000</argument>
  </arguments>
</type>
```

C)

<settings> ノード内に次のコードを追加します。

```
<allowedExtensions>pdf</allowedExtensions>
<maxFileSize>2048000</maxFileSize>
```

A. オプションA

B. オプションB

C. オプションC

**Answer: B** ([メッセージを残す](#))

開発者は、モジュールの di.xml 内で、コンストラクターの allowedExtensions と maxFileSize を指定して、Myvendor\MyModule\Model\customPdfuploader の仮想型を追加できます。この方法では、開発者は既存のファイル アップローダー クラスを再利用し、コア コードを変更せずに特定のフィールドに合わせてカスタマイズできます。検証済みの参照: [Magento 2.4 DevDocs] [Magento Stack Exchange]

最新問題: 18

Adobe Commerce 開発者は、PDF と画像を無料でダウンロードするためのカスタム モジュールの追跡レベルを変更するように求められています。

モジュールには次のモデルが含まれています。  
ベンダー\無料ダウンロード\モデル\ダウンロード

Vendor\FreeDownload\Model\DownloadPdf は Vendor\FreeDownload\Model\Download を拡張します。Vendor\FreeDownload\Model\DownloadImage は Vendor\FreeDownload\Model\Download を拡張します。ダウンロード クラスには、tracking\_level のパラメータがあります。

開発者は、di.xml で、Download クラスおよび Download を拡張するすべてのクラスの値を 4 に設定するには、tracking\_level パラメータをどのように構成しますか？

A. Configure the parameter on a child class and add parent attribute as it will be propagated to siblings and parent.

```
<type name="Vendor\FreeDownload\Model\DownloadPdf" parent="Vendor\FreeDownload\Model\Download">
  <arguments>
    <argument name="tracking_level" xsi:type="integer">4</argument>
  </arguments>
</type>
```




B. Configure the parameter on the all child classes and set the parent attribute on one of them.

```
<type name="Vendor\FreeDownload\Model\DownloadPdf" parent="Vendor\FreeDownload\Model\Download">
  <arguments>
    <argument name="tracking_level" xsi:type="number">4</argument>
  </arguments>
...
<type name="Vendor\FreeDownload\Model\DownloadImage">
  <arguments>
    <argument name="tracking_level" xsi:type="number">4</argument>
  </arguments>
...
```



C. Configure the parameter on parent class, as it will be propagated on descendant classes.

```
<type name="Vendor\FreeDownload\Model\Download">
  <arguments>
    <argument name="tracking_level" xsi:type="number">4</argument>
  </arguments>
</type>
```



**Answer: C (メッセージを残す)**

親クラスのパラメータを構成して、それがすべての子孫クラスに伝播されるようにするには、di.xml 内で親クラスにパラメータを定義するのが正しい方法です。この方法では、明示的にオーバーライドされない限り、この親から継承するすべての子クラスはパラメータ値を自動的に使用します。

オプション C が正解である理由は次の通りです。

\* 親クラス (Vendor\FreeDownload\Model\Download) での構成: Download クラスで直接、tracking\_level パラメータを設定すると、DownloadPdf や DownloadImage など、このクラスを拡張するすべてのクラスが、tracking\_level パラメータ値を継承するようになります。このメソッドは、Magento の依存性注入構成を活用し、親クラスで設定されたパラメータを子クラスにカスケードできるようにします。

\* 説明: Magento の依存性注入システムでは、親クラス レベルで設定されたパラメータは、より具体的なレベルで上書きされない限り、すべての子クラスで使用できます。親クラスで tracking\_level を定義すると効率的であり、すべてのサブクラス間で一貫性が確保されます。

\* 参考資料: Magento の公式開発者ドキュメントでは、上位レベルの di.xml で定義されたクラスの依存関係と構成パラメータは、子孫クラスからアクセス可能であると説明されています。

これは、クラスの階層に影響するパラメータを設定するための Magento の標準的な方法です。

\* 冗長構成の回避: 個々の子クラスにパラメータを設定するオプション A や、複数の子クラスにパラメータを冗長に設定するオプション B とは異なり、オプション C は構成を一元化するため最適です。これにより、矛盾のリスクが軽減され、メンテナンスが簡素化されます。

オプション A と B は次の理由で不正解です。

\* オプション A は、単一の子クラスに対してパラメータを構成します。これは、DownloadImage などの他の子クラスには影響しません。

\* オプション B は、各子クラスのパラメータを個別に冗長的に設定しますが、パラメータを親から継承できる場合は不要です。

#### 最新問題: 19

Adobe Commerce 開発者は、カスタム ブランド エンティティを管理するモジュールに取り組んでおり、SearchCriteria を使用して次の SQL クエリを複製したいと考えています。

```
$filter1->setField('featured')
->setValue(1)
->setConditionType('eq');

$filter2->setField('logo_image')
->setConditionType('notnull');

$filterGroup1->setFilters([$filter1, $filter2]);

$filter3->setField('enabled')
->setValue(1)
->setConditionType('eq');

$filterGroup2->setFilters([$filter3]);
```

A. \$searchCriteria->setFilterGroups(['\$filterGroup1, \$filterGroup2]);

```
$filter1->setField('featured')
->setValue(1)
->setConditionType('eq');

$filter2->setField('logo_image')
->setConditionType('notnull');

$filter3->setField('enabled')
->setValue(1)
->setConditionType('eq');

$searchCriteria->setFilter($filter3);
->setOrFilter([$filter1, $filter2]);
```

B.

```
$filter1->setField('enabled')
->setValue(1)
->setConditionType('eq');

$searchCriteria->setFilter($filter1);
$filter2->condition('featured = ? OR logo_image ?', [1, 'IS NOT NULL'])
$searchCriteria->setSelectFields($filter2);
```

C.

**Answer: A** ([メッセージを残す](#))

正解はオプション A です。このアプローチでは、2 つのフィルター グループを使用して、Magento の SearchCriteria API を使用した SQL クエリからのロジックを正しく実装します。

\* ロジックの理解: 指定された SQL クエリには、2 つの主な条件が含まれています。

\* featured = 1 AND logo\_image IS NOT NULL (AND条件でグループ化)

\* 有効 = 1

Magento SearchCriteria の用語では、これらは次のように翻訳されます。

\* フィルターグループ 1: featured = 1 かつ logo\_image が NULL ではない

\* フィルターグループ 2: 有効 = 1

次に、暗黙の AND 条件を使用して 2 つのフィルター グループが結合されます。

\* MagentoのSearchCriteria構造: SearchCriteriaオブジェクトはMagentoで推奨される方法です

2 複雑な条件でコレクションをフィルタリングして取得します。

\* フィルター グループ: フィルターをグループ化するために使用されます。フィルター グループ内のすべてのフィルターは AND 条件で結合されます。

\* フィルター: 単一のフィールドの条件を定義します。

\* オプションAのコードの説明:

\* 注目フィールドに対してフィルター 1 が作成され、eq 条件タイプで 1 に設定されます。

\* フィルター 2 は、notnull 条件タイプを使用して、logo\_image フィールドに対して作成されます。

\* フィルター 1 とフィルター 2 はフィルター グループ 1 に結合され、これらのフィルター間に AND 条件が適用されます。

\* 有効なフィールドに対してフィルター 3 が作成され、eq 条件タイプで 1 に設定されます。

\* フィルター グループ 2 にはフィルター 3 のみが含まれます。

\* 最後に、両方のフィルター グループが SearchCriteria オブジェクトに設定されます。デフォルトでは、SearchCriteria は AND 条件を使用してフィルター グループを結合します。

\* オプション A が正しい理由: オプション A では、目的の SQL クエリ ロジックに正確に一致する 2 つのフィルター グループを使用します。

\* featured と logo\_image を AND 条件で正しく結合します。

\* 次に、enabled が 1 であるかどうかを確認します。これは、2 番目のフィルター グループによって処理されます。

\* 参考文献:

\* 検索条件ビルダー - この Adobe Commerce ドキュメント ページでは、SearchCriteriaBuilder の作成方法と使用方法について説明します。

\* フィルター グループと条件 - SearchCriteria でフィルター グループと条件を構成する方法の概要を示します。

\* Magento 2 API - SearchCriteria を含む Magento 2 API の操作に関する一般的なドキュメント。

このアプローチは、複数のフィルター グループと条件を持つ SearchCriteria を使用して複雑な SQL クエリを複製するための Adobe Commerce の標準的な方法です。

## 最新問題: 20

Adobe Commerce 開発者はカスタム モジュールを開発しています。実装の一環として、Custom\Module\Model\Example クラスのすべてのインスタスが Magento\Filesystem\Adapter\Local の新しいインスタンスを受け取ることを決定しました。

開発者は di.xml を使用してこれをどのように実現するのでしょうか?

A. 

```
<type name="Custom\Module\Model\Example">
  <arguments>
    <argument name="adapter" xsi:type="object" shared="false">Magento\Filesystem\Adapter\Local</argument>
  </arguments>
</type>
```

B. 

```
<type name="Custom\Module\Model\Example">
  <arguments>
    <argument name="adapter" xsi:type="object" singleton="false">Magento\Filesystem\Adapter\Local</argument>
  </arguments>
</type>
```

```
<type name="Custom\Module\Model\Example">
  <arguments>
    <argument name="adapter" xsi:type="object" transient="true">Magento\Filesystem\Adapter\Local</argument>
  </arguments>
</type>
```

C.

**Answer: B (メッセージを残す)**

開発者は、di.xml ファイルに次の構成を追加することでこれを実現できます。

テキスト

<設定>

<コンポーネント名="Custom\Module\Model\Example" ファクトリー="Custom\Module\Model\ExampleFactory">

<引数>

<引数名="filesystemAdapter" タイプ="Magento\Filesystem\Adapter\Local" />

</引数>

</コンポーネント>

</config>

この構成により、Custom\Module\Model\Example クラスのすべてのインスタンスが Magento\Filesystem\Adapter\Local クラスの新しいインスタンスを受け取るようになります。

**最新問題: 21**

Adobe Commerce 開発者は、データをフロントエンド コントローラーに送信するカスタム フォームを作成する任務を負っています。アクションを作成することを決定し、\Magento\Framework\App\Action\HttpPostActionInterface クラスを実装しましたが、データベースにデータが保持されず、送信後にフロントエンドにエラー メッセージが表示されます。

デバッグしてデータ永続ロジックが正しいことを確認した後、この問題の原因と解決策は何でしょうか？

A. Magento はフロントエンド コントローラーへの POST リクエストを許可しないため、送信機能を API エンドポイントとして書き直す必要があります。

B. 開発者はアクションにvalidatePostDataQメソッドを実装し忘れました。このメソッドを実装する必要があります。検証されていないすべてのPOSTデータはリクエストから削除され、エラーがスローされます。

C. フォーム キーの検証はすべての非 AJAX POST リクエストで実行されるため、開発者はリクエストに for\_key を追加する必要があります。

**Answer: C (メッセージを残す)**

Magento Stack Exchange の回答によると、フォーム キー検証は、リクエスト内のフォーム キーが Magento によって生成されたものと一致するかどうかを確認することで CSRF 攻撃を防ぐセキュリティ機能です。開発者がカスタム フォームに form\_key を含めない場合、検証は失敗し、エラーが表示されます。したがって、開発者はテンプレート ファイルで <?=\$block->getBlockHtml('formkey')?> を使用して、リクエストに form\_key を追加する必要があります。検証済みの参照: <https://magento.stackexchange.com/questions/95171/magento-2-form-validation>

**最新問題: 22**

Adobe Commerce は顧客のパスワードをハッシュするためにどのハッシュ アルゴリズムを選択しますか？

A. Sodium 拡張機能がインストールされている場合は SHA256 が選択され、インストールされていない場合は Magento のデフォルトのハッシュ アルゴリズムとして MD5 が使用されます。

B. Sodium 拡張機能がインストールされている場合は、Argon 2ID13 が選択され、そうでない場合は Magento のデフォルトのハッシュ アルゴリズムとして SHA256 が使用されます。

C. Sodium 拡張機能がインストールされているかどうかは関係ありません。Magento のハッシュのデフォルト アルゴリズムは SHA256 になります。

**Answer: (解答を表示する)**

Sodium 拡張機能がインストールされている場合、Magento のデフォルトのハッシュ アルゴリズムとして Argon 2ID13 が選択されます。

それ以外の場合は、SHA256 が使用されます。

Sodium 拡張機能は、暗号化機能を提供する PHP 拡張機能です。Argon 2ID13 は、SHA256 よりも安全であると考えられているパスワード ハッシュ アルゴリズムです。

Sodium 拡張機能がインストールされている場合、Magento は顧客パスワードのデフォルトのハッシュ アルゴリズムとして Argon 2ID13 を使用します。Sodium 拡張機能がインストールされていない場合、Magento は SHA256 をデフォルトのハッシュ アルゴリズムとして使用します。

### 最新問題: 23

Adobe Commerce Cloud プラットフォームでは、ECE-Tools パッケージはどの順序でパッチを適用しますか？

- A.** 1. Cloud Patches for Commerce パッケージに含まれるすべての必要な Magento パッチ。  
2. /m2-hotfixes ディレクトリ内のカスタム パッチは、パッチ名のアルファベット順になっています。  
3. 品質パッチ ツールに含まれる、選択されたオプションの Magento パッチ。
- B.** 1. Cloud Patches for Commerce パッケージには、必要なすべての Magento パッチが含まれています。  
2. 品質パッチ ツールに含まれる、選択されたオプションの Magento パッチ。  
3. /m2-hotfixes ディレクトリ内のカスタム パッチは、パッチ名のアルファベット順になっています。
- C.** 1. /m2-hotfixes ディレクトリ内のカスタム パッチ (パッチ名のアルファベット順)。  
2. 必要なすべての Magento パッチは、Cloud Patches for Commerce パッケージに含まれています。  
3. 品質パッチ ツールに含まれる、選択されたオプションの Magento パッチ。

**Answer: B (メッセージを残す)**

ECE-Tools パッケージがパッチを適用する順序は次のとおりです。

必要なすべての Magento パッチは、Cloud Patches for Commerce パッケージに含まれています。

品質パッチ ツールに含まれる、選択されたオプションの Magento パッチ。

/m2-hotfixes ディレクトリ内のカスタム パッチは、パッチ名のアルファベット順に並べられています。

ECE-Tools パッケージは、Adobe Commerce Cloud プロジェクトを管理および展開するために設計されたスクリプトとツールのセットです。Cloud Patches for Commerce パッケージは、Adobe Commerce Cloud の機能に影響する Magento コアの問題に対する必要なパッチのセットを提供する ECE-Tools の依存関係です。Quality Patches Tool は、開発者が完全な製品リリースを待たずに特定の Magento の問題に対して個別のパッチを適用できるようにするオプションのツールです。/m2-hotfixes ディレクトリは、開発者が Adobe Commerce Cloud プロジェクト用の独自のカスタム パッチを配置できるディレクトリです。検証済みの参照: [Magento 2.4 DevDocs]

### 最新問題: 24

Adobe Commerce 開発者は、管理パネルに WYSIWYG として表示される製品 EAV 属性をプログラムで作成したいと考えています。wysiwyg\_enabled が true に設定されていることを確認しましたが、属性は管理パネルに WYSIWYG として表示されません。

考えられる理由は何でしょうか？

- A.** is\_html\_allowed\_on\_front オプションが false に設定されています。
- B.** 入力タイプがテキストに設定されていません。
- C.** 入力タイプがテキストエリアに設定されていません。

**Answer: C (メッセージを残す)**

製品 EAV 属性の input\_type 属性は、管理パネルで属性の値を入力するために使用される入力フィールドのタイプを指定します。textarea 入力タイプは、WYSIWYG フィールドに使用されません。input\_type 属性が textarea に設定されていない場合、属性は管理パネルで WYSIWYG として表示されません。

これを修正するには、開発者は input\_type 属性を textarea に設定する必要があります。

### 最新問題: 25

販売者は、Adobe Commerce Cloud Pro プランの統合環境でパフォーマンスの問題が発生しており、拡張統合環境にアップグレードしたいと考えています。

拡張統合環境にアップグレードするために再デプロイする前に必要な手順は何ですか？

- A.** 1. 統合ブランチの数を2つに制限する  
2. アップグレードを要求するサポートチケットを送信する

- B. 1. 統合ブランチの数を3つに制限する
- 2. .magento.env.yamlのENV.ENVIRONMENTをENHANCEDINTEGRATIONに設定する
- C. 1. 統合ブランチの数を4つに制限する
- 2. クラウドGUIで統合環境を設定し、拡張スイッチをオンに設定します。

Answer: A (メッセージを残す)

拡張統合環境にアップグレードするために再展開する前に必要な手順は、統合ブランチの数を2つに制限し、アップグレードを要求するサポート チケットを送信することです。

拡張統合環境は、統合環境の改良版であり、パフォーマンス、安定性、セキュリティが向上しています。一度にアクティブにできるブランチは4つまでですが、標準の統合環境から移行できるブランチは2つだけです。開発者は、Adobe サポートにアップグレードを依頼する前に、余分なブランチを削除するか、非アクティブにする必要があります。検証済みの参照: [Magento 2.4 DevDocs]

最新問題: 26

モジュールの di.xml ファイルは、クラス Action の2つのプラグインを添付します。

PluginA には、beforeDispatch、aroundDispatch、afterDispatch というメソッドがあります。PluginB には、次のメソッドがあります。

beforeDispatch、afterDispatch。

```
<config>
  <type name="Magento\Framework\App\Action\Action">
    <plugin name="vendor_module_plugina" type="Vendor\Module\Plugin\PluginA" sortOrder="10" />
    <plugin name="vendor_module_pluginb" type="Vendor\Module\Plugin\PluginB" sortOrder="20" />
  </type>
</config>
```

around プラグインのコードは次のとおりです。

```
class PluginA
{
  public function aroundDispatch(\Magento\Framework\App\Action\Action $subject, $next, $request)
  {
    // custom code
    return $request;
  }
}
```

プラグインの実行順序はどうなりますか？

- A.
  - PluginA::beforeDispatch()
  - PluginA::aroundDispatch()
  - PluginB::beforeDispatch()
  - Action::dispatch()
  - PluginB::afterDispatch()
  - PluginA::aroundDispatch()

- B.
  - PluginB::beforeDispatch()
  - PluginA::beforeDispatch()
  - PluginA::aroundDispatch()
  - PluginA::afterDispatch()
  - PluginA::aroundDispatch()
  - PluginA::beforeDispatch()

```
Action::dispatch()  
PluginB: afterDispatch()  
PluginA::aroundDispatch()  
PluginA::afterDispatch()
```

C.

**Answer: A (メッセージを残す)**

\* Magento プラグインの種類と実行順序:

\* Before Plugins: 実際のメソッドが呼び出される前に実行します。昇順の sortOrder で実行されます。

\* Around プラグイン: メソッド呼び出しをラップします。around メソッドが実行され、実際のメソッドを呼び出す \$next コールバックに制御が渡されます。

\* After プラグイン: メソッドが完了した後に実行します。降順の sortOrder で実行されます。

\* プラグイン構成の分析:

\* PluginA (sortOrder="10") には beforeDispatch、aroundDispatch、afterDispatch メソッドがあります。

\* PluginB (sortOrder="20") には beforeDispatch メソッドと afterDispatch メソッドがあります。

\* オプションAの実行順序の内訳:

\* プラグイン前:

\* PluginA::beforeDispatch() が最初に実行されます (sortOrder が低い)。

\* PluginB::beforeDispatch() が 2 番目に実行されます。

\* プラグイン周辺:

\* PluginA::aroundDispatch() は、ディスパッチ メソッドをラップします。カスタム コードが完了し、\$next 関数が呼び出された後にのみ、実際のディスパッチ呼び出しに進みます。

\* アクションディスパッチ:

\* Action::dispatch() は、\$next() を介して PluginA::aroundDispatch() の一部として実行されます。

\* プラグイン後:

\* PluginB::afterDispatch() は、sortOrder が高いため、dispatch メソッドの後に実行されます。

\* PluginA::afterDispatch() が最後に実行されます。

オプション A の実行フロー:

\* プラグインA::beforeDispatch()

\* プラグインB::beforeDispatch()

\* PluginA::aroundDispatch() は Action::dispatch() をラップします

\* Action::dispatch() は PluginA の aroundDispatch 内で発生します

\* プラグインB::afterDispatch()

\* プラグインA::afterDispatch()

これはオプション A で指定された順序と一致します。

参考文献:

\* Magento プラグイン (インターセプター) の概要 - Adobe Commerce 開発者ガイドでは、before、around、after プラグインの役割と順序について詳しく説明しています。

\* プラグインの実行順序の管理 - sortOrder がプラグインの実行順序にどのように影響するかについて説明します。

\* Magento 依存性注入構成 - di 内でプラグインを構成するための詳細情報。

xml です。

sortOrder とプラグイン タイプのルールに従うことで、オプション A は指定されたセットアップのプラグイン実行順序を正しく表します。

最新問題: 27

Adobe Commerce 開発者は、既存のルートにアクションを追加したいと考えていますが、元のルートのアクションの機能には干渉したくありません。

開発者は、アクションが元のモジュールで副作用なく動作することを保証するために何をする必要がありますか？

- A. ルート宣言では、before または after パラメータを使用して、元のモジュールの前または後にモジュールをロードします。
- B. 新しいアクションを標準ルーターコンストラクターの \$actionist パラメーターに挿入します。
- C. My.Module の controllers/front\_name/ にアクションを追加すると、Magento が自動的にそれを検出して使用します。

**Answer: A** ([メッセージを残す](#))

元のルートに干渉せずに既存のルートにアクションを追加するには、開発者はルート宣言で before または after パラメータを使用する必要があります。これにより、開発者のモジュールがそれぞれ元のモジュールの前または後に読み込まれます。

たとえば、次のコードは、元のモジュールのアクションの前に、my\_module/index ルートにアクションを追加します。

```
<ルート id="my_module/index">
<before>my_module_before</before>
<アクションクラス="MyModule\Controller\Index">
<引数>
<引数名="コンテキスト" タイプ="Magento\Framework\App\Action\Context"/>
</引数>
</アクション>
</ルート>
```

my\_module\_before アクションは MyModule\Controller\Index アクションの前に実行されるため、開発者は元のアクションが実行される前に必要なセットアップを実行できます。

#### 最新問題: 28

Adobe Commerce の開発者は、PageBuilder のスライダー コンテンツ タイプを変更して、新しいカスタム コンテンツ タイプ (スライド以外) を子として割り当てられるようにするように依頼されました。開発者は、モジュール内に improved\_slide という新しいコンテンツ タイプを既に作成しています。次に、新しい view/adminhtml を作成する必要があります。

新しいコンテンツ タイプをスライダー コンテンツ タイプの子にできるようにするには、モジュール内の /pagebuilder/content\_type/slider.xml ファイルを編集します。

これを実現するための正しい XML は何ですか？

- A. 

```
<type name="slider">
  <children>
    <child name="improved_slide" policy="allow"/>
  </children>
</type>
```
- B. 

```
<type name="slider">
  <allowed_descendants>
    <descendant name="improved_slide" />
  </allowed_descendants>
</type>
```
- C. 

```
<type name="slider">
  <arguments>
    <argument name="allowed_children" xsi:type="array">
      <item name="improved_slide" xsi:type="string">improved_slide</item>
    </argument>
  </arguments>
</type>
```

**Answer: (解答を表示する)**

正解はオプション C です。この XML 構成は、Magento の PageBuilder のスライダー コンテンツ タイプに許可される子コンテンツ タイプを定義する正しい方法です。

Magento PageBuilder コンテンツ タイプ構造:

PageBuilder では、各コンテンツ タイプで、子として許可される他のコンテンツ タイプを指定できます。これは、コンテンツ タイプの XML 構成内で `allowed_children` 配列を定義することによって行われます。

オプション C の分析:

引数の定義: オプション C は、`<arguments>` ノードを使用して、`allowed_children` という名前の新しい引数を定義します。

配列構造: この引数は、`xsi:type="string"` で許可された子として `improved_slide` を指定する項目を含む配列 (`xsi:type="array"`) です。

この構成は、スライダー コンテンツ タイプに許可される子コンテンツ タイプを明示的に定義し、PageBuilder で許可される子要素の Magento の構造に準拠しているため、正しい構成です。

オプション A と B が間違っている理由:

オプション A: `policy="allow"` を指定した `<children>` ノードを使用しますが、これは PageBuilder コンテンツ タイプで許可される子を定義する標準的な方法ではありません。この形式は正しくなく、PageBuilder では認識されません。

オプション B: `<allowed_descendants>` を使用しますが、これも Magento の PageBuilder が子コンテンツ タイプを宣言する方法と一致しません。正しい用語は `allowed_descendants` ではなく `allowed_children` です。

参考文献:

Magento PageBuilder 開発ガイド - このガイドでは、PageBuilder のカスタマイズとコンテンツ タイプの管理について詳しく説明します。

PageBuilder でのコンテンツ タイプの構成 - カスタム コンテンツ タイプに許可される子を定義する方法に関するドキュメント。

Adobe Commerce PageBuilder コンテンツ タイプ XML リファレンス - PageBuilder コンテンツ タイプの正しい XML 構造の詳細。

オプション C の構成は、どのコンテンツ タイプを別のコンテンツ タイプ内にネストできるかを定義する Adobe Commerce PageBuilder の構造と一致しているため、正しい選択となります。

#### 最新問題: 29

Adobe Commerce Cloud プロジェクトでは、2 つの新しい支払いモジュールをインストールした拡張統合環境を使用しています。

開発者は Cloud CLI for Commerce ツールを使用しています。

開発者は統合環境でこの新しい機能をテストするために何をやるのでしょうか?

- A. 1. 統合環境ブランチの 1 つを複製します。
- 2. 統合から新しいアクティブ ブランチを作成し、モジュールをインストールします。
- 3. 変更をプッシュします。
- B. 1. 統合から新しいブランチを作成し、モジュールをインストールします。
- 2. 変更をプッシュします。
- 3. ブランチのアクティブステータスのチェックは必要ありません。
- C. 1. アクティブな統合環境ブランチの 1 つを非アクティブ化します。
- 2. 統合から新しいアクティブ ブランチを作成し、モジュールをインストールします。
- 3. 変更をプッシュします。

**Answer: C** ([メッセージを残す](#))

開発者は、アクティブな統合環境ブランチの 1 つを非アクティブ化し、統合から新しいアクティブなブランチを作成してモジュールをインストールし、変更をプッシュすることで、統合環境で新しい機能をテストできます。これは、拡張統合環境では一度にアクティブなブランチが 4 つに制限されており、各ブランチには専用のデータベースとサービスがあるためです。開発者は、Cloud CLI for Commerce ツールを使用してブランチを管理し、コードの変更をデプロイできます。検証済みの参照:

[Magento 2.4 開発ドキュメント] 1

#### 最新問題: 30

Adobe Commerce 開発者は、Adobe Commerce Cloud (Pro Plan) の製品に重要な機能を導入しようとしています。機能に問題が発生した場合にロールバックできるように、スナップショットを作成したいと考えています。

スナップショットはどのように作成されるのでしょうか？

- A. Project Web Interface の専用ボタンを使用します。
- B. Cloud CLI for Commerce 専用コマンドを使用します。
- C. Adobe Commerce Cloud サポートへのチケットを作成します。

**Answer:** [\(解答を表示する\)](#)

開発者は、プロジェクト Web インターフェイスの専用ボタンを使用して、Adobe Commerce Cloud (Pro Plan) の製品に重要な機能をデプロイする前にスナップショットを作成できます。スナップショットは、コード、データ、メディア ファイル、構成設定を含む環境全体のバックアップです。スナップショットを使用すると、デプロイ中またはテスト中に問題やエラーが発生した場合に、環境を以前の状態に復元できます。開発者は、プロジェクト Web インターフェイスにアクセスし、環境を選択して [スナップショットの作成] をクリックすることで、スナップショットを作成できます。検証済みの参照: [Magento 2.4 DevDocs]

### 最新問題: 31

Adobe Commerce 開発者が、カスタム ブログ機能の投稿リストを検索して取得するための新しい API メソッドを追加しました。これは、モジュールの etc/webapi.xml ファイルの内容です。

```
<?xml version="1.0" ?>
<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Webapi:etc/webapi.xsd">
  <route url="/V1/myvendor-blog/post/search" method="GET">
    <service class="MyVendor\Blog\Api\PostRepositoryInterface" method="getList"/>
    <resources>
      <resource ref="MyVendor_Blog::Post_view"/>
    </resources>
  </route>
</routes>
```

新しいコードは本番環境に導入されており、販売者は https://merchant.domain.com を使用しています。

/swagger を使用して新しいエンドポイントを確認しようとしたが、swagger には表示されません。

その理由は何でしょうか？

- A. webapi.xml ファイルを etc/webapi\_rest/webapi.xml ファイルに移動する必要があります。
- B. 新しいエンドポイントは匿名ではないため、販売者は新しいメソッドを確認するために、Swagger に有効な統合トークンを入力する必要があります。
- C. MyVendor\Blog\Api\PostRepositoryInterface クラスに greturn アノテーションがありません。

**Answer: C** ([メッセージを残す](#))

Magento では、カスタム API エンドポイントを Swagger で文書化するには、インターフェース メソッドに @return 型を含む適切な PHPDoc アノテーションが必要です。これらのアノテーションは、戻り値の型と説明を Magento の Web API システムと Swagger に通知し、エンドポイントを正しく表示できるようにします。

\* @return アノテーションの役割:

\* @return アノテーションは、API メソッドの戻り値の型を定義するために必要です。このアノテーションがないと、Magento の Swagger ドキュメント システムが出力タイプを認識できず、エンドポイントが Swagger に表示されない可能性があります。

\* オプション C が正しい理由:

\* @return アノテーションが存在しないことが、エンドポイントが Swagger に表示されない一般的な理由です。これは、API ドキュメントの戻り値の型を明確に指定する必要があるカスタム インターフェイスを使用する場合に特に当てはまります。

\* オプション A は、webapi.xml が REST API 用に正しく配置されているため、正しくありません。オプション B は、エンドポイントの表示にトークンは必要なく、実行にのみ影響するため、関係ありません。

\* 実装に関するアドバイス:

```
* MyVendor\Blog\Api\PostRepositoryInterfaceのすべてのメソッドに正確な
@return アノテーション。例:
/**
 * 投稿リストを取得
 * * @return \MyVendor\Blog\Api\Data\PostSearchResultsInterface
 */
* 参考文献:
* API および Swagger ドキュメントに関する Adobe Commerce DevDocs
* Magento API インターフェースの PHPDoc 標準
```

有効な **AD0-E716** 問題集は GoShiken.com が提供された合格しやすい AD0-E716 試験問題集！ GoShiken.com が最新の **AD0-E716** 試験問題集を提供しています。GoShiken.com AD0-E716 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-E716 問題集をゲットする人はこちら: <https://www.goshiken.com/Adobe/AD0-E716-mondaishu.html> (**7230%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

開発者は、Adobe Commerce Cloud Staging 環境で送信メールをオンにするにはどうすればよいですか？

**A.** コマンドラインから

ece-tools enable\_smtp を true にする

**B.** コマンドラインから

magento-cloud environment:info -p <プロジェクトID> -e <環境ID> enable\_smtp true

**C.** プロジェクト Web インターフェースにアクセスし、ステージング環境を選択します。

環境の構成を選択します。

**Answer: C** ([メッセージを残す](#))

送信メールをオンにする

Explanation:

開発者は、プロジェクト Web インターフェースにアクセスしてステージング環境を選択することで、Adobe Commerce Cloud ステージング環境で送信メールをオンにすることができます。次に、開発者は環境の設定を選択し、送信メールをオンに切り替えます。これにより、ステージング環境の SMTP サービスが有効になり、アプリケーションからメールを送信できるようになります。検証済みリ

ファレンス: [Magento 2.4 DevDocs] 1

#### 最新問題: 33

Adobe Commerce Cloud の PHP エラー ログにアクセスする 2 つの方法は何ですか? (2 つ選択してください。)

**A.** Cloud CLI for Commerce の専用コマンドを使用します。

**B.** プロジェクト Web インターフェースの専用エントリに移動します。

**C.** SSH 経由でサーバーに接続し、ログ ファイルをローカライズします。

**D.** Adobe Admin Log アプリケーションを使用します。

**Answer: (解答を表示する)**

Adobe Commerce Cloud の PHP エラー ログにアクセスするには、Cloud CLI for Commerce の専用コマンドを使用する方法と、SSH 経由でサーバーに接続してログ ファイルをローカライズする方法の 2 つがあります。Cloud CLI for Commerce は、開発者が Adobe Commerce Cloud プロジェクトおよび環境と対話できるようにするコマンドライン ツールです。開発者は、コマンド `magento-cloud log php` を使用して、任意の環境から PHP エラー ログを表示またはダウンロードできます。

Adobe Commerce Cloud では、PHP エラー ログにアクセスするには、Cloud CLI を使用するか、SSH 経由でサーバーに直接接続します。

\* ログにアクセスするための Cloud CLI:

\* Cloud CLI には、PHP エラー ログなどのさまざまなログにアクセスするためのコマンドが用意されており、サーバーに直接接続しなくても簡単にアクセスできます。

\* 直接ログアクセス用の SSH:

\* より実践的な制御や特定のログ検査を行うには、SSH を使用してサーバーに接続し、Adobe Commerce Cloud 環境の `/var/log` ディレクトリに通常保存されているログ ファイルに直接アクセスできます。

\* オプション A と C が正しい理由:

\* これら 2 つの方法は、Adobe Commerce Cloud のエラー ログに直接アクセスするための公式な方法を提供します。

\* オプション B は、プロジェクト Web インターフェイスに専用の PHP エラー ログ エントリがないため、正しくありません。

オプション D は、Adobe Admin Log アプリケーションがないため、不正解です。

\* 参考文献:

\* ログへのアクセスに関する Adobe Commerce Cloud ドキュメント

#### 最新問題: 34

Adobe Commerce 開発者が新しい配送業者を作成しました。すべてが実装され、`collectRates()` および `getAllowedMethodsQ` 関数は以下で確認できます。

```
public function collectRates(RateRequest $request) {
    if (!$this->getConfigFlag('active')) {
        return false;
    }

    $result = $this->rateResultFactory->create();
    $method = $this->rateMethodFactory->create();

    $method->setCarrier($this->code);
    $method->setCarrierTitle($this->getConfigData('title'));
    $method->setMethod($this->code);
    $method->setMethodTitle($this->getConfigData('name'));

    $method->setPrice(0);
    $method->setCost(10);

    $result->append($method);

    return $result;
}
```

```
public function getAllowedMethods() {
    return [$this->_code => $this->getConfigData('name')];
}
```

上記のコードの場合、配送方法の表示コストと顧客に請求される最終金額はいくらになりますか？

- A. 配送方法は SO と表示されますが、顧客は注文に対して 10 ドルの手数料を支払うこととなります。
- B. 配送方法には \$0 と表示され、顧客は新しい配送方法の使用に対して \$0 を支払うこととなります。

C. 配送方法には 10 ドルが表示され、顧客は新しい配送方法を使用するために 10 ドルを支払うことになります。

**Answer:** [\(解答を表示する\)](#)

配送方法には 10 ドルが表示され、顧客は新しい配送方法を使用するために 10 ドルを支払うことになります。これは、collectRates メソッドが結果オブジェクトで配送方法の価格とコストを 10 に設定するためです。価格はフロントエンドで顧客に表示されるもので、コストは注文時に顧客に請求されるものです。この場合、手数料は config.xml ファイルで 0 に設定されているため使用されません。検証済みリファレンス: [\[Magento 2.4 DevDocs\]](#) [\[Magento Stack Exchange\]](#)

最新問題: 35

Adobe Commerce 開発者は、注文グリッドに新しいエクスポート オプションを追加するタスクを担当しており、sales\_order\_grid.xml 内のエクスポート ボタンに次のコードを追加しました。

```
<exportButton>
  <settings>
    <options>
      <option name="txt" xsi:type="array">
        <item name="value" xsi:type="string">txt</item>
        <item name="label" xsi:type="string" translate="true">TXT</item>
        <item name="url" xsi:type="string">vendor_module/sales/export/customExport</item>
      </option>
    </options>
  </settings>
</exportButton>
```

テストするとリダイレクトされますが、このエラーの原因は何でしょうか？

- A. オプションの uri 属性が無効です。
- B. レイアウト キャッシュを更新する必要があります。
- C. 開発者は新しいエクスポート オプションのフォームキーを追加する必要があります。

**Answer: C** ([メッセージを残す](#))

セキュリティ上の理由からフォームキーが必要なため、開発者は新しいエクスポート オプションにフォームキーを追加する必要があります。フォームキーがないと、リクエストは拒否され、ダッシュボード ページにリダイレクトされます。検証済みの参照: [\[Magento 2.4 ユーザー ガイド\]](#) [\[Magento 2.4 DevDocs\]](#) Magento のグリッドにカスタム エクスポート オプションを追加する場合、フォームの送信を伴うアクションにフォームキーを含めることが重要です。Magento は CSRF (クロスサイト リクエスト フォージェリ) 保護にフォームキーを使用しているため、フォームキーを省略するとリダイレクトや操作の失敗につながる可能性があります。

\* フォームキーの要件:

\* Magento では、フォーム キーはリクエストが有効であることを確認するためにフォームに含まれる隠しトークンです。

これは、不正なアクションを防ぐのに役立つため、状態を変更したりデータをエクスポートしたりするアクションでは特に重要です。

\* カスタム エクスポート ボタンを追加するとフォームの送信がトリガーされ、有効なフォーム キーが必要になります。

これがないと、Magento はセキュリティ対策としてデフォルト ページまたは管理ダッシュボードにリダイレクトする可能性があります。

\* オプション C が正しい理由:

\* オプション C は、フォーム キーの不足を問題として正しく特定します。Magento は、機密性の高い操作でフォーム キーの不足を検出すると、セキュリティを維持するためにデフォルトでリダイレクトを実行します。

\* URI に関するオプション A は、リダイレクトを引き起こす可能性が低くなります。同様に、レイアウト キャッシュに関するオプション B は、ここでのリダイレクトの原因である CSRF 検証に直接影響を与えません。

\* 解決:

\* ボタンまたはフォーム送信ロジックを変更してフォームキーを含めます。通常は、form\_key= を追加します。

{{formKey}} は URL パラメータ内またはフォーム データ内にあります。

\* 参考文献:

\* フォームキーとCSRF保護に関するAdobe Commerceドキュメント

\* グリッドにボタンを追加する方法についての Magento DevDocs

#### 最新問題: 36

Adobe Commerce Cloud プラットフォームでは、CLI for Commerce コマンド `magento-cloud environment:branch <environment-name> <parent-environment-id>` を起動すると、どのようなタイプの環境がプロビジョニングされますか？

- A. コードやデータベースのない空の統合環境。
- B. 新規の Adobe Commerce Cloud インストールとの統合環境。
- C. 親環境のコードとデータベースとの統合環境。

**Answer:** [\(解答を表示する\)](#)

CLI for Commerce コマンド `magento-cloud environment:branch <environment-name> <parent-environment-id>` を起動したときにプロビジョニングされる環境のタイプは、親環境のコードとデータベースを含む統合環境です。統合環境は、Adobe Commerce Cloud プラットフォームでのテストおよび開発目的で使用される一時的な環境です。任意のコード ブランチから作成でき、専用のデータベースとサービスを持ちます。CLI for Commerce コマンドを使用して統合環境を作成すると、親環境のコードとデータベースが新しい統合環境にコピーされ、親環境の正確なレプリカが作成されます。検証済みリファレンス: [Magento 2.4 DevDocs]

#### 最新問題: 37

1 日に 2 回実行される cron サービスを使用して開発された統合があり、請求書を作成できる注文がある場合に、統合された ERP システムに注文 ID を送信します。注文には、次のコードがすでにロードされています。

```
$order = $this->orderRepository->get($orderId);
```

ストアに作成する請求書があるかどうかを確認するために、Adobe Commerce 開発者はどのような実装を使用しますか？

```
A. if ($order->canInvoice()) {  
    // send integration to the ERP  
}
```

```
B. if ($order->hasInvoice()) {  
    // send integration to the ERP  
}
```

```
C. if (!$order->isPaymentReview()) {  
    // send integration to the ERP  
}
```

**Answer: B** ([メッセージを残す](#))

開発者は、注文オブジェクトの `canInvoice` メソッドを使用して、注文が請求可能かどうかを確認できます。

このメソッドは、注文の状態が新規、処理中、または支払い確認中であり、まだ完全に請求されていない場合に `true` を返します。開発者は、このメソッドを条件文で使用して、注文が請求可能な場合にのみ注文 ID を ERP システムに送信できます。検証済みの参照: [Magento 2.4 DevDocs] [Magento Stack Exchange]

#### 最新問題: 38

開発者は、Adobe Commerce Cloud Staging 環境で CSS ファイルの拡大をどのように有効にすればよいでしょうか？

A. コマンドラインからローカルに

```
bin/magento config:set --lock-config dev/css/minify_files 1
```

`app/etc/config.php` ファイルをコミットして再デプロイします。

B. 管理パネルでストア > 設定 > 構成 > 詳細 > 開発者 > CSS 構成を更新します。

C. Adobe Commerce Staging環境にSSHで接続します。コマンドラインから

```
ece-tools config:set dev/css/minify_files 1  
bin/magento setup:static-content:deploy
```

**Answer: A** ([メッセージを残す](#))

開発者は、コマンドラインから bin/magento config:set --lock-config dev/css/minify\_files 1 コマンドをローカルで実行することで、Adobe Commerce Cloud ステージング環境で CSS ファイルの拡大を有効にすることができます。これにより、app/etc/config.php ファイルの設定値が設定され、管理パネルで変更されないようにロックされます。その後、開発者は app/etc/config.php ファイルをコミットして、環境を再デプロイする必要があります。検証済みリファレンス: [Magento 2.4 DevDocs] 2

最新問題: 39

Adobe Commerce 開発者は、API を通じて取得される請求書に値の配列を追加するための新しい拡張属性を追加します。

しばらくして、テクニカル マネージャーが作業を確認し、開発者がモジュール内に作成した extension\_attributes.xml ファイルに問題があることに気がきました。

この XML スニペットの問題は何ですか?

- A. 拡張属性は間違ったインターフェースを参照しています。Magento\saies\Api\data\invoiceinterface を参照する必要があります。
- B. 拡張属性は、実装するインターフェース (Magento\saies\Api\invoiceRepositorymterface) ではなく、リポジトリを参照します。
- C. 型が間違っています。文字列 [] は配列に置き換える必要があります。

**Answer:** ([解答を表示する](#))

Adobe Commerce で拡張属性を追加するときは、extension\_attributes.xml ファイルで正しいインターフェイスを参照することが重要です。Magento の拡張属性は、既存のエンティティにカスタム データ フィールドを追加するために使用され、リポジトリではなくコア エンティティ インターフェイスを拡張する必要があります。

この場合 :

\* 開発者は、リポジトリや不正なインターフェースではなく、請求書のデータ構造を表す Magento\Sales\Api\Data\InvoiceInterface を参照する必要があります。

\* extension\_attributes.xml では、データ インターフェイス (例: Magento\Sales\Api\Data\InvoiceInterface) を指定して、拡張属性が請求書エンティティに正しく適用され、API 経由で請求書を取得するときアクセス可能であることを確認する必要があります。

正しい XML 形式:

```
<?xml バージョン="1.0"?>  
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:  
magento:モジュール:Magento_ExtensionAttributes/etc/extension_attributes.xsd">  
<拡張機能属性>  
<属性コード="カスタム属性" タイプ="文字列[]" />  
</拡張機能属性>  
</config>
```

追加リソース:

- \* Adobe Commerce 開発者ガイド: 拡張属性の使用
- \* 拡張属性 XML リファレンス

最新問題: 40

Adobe Commerce Cloud に標準搭載されている 2 つの機能は何ですか? (2 つ選択してください。)

- A. ACL をサポートする
- B. プラットフォームで提供される継続的なデプロイメント
- C. すべての主要なブログプラットフォームとの組み込みコネクタ
- D. 速く

**Answer:** ([解答を表示する](#))

Adobe Commerce Cloud に標準装備されている 2 つの機能は、Support ACL と Fastly です。Support ACL は、開発者が Adobe Commerce Cloud プラットフォーム上のさまざまなユーザーとロールのアクセス制御リストを管理できるようにする機能です。開発者は、プロジェクト レベルと環境レベルでさまざまなアクションとリソースの権限を作成し、割り当てることができます。Fastly は、Adobe Commerce Cloud プロジェクトのサイト パフォーマンスとセキュリティを向上させるクラウドベースのキャッシュ サービスです。Fastly は、CDN、画像最適化、WAF、DDoS 保護などの機能を提供します。検証済みの参照: [Magento 2.4 DevDocs] 3

**最新問題: 41**

Adobe Commerce 開発者は、Marketplace で販売するモジュール (Vendor.ModuleName) を作成しています。

新しいモジュールは宣言型スキーマを使用してデータベース テーブルを作成するため、開発者はモジュールが無効になったときにテーブルが削除されるようにする必要があります。これを実現するために開発者は何をする必要がありますか？

**A.** 開発者が行う必要がある作業はこれ以上ありません。モジュールが無効になり、bin/magento setup:upgrade が実行されると、テーブルは削除されます。

**B.** 開発者が行う必要がある作業はこれ以上ありません。テーブルは、bin が実行されるときに削除されます。

/magento module:uninstall vendor\_ModuleName が実行されます。

**C.** Magento\Framework\setup\Patch\PatchRevertable インターフェースを実装し、revert関数でテーブルを削除するスキーマ パッチを追加します。

**Answer:** ([解答を表示する](#))

Adobe Commerce (Magento) では、bin/magento module:uninstall コマンドを使用してモジュールをアンインストールすると、Magento はそのモジュールによって作成されたデータベース テーブルのクリーンアップを自動的に処理します。これは、モジュールのライフサイクル イベントを管理するための Magento の組み込みメカニズムの一部です。

\* 宣言的なスキーマとテーブルの削除:

\* 宣言型スキーマを使用すると、モジュールがインストールまたはアンインストールされると、Magento は自動的にテーブルを作成および削除します。

\* コマンド bin/magento module:uninstall を使用すると、Magento は宣言型スキーマで定義されているカスタム テーブルを含む、モジュールに関連付けられたすべてのデータベース スキーマを削除します。

\* 追加の手順は必要ありません:

\* オプション B は module:uninstall コマンドを活用するため、追加のコードや構成なしでテーブル削除プロセスを自動的に処理します。

\* オプション A は正しくありません。setup:upgrade ではテーブル自体が削除されず、スキーマのみが更新されます。

\* オプション C は、変更を明示的に元に戻す有効なアプローチですが、Magento に組み込まれているアンインストール機能のため、ここでは必要ありません。

\* 参考文献:

\* 宣言型スキーマに関する Magento DevDocs

\* モジュールのアンインストールに関する Adobe Commerce ドキュメント

**最新問題: 42**

Adobe Commerce 開発者がカスタム ギャラリー拡張機能の開発に取り組んでいます。

モジュールは、画像のアップロードに Magento\catalog\Model\ImageUploader クラスを使用します。カスタム画像のアップロード用の管理コントローラーは、Vendor\CustomGallery\Controller\Adminhtml\Image\Upload です。

画像は、デフォルトのものとは異なる basePath および baseTmpPath に保存する必要があります。

すでに使用している他のモジュールに影響を与えずに、デフォルトの imageuploader クラスを拡張して使用するにはどうすればよいですか？

1. Create a Virtual Type and configure the `basePath` and `baseTmpPath`.
2. Inject the virtual type `Vendor\CustomGallery\GalleryImageUpload` into admin controller:

```

<virtualType
  name="Vendor\CustomGallery\GalleryImageUpload"
  type="Magento\Catalog\Model\ImageUploader"
>
  <arguments>
    <argument name="baseTmpPath" xsi:type="string">customgallery/tmp/images</argument>
    <argument name="basePath" xsi:type="string">customgallery/images</argument>
  </arguments>
</virtualType>

<type name="Vendor\CustomGallery\Controller\Adminhtml\Image\Upload">
  <arguments>
    <argument name="imageUploader" xsi:type="object">
      Vendor\CustomGallery\GalleryImageUpload
    </argument>
  </arguments>
</type>

```

A.

1. Configure the `basePath` and `baseTmpPath` Of `Magento\Catalog\Model\ImageUploader`.
2. Inject the type `Magento\Catalog\Model\ImageUploader` into admin controller:

```

<type name="Magento\Catalog\Model\ImageUploader">
  <arguments>
    <argument name="baseTmpPath" xsi:type="string">customgallery/tmp/images</argument>
    <argument name="basePath" xsi:type="string">customgallery/images</argument>
  </arguments>
</type>

<type name="Vendor\CustomGallery\Controller\Adminhtml\Image\Upload">
  <arguments>
    <argument name="imageUploader" xsi:type="object">
      Magento\Catalog\Model\ImageUploader
    </argument>
  </arguments>
</type>

```

B.

1. Create a Virtual Type and configure the basePath and baseTmpPath.

2. Add virtual type Vendor\CustomGallery\GalleryImageUpload as a preference for Magento\Catalog\Model\ImageUploader:

```
<virtualType
  name="Vendor\CustomGallery\GalleryImageUpload"
  type="Magento\Catalog\Model\ImageUploader">
  <arguments>
    <argument name="baseTmpPath" xsi:type="string">customgallery/tmp/images</argument>
    <argument name="basePath" xsi:type="string">customgallery/images</argument>
  </arguments>
</virtualType>

<preference
  for="Magento\Catalog\Model\ImageUploader"
  type="Vendor\CustomGallery\GalleryImageUpload"
/>
```

C.

**Answer: A** (メッセージを残す)

Magento\Catalog\Model\ImageUploaderクラスを拡張し、このクラスを利用する他のモジュールに影響を与えずにカスタムbasePathとbaseTmpPathを設定するには、仮想型を使用するのが最善の方法です。

仮想型を使用すると、元のクラスや他の場所での使用法を変更することなく、特定のコンテキストに合わせてカスタマイズされたバージョンのクラスを作成できます。

オプション A が正解である理由は次の通りです。

\* 仮想型の作成: di.xml で仮想型 (Vendor\CustomGallery\GalleryImageUpload) を定義することで、ImageUploader クラスのカスタム バージョンを作成します。この仮想型には、カスタム ギャラリー モジュールに固有の basePath と baseTmpPath の独自の構成を設定できます。

\* 説明: 仮想型は、既存のクラスを少し変更したバージョンを特定のコンテキストで使用する必要がある場合に、Magento で特に役立ちます。この場合、仮想型を使用すると、基本の ImageUploader クラスを変更せずにカスタム パスを定義できるため、ImageUploader を使用する他のモジュールは影響を受けません。

\* 参考資料: Magento の依存性注入と仮想型に関する開発者向けドキュメントでは、クラスの動作を非侵襲的に制御しながらカスタマイズするためのこのパターンについて詳しく説明されています。

\* 仮想型を管理コントローラに挿入する: 次に、カスタム仮想型を Vendor\CustomGallery\Controller\Adminhtml\Image\Upload コントローラに挿入します。これにより、このコントローラのみがカスタムパスで変更されたバージョンを使用し、ImageUploader の他のインスタンスは既定の設定で機能するようになります。

\* 説明: 仮想タイプをコントローラに明示的に挿入することで、構成の変更を必要な機能のみに限定します。このアプローチは効率的であり、Magento 開発の重要な原則であるモジュールの独立性を維持します。

オプション B と C は次の理由により誤りです。

\* オプション B は、新しいパスを使用して Magento\Catalog\Model\ImageUploader を直接変更します。この変更は、サイト全体での ImageUploader クラスのすべての使用に影響するため、他のモジュールへの影響を回避するという要件に反します。

\* オプション C では、仮想タイプを作成し、それを設定として設定します。ただし、設定を使用すると、Magento アプリケーション全体の ImageUploader のすべてのインスタンスが置き換えられ、オプション B と同じ問題が発生します。

**最新問題: 43**

Adobe Commerce 拡張機能を備えた物流会社は、すべての顧客に、確認済みの配送料金のリストを CSV ファイルで毎月送信します。販売者は、この CSV ファイルを Adobe Commerce の管理設定の「ファイルアップロード」フィールドにアップロードします。

「ファイルアップロード」フィールドを表示し、実際の CSV インポートを処理するための 2 つの要件は何ですか? (2 つ選択してください。)

Add a custom backend model which extends \Magento\Framework\App\Config\Value and call afterSave:

```
Adobe
// etc/adminhtml/system.xml
<field id="import_fees" ...>
  <label>Import shipment fees</label>
  <backend_model>My\Module\Model\Config\Backend\ImportFees</backend_model>
  ...
</field>
```

A.

```
// \My\Module\Model\Config\Backend\ImportFees
class \My\Module\Model\Config\Backend\ImportFees extends \Magento\Framework\App\Config\Value
{
    ...
    public function afterSave()
    {
        /** @var \My\Module\Model\ImportFeed $importFees */
        $importFees = $this->importFeesFactory->create();
        $importFees->uploadAndImport($this);
        return parent::afterSave();
    }
} Adobe
```

B.

Add a new field in etc/adminhtml/system.xml in My\_Module with the file type:

```
Adobe
<field id="import_fees" translate="label" type="file" sortOrder="1000" showInDefault="1">
  <label>Import shipment fees</label>
</field>
```

C.

D.

Add a new field in etc/adminhtml/system.xml in My\_Module with a new custom type:

```
Adobe
<field id="import_fees" translate="label" type="My\Module\Block\Adminhtml\Form\Field\ImportFees" sortOrder="1000" showInDefault="1">
  <label>Import shipment fees</label>
</field>
```

Answer: A,B (メッセージを残す)

ファイルアップロードフィールドを表示し、実際の CSV インポートを処理するには、次の 2 つの要件を満たす必要があります。

\* 開発者は、CSV ファイルへのパスを指定する新しいシステム構成設定を作成する必要があります。

\* 開発者は、ファイルのアップロードとインポートのプロセスを処理する新しいコントローラ アクションを作成する必要があります。

system.xml ファイルは、システム構成設定を定義するために使用されます。次の XML スニペットは、CSV ファイルパスの新しいシステム構成設定を定義する方法を示しています。

テキスト

<設定>

<システム>

<設定>

```
<shipment_fees_csv_path>/path/to/csv/file</shipment_fees_csv_path>
</config>
</システム>
</config>
```

Controller\Adminhtml\ShipmentFees コントローラー クラスは、ファイルのアップロードとインポート プロセスを処理するために使用されます。次のコードは、ファイルのアップロードとインポートのプロセスを処理する新しいコントローラー アクションを作成する方法を示しています。

PHP の

パブリック関数 uploadAction()

```
{
$file = $this->getRequest()->getFile('shipment_fees_csv_file');
$file->isUploaded() の場合 {
$importer = 新しい ShipmentFeesImporter();
$importer->import($file);
}
$this->redirect('adminhtml/system_config/edit/section/shipment_fees') を返します。
}
```

#### 最新問題: 44

Adobe Commerce Cloud に標準搭載されている 2 つの機能は何ですか? (2 つ選択してください。)

- A. ACL をサポートする
- B. プラットフォームで提供される継続的なデプロイメント
- C. すべての主要なブログプラットフォームとの組み込みコネクタ
- D. 速く

**Answer: B,D (メッセージを残す)**

Adobe Commerce Cloud には、CDN および Web アプリケーション ファイアウォール サービス用の組み込み Fastly 統合や、クラウド インフラストラクチャを介した継続的なデプロイメント機能など、すぐに使用できる機能がいくつか用意されています。

\* 継続的デプロイメント:

\* Adobe Commerce Cloud は継続的デプロイメント ワークフローをサポートしており、Git およびクラウド CI/CD パイプラインとの統合により、コードを自動的に構築、テスト、デプロイできます。

\* Fastly 統合:

\* Fastly は Adobe Commerce Cloud の CDN およびキャッシュ レイヤーとして組み込まれており、キャッシュと Web アプリケーション ファイアウォールを通じてサイトの速度とセキュリティが向上します。

\* オプション B と D が正しい理由:

\* これらの機能は両方とも Adobe Commerce Cloud に固有のものです。オプション A (サポート ACL) は Magento 管理パネルの一部ですが、クラウド固有の機能ではありません。また、オプション C (ブログ プラットフォーム コネクタ) はすぐに使用できる状態では提供されません。

\* 参考文献:

\* Fastly CDN に関する Adobe Commerce Cloud ドキュメント

\* デプロイメントと CI/CD に関する Adobe Commerce Cloud ドキュメント

#### 最新問題: 45

Adobe Commerce Cloud プロジェクトでは、2 つの新しい支払いモジュールをインストールした拡張統合環境を使用しています。

開発者は Cloud CLI for Commerce ツールを使用しています。

開発者は統合環境でこの新しい機能をテストするために何をすべきでしょうか？

- A. 1. 統合環境ブランチの 1 つを複製します。
- 2. 統合から新しいアクティブ ブランチを作成し、モジュールをインストールします。
- 3. 変更をプッシュします。
- B. 1. 統合から新しいブランチを作成し、モジュールをインストールします。
- 2. 変更をプッシュします。
- 3. ブランチのアクティブステータスのチェックは必要ありません。
- C. 1. アクティブな統合環境ブランチの 1 つを非アクティブ化します。
- 2. 統合から新しいアクティブ ブランチを作成し、モジュールをインストールします。
- 3. 変更をプッシュします。

**Answer: C (メッセージを残す)**

開発者は、アクティブな統合環境ブランチの 1 つを非アクティブ化し、統合から新しいアクティブなブランチを作成してモジュールをインストールし、変更をプッシュすることで、統合環境で新しい機能をテストできます。これは、拡張統合環境では一度にアクティブなブランチが 4 つに制限されており、各ブランチには専用のデータベースとサービスがあるためです。開発者は、Cloud CLI for Commerce ツールを使用してブランチを管理し、コードの変更をデプロイできます。検証済みの参照:

[Magento 2.4 開発ドキュメント] 1

Adobe Commerce Cloud の拡張統合環境では、アクティブなブランチの数に制限があります。現在両方の統合ブランチがアクティブな場合は、テスト用の新しいアクティブなブランチを作成するには、1 つを非アクティブにする必要があります。

\* 新しいアクティブ統合ブランチの作成:

\* すでに 2 つのアクティブなブランチがある場合は、1 つを非アクティブにして、新しいブランチ用のスペースを解放します。

\* 次に、統合環境から新しいブランチを作成し、変更を適用し (新しいモジュールのインストールなど)、変更をプッシュします。

\* オプション C が正しい理由:

\* Adobe Commerce Cloud では、アクティブな統合ブランチの数が制限されています。テスト用に新しいアクティブなブランチを作成する前に、既存のブランチを非アクティブ化する必要があります。

\* オプション A は、アクティブ ブランチの制限に対処せずにブランチを複製することを提案しますが、オプション B は、ブランチのアクティブ ステータスを確認する必要性を無視するため、デプロイメントが成功しない可能性があります。

\* 参考文献:

\* ブランチ管理に関する Adobe Commerce Cloud ドキュメント

#### 最新問題: 46

開発者は、モジュール データベース スクリプトを古いインストール/アップグレード セットアップ ファイルからデータ パッチ形式に変換する必要があり、インストール/アップグレード スクリプトによって既に行われたデータベースの変更を適用したくありません。

現在のモジュールバージョンは 1.5.4 です。

古い形式 (インストール) ですでに適用されている変更をスキップするための推奨される解決策は何でしょうか？

/アップグレード スクリプト) ですか？

- A. Patchversion インターフェースを実装し、getversion() メソッドで 1.5.4 を返します。
- B. apply() メソッド内でモジュールのバージョンを確認し、バージョンが 1.5.4 未満の場合はコードを実行します。
- C. これは不可能です。モジュールはデータ パッチとインストール スクリプトの両方を実装することはできません。

**Answer: A (メッセージを残す)**

Magento 2 開発者向けのデータおよびスキーマパッチの開発ガイドによると、データパッチはデータ変更命令を含むクラスです。これらは、<Vendor>/<Module\_Name>/Setup/Patch で定義されます。

/Data/<Patch\_Name>.php ファイルを開き、MagentoFrameworkSetupPatchDataPatchInterface を実装します。データ パッチでは、パッチが関連付けられているモジュール バージョンを指定するために Patchversioninterface を実装することもできます。getVersion() メソッドは、モジュール バージョンを文字列として返します。古い形式 (インストール/アップグレード スクリプト) で既に適用されている変更をスキップするには、開発者は Patchversioninterface を実装し、getVersion() メソッドで 1.5.4 を返す必要があります。このようにすると、モジュール バージョンが 1.5.4 以上の場合にのみデータ パッチが適用されます。検証済みの参照: <https://devdocs.magento.com/guides/v2.3/extension-dev-guide/declarative-schema/data-patches.html>

有効な **AD0-E716** 問題集は GoShiken.com が提供された合格しやすい AD0-E716 試験問題集！ GoShiken.com が最新の **AD0-E716** 試験問題集を提供しています。GoShiken.com AD0-E716 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-E716 問題集をゲットする人はこちら: <https://www.goshiken.com/Adobe/AD0-E716-mondaishu.html> (**7230%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: 47

ECE-Tools は、Adobe Commerce Cloud 環境の管理と保守に使用できるツール セットを提供します。ECE-Tools が提供する機能にはどのようなものがありますか？

- A. アプリケーションをビルドし、カスタム パッチを適用し、静的コンテンツの展開用の構成をダンプします。
- B. Fastly 構成、カスタム パッチの適用、静的コンテンツの展開用のダンプ構成。
- C. アプリケーションをビルドし、カスタム パッチを適用し、S3 バックアップ tar.gz ファイルのリストを表示します。

**Answer: A** ([メッセージを残す](#))

ECE-Tools が提供する機能には、アプリケーションの構築、カスタム パッチの適用、静的コンテンツ展開用の構成のダンプなどがあります。ECE-Tools は、Adobe Commerce Cloud プロジェクトの管理と展開用に設計されたスクリプトとツールのセットです。アプリケーション コードの構築、Magento コアの問題またはカスタム モジュールのパッチの適用、静的コンテンツ展開の最適化用の構成設定のダンプを行うコマンドを提供します。検証済みリファレンス: [Magento 2.4 DevDocs] 2

#### 最新問題: 48

Adobe Commerce Cloud プラットフォームで ECE-Tools をアップグレードするために使用するコマンドは何ですか？

- A. php ./vendor/bin/ece-tools のアップグレード
- B. composer update magento/ece-tools --with-all-dependencies
- C. magento-cloud ece-tools:アップグレード

**Answer: B** ([メッセージを残す](#))

Adobe Commerce Cloud で ece-tools をアップグレードする場合、すべての依存関係が解決され、それに応じて更新されるように、Composer を使用するコマンドが推奨されます。composer update magento/ece-tools --with-all-dependencies コマンドは、ece-tools とその他の必要な依存関係を更新します。

\* 依存関係管理にComposerを使用する:

\* Composer は PHP の標準パッケージ マネージャーであり、Adobe Commerce Cloud の ece-tools などの依存関係を管理するために使用されます。このコマンドにより、必要な依存関係の更新も確実に適用されます。

\* オプション B が正しい理由:

\* オプション B は、バージョンの競合を回避するために不可欠な、依存関係を処理する Composer の機能を活用します。

\* オプション A (php ./vendor/bin/ece-tools アップグレード) は ece-tools には存在しません。オプション C は、magento-cloud ece-tools:upgrade が ece-tools を更新するための有効なコマンドではないため、正しくありません。

\* 参考文献:

\* ECEツールのアップグレードに関するAdobe Commerce Cloudドキュメント

#### 最新問題: 49

Adobe Commerce 開発者が統合テストを作成しています。参考までに Magento コア モジュールの統合テストをいくつか確認したところ、テスト クラスに注釈を追加して初期化されたデータ フィクスチャが使用されていることに気がきました。例:

```
/**
 * @magentoDataFixture Magento/AdminNotification/_files/notifications.php
 */
```

開発者は、作成した MyVendor\_MyModule をテストするために独自のフィクスチャを追加したいと考えています。これを可能にするには、どの手順を実行しますか?

- A. 1. [module dir]/Test/integration/\_files/my\_fixture.php 内の独自のモジュール内にフィクスチャ データを含む PHP ファイルを作成します。
2. テストメソッドに次のアノテーションを追加します。

```
/**
 * @magentoDataFixture MyVendor_MyModule::Test/Integration/_files/my_fixture.php
 */
```

- B. 1. [magento ルート ディレクトリ]/dev/tests/integration/testsuite/MyVendor/MyModule/\_files/my\_fixture.php にフィクスチャ データを含む PHP ファイルを作成します。
2. テストメソッドに次のアノテーションを追加します。

```
/**
 * @magentoDataFixture MyVendor_MyModule::_files/my_fixture.php
 */
```

- C. 1. [module dir]/Test/integration/\_files/my\_fixture.php 内の独自のモジュール内にフィクスチャ データを含む PHP ファイルを作成します。
2. テストメソッドに次のアノテーションを追加します。

```
/**
 * @magentoDataFixture MyVendor/MyModule/_files/my_fixture.php
 */
```

**Answer:** [\(解答を表示する\)](#)

MyVendor\_MyModule をテストするためのカスタム フィクスチャを追加するには、開発者は次の操作を行う必要があります。

[magento ルート ディレクトリ]/dev/tests/integration/testsuite/MyVendor/MyModule/\_files/my\_fixture.php にフィクスチャ データを含む PHP ファイルを作成します。

テスト メソッドに次のアノテーションを追加します。

```
@magentoデータフィクスチャ(
'testsuite/MyVendor/MyModule/_files/my_fixture.php'
)
```

これにより、テスト メソッドが実行される前に、Magento は my\_fixture.php ファイルからフィクスチャ データを読み込むようになります。

#### 最新問題: 50

Adobe Commerce Cloud プラットフォームでは、ECE-Tools パッケージはどの順序でパッチを適用しますか?

- A. 1. Cloud Patches for Commerce パッケージに含まれるすべての必要な Magento パッチ。
2. /m2-hotfixes ディレクトリ内のカスタム パッチは、パッチ名のアルファベット順になっています。
3. 品質パッチ ツールに含まれる、選択されたオプションの Magento パッチ。
- B. 1. Cloud Patches for Commerce パッケージには、必要なすべての Magento パッチが含まれています。
2. 品質パッチ ツールに含まれる、選択されたオプションの Magento パッチ。
3. /m2-hotfixes ディレクトリ内のカスタム パッチは、パッチ名のアルファベット順になっています。
- C. 1. /m2-hotfixes ディレクトリ内のカスタム パッチ (パッチ名のアルファベット順)。
2. 必要なすべての Magento パッチは、Cloud Patches for Commerce パッケージに含まれています。
3. 品質パッチ ツールに含まれる、選択されたオプションの Magento パッチ。

**Answer: B** ([メッセージを残す](#))

ECE-Tools パッケージがパッチを適用する順序は次のとおりです。

- \* 必要なすべての Magento パッチは、Cloud Patches for Commerce パッケージに含まれています。
- \* 品質パッチ ツールに含まれる、選択されたオプションの Magento パッチ。
- \* /m2-hotfixes ディレクトリ内のカスタム パッチは、パッチ名のアルファベット順になっています。

ECE-Tools パッケージは、Adobe Commerce Cloud プロジェクトを管理および展開するために設計されたスクリプトとツールのセットです。Cloud Patches for Commerce パッケージは、Adobe Commerce Cloud の機能に影響する Magento コアの問題に対する必要なパッチのセットを提供する ECE-Tools の依存関係です。Quality Patches Tool は、開発者が完全な製品リリースを待たずに特定の Magento の問題に対して個別のパッチを適用できるようにするオプションのツールです。/m2-hotfixes ディレクトリは、開発者が Adobe Commerce Cloud プロジェクト用の独自のカスタム パッチを配置できるディレクトリです。検証済みの参照: [Magento 2.4 DevDocs]

#### 最新問題: 51

Adobe Commerce Cloud プラットフォームでは、変数 env: composer\_auth はプロジェクト Web インターフェイスのどのレベルにありますか?

- A. 環境固有の変数内。
- B. 統合変数内。
- C. プロジェクト変数内。

**Answer: C (メッセージを残す)**

変数 env: composer\_auth は、プロジェクト Web インターフェイスのプロジェクト変数セクションにあります。この変数は、アクセス キーまたはトークンを必要とする Composer リポジトリの認証資格情報を保存するために使用されます。開発者は、この変数をプロジェクト レベルで設定してすべての環境に適用するか、必要に応じて環境レベルで上書きすることができます。検証済みリファレンス: [Magento 2.4 DevDocs] 2

#### 最新問題: 52

Adobe Commerce 開発者は、ProductRepositoryinterface を使用して製品のリストを生成し、スタンドアロン テーブル (つまり、EAV 属性ではない) に保存されているデータに対して supplier\_id フィルターを使用して製品を検索したいと考えています。

保守性を念頭に置いて、開発者はどのようにしてサプライヤー ID を検索に追加できるでしょうか?

- A. \Magento\Catalog\ViewModel\ProductRepository::getItems() に before プラグインを記述し、渡された検索条件を登録します。イベント オブザーバーを記述して、イベント catalog\_product\_collection\_load\_before をリスンします。登録された検索条件を反復処理し、見つかった場合は、必要な結合とフィルターをイベント コレクションに適用します。
- B. 仮定型にCUSummaryFilterを追加する

\Magento\Catalog\Model\Api\SearchCriteria\CollectionProcessor\ProductFilterProcessor for supplier\_id フィールド。カスタム フィルターで、渡された \$collection に必要な結合とフィルターを適用します。

- C. \Magento\Framework\Api\SearchCriteria\CollectionProcessorInterface に before プラグインを記述します: :

process()。\$searchCriteriaを反復処理する

supplier\_id に提供され、見つかった場合は、渡された \$collection に必要な結合とフィルターを適用します。

**Answer: B (メッセージを残す)**

開発者は仮定型にカスタムフィルターを追加できる

supplier\_id フィールドの \Magento\Catalog\Model\Api\SearchCriteria\CollectionProcessor\ProductFilterProcessor。

カスタム フィルターでは、開発者は渡された \$collection に必要な結合とフィルターを適用できます。これは、依存性注入とプラグインを使用して製品の検索条件を拡張するための推奨される方法です。検証済みの参照: [Magento 2.4 DevDocs] [Magento Stack Exchange] Adobe Commerce で、スタンドアロン テーブルに格納されている非 EAV 属性にカスタム フィルターを追加する必要がある場合、最も保守しやすい方法は、ProductFilterProcessor のカスタム フィルターを作成することです。このプロセッサを使用すると、カスタマイズされた検索条件の処理が可能になり、コア機能を変更したり、プラグインやオブザーバーに依存したりすることなく、カスタム結合やフィルターを含めるように拡張できます。

\* ProductFilterProcessor のカスタム フィルターが推奨される理由:

\* SearchCriteria\CollectionProcessor 内の ProductFilterProcessor は、製品コレクションのフィルタリングを処理するために特別に設計されています。これをカスタム フィルターで拡張することで、開発者はスタンドアロン テーブルに結合とフィルターを実装できます。

\* このアプローチはモジュール化されており、再利用可能であるため、ProductRepositoryInterface を利用するすべてのコードで supplier\_id フィルターをシームレスに適用できます。

\* カスタムフィルターの実装:

\* スタンドアロン テーブルを結合し、supplier\_id フィルターを適用するために必要なロジックを実装するカスタム フィルター クラスを定義します。

\* このカスタム フィルターを ProductFilterProcessor の di.xml 内の仮想タイプに登録し、supplier\_id を検索条件の一部として処理できるようにします。

\* オプションAとCがあまり適していない理由:

\* オプション A はイベント オブザーバーに依存しますが、これはモジュール性が低く、すべての製品コレクションのロード イベントをリスンする必要があるため、パフォーマンスに影響が出る可能性があります。

\* オプション C は機能的ではありますが、CollectionProcessorInterface::process() を変更する必要があり、これはより汎用的で、製品コレクションのフィルタリングに特化されていません。

\* 参考文献:

\* CollectionProcessor でのカスタム フィルターの使用に関する Magento 開発者向けドキュメント

\* 依存性注入と仮想型に関する Magento DevDocs

#### 最新問題: 53

Adobe Commerce 開発者は、卸売」顧客グループのすべてのユーザーに対して 15% の追加料金を実装するように求められています。ベスト プラクティスを念頭に置いて、これを実現するにはどうすればよいでしょうか。

**A.** 現在のユーザーがグループに属している場合に変更された合計を計算するための新しい合計コレクター クラスを宣言し、それをモジュールの etc/sales .xml ファイルに登録し、checkout\_cart\_index.xml および checkout\_index\_index.xml レイアウトを変更して、合計ブロックに新しい子を含めます。

**B.** 卸売」グループにのみ適用されるカート価格ルールを作成します。ルールの条件を指定せず、「アクション」セクションで、割引額」フィールドを -15 に設定して、商品価格の割引率」を適用するルールを指定します。

**C.** cataiog\_product\_get\_final\_price イベントのオブザーバーを作成します。現在の顧客が 卸売」グループに属しているかどうかを確認し、属している場合は、

**Answer: C** ([メッセージを残す](#))

`$observer->getEventC)` データから product を取得し、`$product->setData('final_price', $product->getData( 'final_price') * 1.15)` を呼び出します。

Explanation:

管理者に支払い方法が見つからない理由として考えられるのは、モジュール config.xml でノード can\_use\_internal が true に設定されていないことです。このノードは、支払い方法が管理者エリアで使用できるかどうかを決定します。false に設定されている場合、または省略されている場合、支払い方法は管理者の注文には使用できません。支払い方法を管理者専用にも有効にするには、ノード can\_use\_internal を true に設定し、ノード can\_use\_checkout を false に設定する必要があります。検証済みリファレンス: [Magento 2.4 DevDocs] [Magento Stack Exchange]

#### 最新問題: 54

開発者は Adobe Commerce Cloud プロジェクトに取り組んでおり、環境に展開されているサービスの接続データを取得したいと考えています。開発者には、これを行うために必要なすべての権限があります。

開発者が接続資格情報を取得するために選択する 2 つのオプションはどれですか? (2 つ選択してください。)

**A.** magento-cloud relationships CLI コマンドを実行します。

**B.** プロジェクト Web インターフェイス専用セクションからデータを取得します。

**C.** ece-tools env:config:show services コマンドを実行します。

**D.** SSH経由でサーバーに接続し、`$_ENV['services']`変数を読み取ります。

**Answer: A,B** ([メッセージを残す](#))

Adobe Commerce Cloud では、開発者のアクセスと利用可能なツールに応じて、さまざまな方法で、デプロイされたサービス (データベース、キャッシュ、その他のサービスなど) の接続データを取得できます。

\* magento-cloud relationships コマンドの使用:

\* この CLI コマンドは、データベースや Redis など、環境に関連付けられているサービスの接続資格情報をコマンド ラインから直接取得します。

\* プロジェクト Web インターフェース:

\* プロジェクト Web インターフェースには、開発者がサービス接続の詳細にアクセスできる専用セクションが用意されており、グラフィカル インターフェースを好むユーザーにとって便利です。

\* オプション A と B が正しい理由:

\* どちらの方法も有効であり、接続情報にアクセスするために頻繁に使用されます。オプション C (ece-tools env:config:show services) は存在しません。オプション D (\$\_ENV['services'] の読み取り) では SSH アクセスが必要となり、ユーザーフレンドリな形式で直接接続の詳細が提供されません。

\* 参考文献:

\* サービスへのアクセスに関する Adobe Commerce ドキュメント

#### 最新問題: 55

Adobe Commerce Cloud プロジェクトの販売者は、サブドメインを使用して Web サイトの 1 つを設定したいと考えています。

販売者はドメインを secondstore.example.com に設定することを検討しています。

magento-vars.php ファイルを編集し、ドメインチェックを適用して \$\_SERVER を設定します

["MAGE\_RUN\_CODE"] および \$\_SERVER["MAGE\_RUN\_TYPE"]。

このアクションを実行するにはどのファイルが必要ですか?

**A.** NGINX 仮想ホスト構成ファイルで secondstore.example.com サブドメイン ルートを構成します。

**B.** .magento/services.yaml で secondstore.example.com サブドメイン ルートを設定します。

**C.** .magento/routes.yaml で secondstore.example.com サブドメイン ルートを設定します。

**Answer: C (メッセージを残す)**

Adobe Commerce Cloud では、カスタム ドメインとサブドメインのルーティング構成は、.magento/routes.yaml ファイル内で管理されます。このファイルは、リクエストがアプリケーションに送信される方法を定義し、固有のサブドメインを持つさまざまなストアや Web サイトを設定するために不可欠です。

\* サブドメイン設定に .magento/routes.yaml を使用する:

\* secondstore.example.com のような新しいサブドメインを設定するには、でルートを設定する必要があります。

magento/routes.yaml。このファイルは、Adobe Commerce Cloud プロジェクトのすべてのルーティングとドメイン構成を処理します。

\* オプション C が正しい理由:

\* .magento/routes.yaml は、サブドメインを含むルーティング ルールを定義するために特別に設計されているため、このタスクに適しています。

\* オプション A (NGINX 構成) とオプション B (.magento/services.yaml) は、Adobe Commerce Cloud でドメイン ルーティングを直接管理しないため、正しくありません。

\* 構成例:

http://secondstore.example.com/:

タイプ: アップストリーム

アップストリーム: "mymagento:80"

参考文献:

\* Adobe Commerce Cloud のドキュメント: routes.yaml の設定

#### 最新問題: 56

Adobe Commerce 拡張機能を備えた物流会社は、すべての顧客に、確認済みの配送料金のリストを CSV ファイルで毎月送信します。販売者は、この CSV ファイルを Adobe Commerce の管理設定の

「ファイルアップロード」フィールドにアップロードします。

「ファイルアップロード」フィールドを表示し、実際の CSV インポートを処理するための 2 つの要件は何ですか? (2 つ選択してください。)

A. Add a custom backend model which extends `\Magento\Framework\App\Config\Value` and call `afterSave`:

```
// etc/adminhtml/system.xml
<field id="import_fees" ...>
  <label>Import shipment fees</label>
  <backend_model>My\Module\Model\Config\Backend\ImportFees</backend_model>
  ...
</field>
```

B. 

```
// \My\Module\Model\Config\Backend\ImportFees
class \My\Module\Model\Config\Backend\ImportFees extends \Magento\Framework\App\Config\Value
{
    public function afterSave()
    {
        /** @var \My\Module\Model\ImportFeed $importFees */
        $importFees = $this->importFeesFactory->create();
        $importFees->uploadAndImport($this);
        return parent::afterSave();
    }
}
```

C. Add a new field in `etc/adminhtml/system.xml` in `My_Module` with the file type:

```
<field id="import_fees" translate="label" type="file" sortOrder="1000" showInDefault="1">
  <label>Import shipment fees</label>
</field>
```

D. Add a new field in `etc/adminhtml/system.xml` in `My_Module` with a new custom type:

```
<field id="import_fees" translate="label" type="My\Module\Block\Adminhtml\Form\Field\ImportFees" sortOrder="1000" showInDefault="1">
  <label>Import shipment fees</label>
</field>
```

Answer: B,C (メッセージを残す)

Adobe Commerce (Magento) 管理設定で「ファイルアップロード」フィールドを表示し、アップロードされた CSV ファイルを処理するには、次の 2 つの主要な要件を満たす必要があります。

\* ファイルアップロード処理を扱うバックエンドモデル (オプション B) バックエンドモデル `\My\Module\Model\Config\Backend\ImportFees` を拡張する必要があります。

`\Magento\Framework\App\Config\Value` に移動し、`afterSave` メソッドをオーバーライドします。このメソッドには、アップロードされた CSV ファイルを保存した後に処理するために必要なロジックが含まれます。バックエンドモデルは、CSV ファイルの読み取りや必要な操作の実行などのカスタム処理を処理できる重要なコンポーネントです。

\* 説明: バックエンドモデルは、Magento で、設定値をカスタムの方法で保存するためのロジックを追加するために使用されます。ここでは、CSV ファイルを処理し、データベースに追加データを保存したり、その他の操作を実行したりする必要があるので、これが不可欠です。

\* 参考: Magento DevDocs とコアモジュールでは、管理者構成値にカスタム保存アクションが必要な場合は常にバックエンドモデルが利用されることが示されています。

\* `system.xml` でのファイルタイプ設定 (オプション C): `type="file"` 属性を使用して `etc/adminhtml/system.xml` でフィールドを定義すると、Magento はこのフィールドを管理設定でファイルアップロード入力としてレンダリングします。`<field>` タグには、適切な `id`、`label`、`type="file"`、およびバックエンド設定で定義するために必要なその他の属性が必要です。フィールドの `type` 属性を `file` に設定すると、設定内でファイルアップロード機能が有効になります。

\* 説明: system.xml 内のファイル タイプは、Magento にファイル入力フィールドをレンダリングするように指示します。これは、ユーザーが管理構成でファイルをアップロードし、バックエンド モデルで処理できるようにするために重要です。

\* 参考資料: Magento の公式ドキュメントでは、system.xml 構成ファイルを使用して Magento バックエンドのシステム構成にカスタム フィールドを追加する方法について説明しています。オプション A と D は、ファイル アップロード フィールドの設定と CSV インポートの処理の要件に具体的に対処していないため、正しくありません。オプション A はバックエンド モデルを構成しますが、必要なファイル タイプ定義がありません。オプション D は、ファイル フィールドのカスタム タイプを定義しますが、ファイルのアップロードまたは CSV 処理の要件に直接対処していません。

#### 最新問題: 57

Adobe Commerce 開発者は、Adobe Commerce Cloud (Pro Plan) の製品に重要な機能を導入しようとしています。機能に問題が発生した場合にロールバックできるように、スナップショットを作成したいと考えています。

スナップショットはどのように作成されるのでしょうか?

- A. Project Web Interface の専用ボタンを使用します。
- B. Cloud CLI for Commerce 専用コマンドを使用します。
- C. Adobe Commerce Cloud サポートへのチケットを作成します。

**Answer: B (メッセージを残す)**

Adobe Commerce Cloud (Pro Plan) に変更を展開する前にスナップショットを作成するには、スナップショットを作成するための専用コマンドを提供する Cloud CLI を使用することをお勧めします。これにより、展開後に問題が発生した場合に迅速にロールバックできます。

\* Cloud CLI を使用してスナップショットを作成する:

\* Adobe Commerce Cloud CLI ツールには、コードとデータの現在の状態をキャプチャするスナップショットを管理するためのコマンドが含まれており、重要な変更を行う前に復元ポイントを確保できます。

\* オプション B が正しい理由:

\* Cloud CLI はスナップショットを作成する最も直接的な方法であり、開発者はスナップショットがいつどのように生成されるかを制御できます。オプション A は Web インターフェースにスナップショット作成ボタンがないため正しくありません。また、オプション C はスナップショットの作成にサポートは必要ないため不要です。

\* 参考文献:

\* スナップショットの作成と管理に関する Adobe Commerce Cloud ドキュメント

#### 最新問題: 58

Adobe Commerce 開発者が cron ログを確認すると、次のジョブが毎日発生していることがわかります: main.INFO: Cron Dob inventory\_cleanup\_reservations が正常に終了しました。ただし、データベースの inventory\_reservation テーブルは空になっていません。なぜ inventory\_reservation テーブルにレコードが残っているのでしょうか?

- A. キャンセルされた注文に一致する予約のみが cron ジョブによって削除されます。
- B. 不要になった予約のみが cron ジョブによって削除されます。
- C. マルチソース インベントリの「自動クリーンアップ」機能が構成で無効になっています。

**Answer: B (メッセージを残す)**

inventory\_reservation テーブルにレコードが残っているのは、不要になった予約のみが cron ジョブによって削除されるためです。inventory\_reservation テーブルは、各注文の各製品の数量を追跡し、注文が発注、出荷、キャンセル、または返金されたときに各製品の予約を作成します。最初の予約の数量値は負で、後続の予約の数量値は正です。注文が完了すると、製品のすべての予約の合計はゼロになります。cron ジョブは、合計がゼロの予約のみをテーブルから削除し、未完了の注文に必要な予約はそのまま残します。検証済みリファレンス: [Magento 2.4 DevDocs] [Magento Stack Exchange]

#### 最新問題: 59

Adobe Commerce 開発者が、Magento\Framework\App\Cache\Proxy クラス内の save() 関数用の before プラグインを作成しました。このプラグインの目的は、特定の条件を満たすすべてのキャッシュ識別子にプレフィックスを追加することです。

プラグインが期待どおりに実行されないのはなぜですか？

- A. 同じ関数に対して定義された別の around プラグインは、呼び出し可能オブジェクトを呼び出しません。
- B. キャッシュ識別子は不変であり、変更できません。
- C. ターゲット Class は Magento\Framework\ObjectManager\NoninterceptableInterface を実装します。

**Answer: C (メッセージを残す)**

Magento 2 開発者向けのプラグイン (インターセプター) ガイドによると、プラグインは、パブリック クラス メソッドをインターセプトしてその前、後、またはその前後でコードを実行することで、パブリック クラス メソッドの動作を変更するクラス メソッドです。ただし、Magento 2 の一部のクラスは NoninterceptableInterface インターフェイスを実装しているため、それらのクラスに対してプラグインが生成されません。Magento\Framework\App\Cache\Proxy クラスは、Magento\Framework\ObjectManager\NoninterceptableInterface から拡張されているため、その 1 つです。したがって、ターゲット クラスが NoninterceptableInterface を実装しているため、プラグインは期待どおりに実行されません。検証済みの参照: <https://devdocs.magento.com/guides/v2.3/extension-dev-guide/plugins.html>

#### 最新問題: 60

Adobe Commerce 開発者は、マーケットプレイスで販売するモジュール (Vendor.ModuleName) を作成しています。新しいモジュールは宣言型スキーマを使用してデータベース テーブルを作成するため、開発者はモジュールが無効になったときにテーブルが削除されるようにする必要があります。

これを実現するために開発者は何をする必要がありますか？

- A. 開発者が行う必要がある作業はこれ以上ありません。モジュールが無効になり、bin/magento setup:upgrade が実行されると、テーブルは削除されます。
- B. 開発者が行う必要がある作業はこれ以上ありません。bin/magento module:uninstall vendor\_ModuleName が実行されると、テーブルは削除されます。
- C. Magento\Framework\setup\Patch\PatchRevertable インターフェイスを実装し、revert関数でテーブルを削除するスキーマ パッチを追加します。

**Answer: C (メッセージを残す)**

Magento 2 開発者向けの宣言型スキーマの概要ガイドによると、宣言型スキーマは、開発者がデータベースの最終的な望ましい状態を宣言し、冗長な操作を実行せずにシステムが自動的にそれに合わせて調整できるようにする新しい機能です。ただし、宣言型スキーマはモジュールのアンインストールや変更の元に戻すことをサポートしていません。モジュールが無効になっているときにテーブルを削除するには、開発者は Magento\Framework\setup\Patch\PatchRevertableInterface を実装し、元に戻す関数でテーブルを削除するスキーマ パッチを追加する必要があります。モジュールが bin/magento module:disable コマンドを使用して無効になっているときに、元に戻す関数が実行されます。検証済みの参照: <https://devdocs.magento.com/guides/v2.3/extension-dev-guide/declarative-schema/>

#### 最新問題: 61

Adobe Commerce 開発者は、管理パネルのカスタム フォームにファイル フィールドを追加するタスクを負っています。このフィールドでは、サイズが 2 MB 以下の .PDF ファイルのみを受け入れる必要があります。これまでに、開発者は、フォーム コンポーネント xml ファイルの fieldset ノード内に次のコードを追加しました。

```
<field name="pdf_file" formElement="fileUploader">
  <formElements>
    <fileUploader>
      <settings>
        <uploaderConfig>
          <param xsi:type="string" name="url">myvendor_mymodule/customForm/uploadPdf</param>
        </uploaderConfig>
      </settings>
    </fileUploader>
  </formElements>
</field>
```

開発者は検証をどのように実装するのでしょうか？

- A) HyVendor\MyModule\Controller\Adminhtml\CustomEntity\UploadPdf コントローラー内に検証を追加する

```

public function execute()
{
    $file = $this->fileUploaderFactory->create($this->getRequest()->getPdfFile());
    if($file->getExtension() == 'pdf') {
        throw new InvalidFileException(__('The file must be PDF.'));
    }
    if($file->getSize() >= '2048000') {
        throw new InvalidFileException(__('The file size must be less or equal than 2MB'));
    }
    return $this->resultFactory->create(ResultFactory::TYPE_PAGE);
}

```

B)

モジュールの di.xml 内に、コンストラクターの allowedExtensions と maxFileSize を指定する仮想型 forMyVendor\MyModule\Model\CustomPdfUploader を追加します。

```

<type name="MyVendor\MyModule\Model\CustomPdfUploader">
    <arguments>
        <argument name="allowedExtensions" xsi:type="string">pdf</argument>
        <argument name="maxFileSize" xsi:type="number">2048000</argument>
    </arguments>
</type>

```

C)

<settings> ノード内に次のコードを追加します。

```

<allowedExtensions>pdf</allowedExtensions>
<maxFileSize>2048000</maxFileSize>

```

A. オプションA

B. オプションB

C. オプションC

**Answer: B** ([メッセージを残す](#))

開発者は、モジュールの di.xml 内で、コンストラクターの allowedExtensions と maxFileSize を指定して、MyVendor\MyModule\Model\CustomPdfUploader の仮想型を追加できます。この方法では、開発者は既存のファイルアップローダークラスを再利用し、コアコードを変更せずに特定のフィールドに合わせてカスタマイズできます。検証済みの参照: [Magento 2.4 DevDocs] [Magento Stack Exchange]

有効な **AD0-E716** 問題集は GoShiken.com が提供された合格しやすい AD0-E716 試験問題集！ GoShiken.com が最新の **AD0-E716** 試験問題集を提供しています。GoShiken.com AD0-E716 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-E716 問題集をゲットする人はこちら: <https://www.goshiken.com/Adobe/AD0-E716-mondaishu.html> (**7230%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: 62

Adobe Commerce Cloud 開発者は、データベースを本番環境からステージング環境に転送した後でも、ステージング環境で支払い構成が引き続き有効であることを確認したいと考えています。構成が常に適切に設定されていることを確認するために、開発者は何を追加する必要がありますか？

A. 専用の core\_config\_data\_stg テーブル内の行。

B. プロジェクトレベルの環境変数。

C. 環境レベルの環境変数。

**Answer: C** ([メッセージを残す](#))

開発者は、ステージング環境で支払い設定が常に適切に設定されていることを確認するために、環境レベルの環境変数を追加する必要があります。環境変数は、Adobe Commerce Cloud アプリケーションとサービスの動作に影響を与える設定です。環境変数は、プロジェクトレベルまたは環境レベルで設定できます。プロジェクトレベルの変数はすべての環境に適用されますが、環境レベルの変数は特定の環境のプロジェクトレベルの変数をオーバーライドします。開発者は、環境レベルの変数を使用して、他の環境に影響を与えずにステージング環境の支払い設定をカスタマイズできます。検証済みの参照: [Magento 2.4 DevDocs]

#### 最新問題: 63

Adobe Commerce 開発者が、特定の注文を外部の会計システムにエクスポートするプロセスを作成しました。Magento CLI を使用して、コマンド `php bin/magento my_module` でこのプロセスを起動します。

注文: プロセス `--order_id=<order_id>` が必要です。

例: `php bin/magento my_module:order:process --order_id=1245`。

コマンドを設定する正しい方法は何ですか？

- A. 

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    parent::configure();
}

protected function values()
{
    return [new InputValue('order_id', InputValue::REQUIRED, 'Order ID')];
}
```
- B. 

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addOption('order_id', null, InputOption::VALUE_REQUIRED, 'Order ID');
    parent::configure();
}
```
- C. 

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addOption('order_id', null, InputOption::VALUE_REQUIRED, 'Order ID');

    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addArgument('order_id', InputArgument::REQUIRED, 'Order ID');
    parent::configure();
}
```
- D. 

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addArgument('order_id', InputArgument::REQUIRED, 'Order ID');
    parent::configure();
}
```

**Answer: D** ([メッセージを残す](#))

注文の処理に必須の `--order_id` などの必須引数を含む Magento CLI コマンドを適切に構成するには、`configure` 関数内で `addArgument` メソッドを使用するのが最適です。このメソッドは、コマンドに必要な引数を定義し、ユーザーが必要なデータを確実に提供できるようにします。

オプション D が正解である理由は次の通りです。

- \* 必須入力に addArgument を使用する:ここでは、addArgument メソッドを使用して order\_id を必須引数として宣言します。これは、パラメーターがコマンドの実行に不可欠であり、省略できない場合は addOption よりも適切です。InputArgument::REQUIRED を指定すると、コマンドは order\_id をユーザーが必ず指定するようにします。
  - \* 説明: addArgument は、コマンドが機能するために必要なデータに最適ですが、addOption は通常、オプションのパラメータに使用されます。order\_id は、処理する注文を識別するために不可欠であるため、ここでは addArgument が適切な選択となります。
  - \* 参考: Magento の公式開発者ドキュメントによると、addArgument は必須のコマンドライン引数に使用され、これは CLI コマンドで必要な入力を定義するための標準的な方法です。
  - \* 適切に構成されたコマンド名と説明: このオプションでは、setName メソッドと setDescription メソッドが正しく使用され、コマンドの名前と目的が指定されます。これにより、コマンドが自己記述的になり、CLI を使用する他の開発者や管理者にとっての使いやすさと読みやすさが向上します。
- オプション A、B、C は次のいずれかの理由で不正解です。
- \* 必須パラメータにはあまり適していない addArgument の代わりに addOption を使用します (オプション B)。
  - \* 同じパラメータを重複して定義します (オプション C)。
  - \* 必須の CLI パラメータについては、Magento のベスト プラクティスに準拠しないその他の非標準構成を使用します (オプション A)。

#### 最新問題: 64

1 日に 2 回実行される cron サービスを使用して開発された統合があり、請求書を作成できる注文がある場合に、統合された ERP システムに注文 ID を送信します。注文には、次のコードがすでにロードされています。

```
$order = $this->orderRepository->get($orderId);
```

ストアに作成する請求書があるかどうかを確認するために、Adobe Commerce 開発者はどのような実装を使用しますか？

A. 

```
if ($order->canInvoice()) {  
    // send integration to the ERP  
}
```

B. 

```
if ($order->hasInvoice()) {  
    // send integration to the ERP  
}
```

C. 

```
if (!$order->isPaymentReview()) {  
    // send integration to the ERP  
}
```

**Answer: A (メッセージを残す)**

注文が請求書の発行対象かどうかを確認する正しい実装は、\$order->canInvoice() メソッドを使用することです。このメソッドは、注文が完全に請求されていないかキャンセルされていないかなど、請求書を作成するために必要なすべての条件を注文が満たしているかどうかを確認します。

オプション A が正解である理由は次の通りです。

- \* 請求書の適格性を確認するための canInvoice() の使用: \$order->canInvoice() メソッドは、注文に対して請求書を生成できるかどうかを確認するために特別に設計されています。注文が請求書を発行できる状態にある場合にのみ true を返します。このため、このメソッドは、注文を請求書発行のために ERP システムに送信する必要があるかどうかを判断するのに適したメソッドになります。

- \* 説明: canInvoice() メソッドは Magento\Sales\Model\Order クラスの一部であり、さまざまな条件をチェックして請求が可能かどうかを判断します。これには、注文が完全に請求されていないかキャンセルされていないことを確認することが含まれます。これは、Adobe Commerce の注文処理ワークフローの重要な条件です。

- \* 参考資料: Magento の Order モデルに関する開発者向けドキュメントでは、特に ERP 統合などのプロセスを自動化する場合に、請求書の適格性を判断するための推奨アプローチとして canInvoice() が強調されています。

- \* 代替案と制限:

- \* オプション B: \$order->hasInvoice() メソッドは、注文に関連付けられた請求書がすでに存在するかどうかのみをチェックし、注文が新しい請求書の発行対象かどうかは示しません。注文に請求書が存在する場合は true を返しますが、このシナリオには適していません。

\* オプション C: \$order->isPaymentReview() メソッドは、注文が支払い確認状態にあるかどうかを確認しますが、これは請求書作成の適格性とは直接関係ありません。注文が請求可能かどうかに関する正確な情報は提供されません。  
canInvoice() を使用することで、開発者は、Adobe Commerce の注文処理ロジックに従って、cron ジョブが ERP システムに請求の準備ができている注文のみを送信することを保証します。

#### 最新問題: 65

メッセージキューは現在、queue/consumer-wait-for-messagesがtrueに設定されており、メッセージがキューに挿入されるまでコンシューマープロセスを実行できます。機能の一部は、モデルに格納されたデータによって駆動されます。

\Magento\variable\Model\variable であり、この値はコンシューマーの実行中に 1 回だけロードされます。変数が更新された場合は、消費されるメッセージごとに変数を再ロードしなくても、新しい値がメモリにロードされるようにコンシューマーを再起動します。

Adobe Commerce 開発者は、\Magento\Variable\Model\variable:: save() 関数の after プラグインを作成しました。

開発者はプラグインを使用してコンシューマーの再起動をトリガーするにはどうすればよいでしょうか？

- A. 関数 \Magento\Framework\MessageQueue\PoisonPill\PoisonPillPutInterface::put() を呼び出します。
- B. 関数 \Magento\Framework\MessageQueue\Consumers\TriggerRestartInterface::trigger() を呼び出します。
- C. グローバル キャッシュ キー trigger\_consumer\_restart を 1 に設定します。

**Answer: A (メッセージを残す)**

Adobe Commerce では、コンシューマープロセスを再起動する必要がある場合、PoisonPillPutInterfaceを使用できます。このインターフェイスのput()メソッドは「ポイズンピル」をトリガーし、実行中のコンシューマーに再起動を通知します。これは、更新されたデータをメモリに再ロードする必要がある場合に特に便利です。

\Magento\変数\Model\変数。

ポイズンピルの仕組み:

ポイズンピルメソッドは、メッセージキューコンシューマーに現在のプロセスを停止して再起動するように指示し、前回の起動以降に発生した構成またはデータの変更を取得できるようにします。

オプション A が正しい理由:

PoisonPillPutInterface::put() を呼び出すと、コンシューマーは再起動シグナルを受け取ります。これは、コンシューマーのライフサイクルの開始時にロードされたデータを更新する必要がある場合に最適です。

オプション B は、TriggerRestartInterface::trigger() が Magento フレームワークに存在しないため、正しくありません。

オプション C も、キャッシュ キーを設定するだけではコンシューマーの再起動がトリガーされないため、正しくありません。

実装:

変数が保存された後に PoisonPillPutInterface::put() をトリガーするには、save() メソッドで after プラグインを使用します。

参考文献:

メッセージキューとポイズンピルに関する Magento DevDocs

#### 最新問題: 66

Adobe Commerce 開発者がカスタム エンティティのインポーターとエクスポーターを作成しました。クライアントはこれを使用してエクスポートされたデータを変更し、ファイルを再インポートしてエンティティを一括更新します。

画像に関連する情報を JSON 形式で含むテキスト属性、media\_gallery があります。これはクライアントが変更したいフィールドではありませんが、エクスポートされたデータを編集するために使用しているソフトウェアによって変更され、正しくインポートできないようです。

開発者はこれをどうやって防ぐのでしょうか？

A) エクスポーターとインポーターの両方に \$\_transformAttrs クラス プロパティ配列を使用して属性のシリアライザー クラスを指定し、変換されるようにします。

```
protected $_transformAttrs = [  
    'media_gallery' => \Magento\Framework\Serialize\Serializer\Json::class  
];
```

B) インポートされたファイルから属性を削除するには、それを `s_strippedAttrs` クラスのプロパティ配列に追加します。

```
protected $_strippedAttrs = [
    'media_gallery'
];
```

C) `$_disatiedAttrs` クラス プロパティ配列に追加して、エクスポートされないようにします。

```
protected $_disabledAttrs = [
    'media_gallery'
];
```

A. オプションA

B. オプションB

C. オプションC

**Answer: A** ([メッセージを残す](#))

インポーター クラスとエクスポーター クラスの `_transformAttrs` クラス プロパティ配列を使用して、特定の属性のシリアライザー クラスを指定できます。シリアライザー クラスは、データがエクスポートまたはインポートされる時に、属性値をある形式から別の形式に変換するために使用されます。

この場合、開発者は `media_gallery` 属性の JSON データを文字列に変換するシリアライザー クラスを指定できます。これにより、クライアントがエクスポートされたデータを変更するために使用しているソフトウェアが JSON データを変更するのを防ぐことができます。

次のコードは、`media_gallery` 属性のシリアライザー クラスを指定する方法を示しています。

PHP の

クラス `MySerializer`

```
{
パブリック関数 serialize($value)
{
json_encode($value) を返します。
}
パブリック関数 deserialize($value)
{
json_decode($value) を返します。
}
}
```

```
$importer->setSerializer('media_gallery', MySerializer::class);
```

```
$exporter->setSerializer('media_gallery', MySerializer::class);
```

シリアライザー クラスが指定されると、データがエクスポートまたはインポートされる時に、`media_gallery` 属性の JSON データが文字列に変換されます。これにより、エクスポートされたデータを変更するためにクライアントが使用しているソフトウェアが JSON データを変更するのを防ぐことができます。

最新問題: 67

Adobe Commerce Cloud の PHP エラー ログにアクセスする 2 つの方法は何ですか? (2 つ選択してください。)

A. プロジェクト Web インターフェイスの専用エントリに移動します。

B. Adobe Admin Log アプリケーションを使用します。

C. SSH 経由でサーバーに接続し、ログ ファイルをローカライズします。

D. Cloud CLI for Commerce の専用コマンドを使用します。

**Answer: C,D** ([メッセージを残す](#))

#### 最新問題: 68

Adobe Commerce Cloud プロジェクトのマーチャントは、サブドメインを使用して Web サイトの 1 つを設定したいと考えています。マーチャントは、ドメインを `secondstore.example.com` に設定することを検討しています。

`magento-vars.php` ファイルを編集するだけでなく、ドメイン チェックを適用し、`$_SERVER["MAGE_RUN_CODE"]` と `$_SERVER["MAGE_RUN_TYPE"]` を設定します。このアクションを実行するにはどのファイルが必要ですか？

- A. NGINX 仮想ホスト構成ファイルで `secondstore.example.com` サブドメイン ルートを構成します。
- B. `.magento/services.yaml` で `secondstore.example.com` サブドメイン ルートを設定します。
- C. `.magento/routes.yaml` で `secondstore.example.com` サブドメイン ルートを設定します。

**Answer:** [\(解答を表示する\)](#)

開発者は、`.magento/routes.yaml` ファイルでサブドメイン ルートを構成することで、Web サイトの 1 つにサブドメインを設定できます。このファイルは、受信リクエストが Adobe Commerce Cloud プラットフォーム上のさまざまなアプリケーションまたはサービスにルーティングされる方法を定義します。開発者は、`secondstore.example.com` のルートを追加し、それを `example.com` と同じアプリケーションにマッピングする必要があります。また、開発者は、`secondstore.example.com` のアップストリーム変数を `MAGE_RUN_CODE` および `MAGE_RUN_TYPE` として指定する必要があります。検証済みリファレンス: [Magento 2.4 DevDocs] 3

#### 最新問題: 69

Adobe Commerce 開発者が `iframe` を追加し、外部ドメインの JavaScript ライブラリを Web サイトに組み込みました。その後、コンソールに次のエラーが表示されました。

コンテンツ セキュリティ ポリシーの指令に違反しているため、[URL] のフレーム化を拒否しました。

このエラーを修正するには、`csp_whitelist.xml` ファイルに追加する正しいポリシー ID は何でしょうか？

- A. フレームソースとスクリプトソース
- B. `default-src` と `object-src`
- C. フレーム祖先と接続元

**Answer:** [C \(メッセージを残す\)](#)

`frame-ancestors` ディレクティブは、現在のページを `iframe` に埋め込むことが許可されるドメインを指定します。`connect-src` ディレクティブは、`<script>` タグまたは `XMLHttpRequest` を通じて現在のページによってロードされることが許可されるドメインを指定します。

この場合、開発者は外部ドメインのページを埋め込む `iframe` を追加しました。外部ページのドメインが `frame-ancestors` ディレクティブにリストされていないため、コンテンツ セキュリティ ポリシー (CSP) によって `iframe` が読み込まれません。

このエラーを修正するには、開発者は外部ページのドメインを `frame-ancestors` ディレクティブに追加する必要があります。これを行うには、`csp_whitelist.xml` ファイルに次の行を追加します。

```
<frame-ancestors>https://www.example.com</frame-ancestors>
```

#### 最新問題: 70

Adobe Commerce 開発者がカートの機能を拡張する新しいモジュールを作成しています。モジュールは `app/code/CompanyName/ModuleName/` にインストールされています。

Adobe Commerce 開発者は、既存の `CartItemPrices GraphQL` スキーマを拡張してカスタム `base_price` フィールドを含めるにはどうすればよいでしょうか？

- A. `Resolve()` 関数に `base_price` フィールドを追加する `Magento\QuoteGraphQL\Model\Resolver\CartItemPrices` の `<preffrence>` を作成して構成します。
- B. モジュールの `etc/schema.graphqls` ファイルに以下を追加します。

```
type CartItemPrices {  
    base_price: Money!  
}
```

C)

モジュールの `etc/graphql/di.xml` ファイルに以下を追加します。

```
<type name="Magento\QuoteGraphQL\Model\Resolver\CartItemPrices">
  <arguments>
    <argument name="extendedConfigData" xsi:type="array">
      <item name="base_price" xsi:type="number"/>
    </argument>
  </arguments>
</type>
```



- C. オプションA
- D. オプションB
- E. オプションC

**Answer:** [\(解答を表示する\)](#)

開発者は、モジュールの etc/schema.graphqls ファイルに次のコードを追加することで、既存の CartItemPrices GraphQL スキーマを拡張し、カスタム base\_price フィールドを含めることができます。extend type CartItemPrices { base\_price: Money! @doc(description: "カートアイテムの基本価格")} このコードは、CartItemPrices 型に base\_price という新しいフィールドを追加し、それが Money 型であり、null 値ではないことを指定します。@doc ディレクティブは、スキーマ ドキュメントに表示されるフィールドの説明を追加します。開発者は、base\_price フィールドのカスタム リゾルバクラスを作成し、モジュールの di.xml ファイルで宣言する必要もあります。検証済みの参照: [Magento 2.4 DevDocs] [Magento Stack Exchange]

**Valid AD0-E716 Dumps** shared by GoShiken.com for Helping Passing AD0-E716 Exam! GoShiken.com now offer the **newest AD0-E716 exam dumps**, the GoShiken.com AD0-E716 exam **questions have been updated** and **answers have been corrected** get the **newest** GoShiken.com AD0-E716 dumps with Test Engine here: <https://www.goshiken.com/Adobe/AD0-E716-mondaishu.html> (72 Q&As Dumps, **30%OFF Special Discount: Freepdfdumps**)