



**Free Questions for Sitecore-XM-Cloud-Developer by  
dumpssheet**

**Shared by Kim on 29-02-2024**

**For More Free Questions and Preparation Resources**

**Check the Links on Last Page**

# Question 1

---

**Question Type:** MultipleChoice

---

A developer tried to exclude fields in a Sitecore Content Serialization package. Which file should they use to specify the excluded fields?

## Options:

---

A- user.json

B- sitecore.config

C- sitecore.json

D- xmcloud.build.json

Explanation:

The excluded fields feature is available from CLI 4.0 or later<sup>1</sup>. This way, you can reduce the size of your Sitecore Content Serialization package and avoid unnecessary serialization of fields that are not relevant for your deployment scenario<sup>1</sup>.

## Answer:

---

C

## Explanation:

---

[Sitecore Content Serialization structural overview](#)

[Configure excluded fields](#)

## Question 2

---

**Question Type:** MultipleChoice

---

What Sitecore Content Serialization command can a developer use to ensure that they don't forget to pull changes made within a developer-managed location?

### Options:

---

- A- The watch command
- B- The sync command
- C- The connect command
- D- The package create command

### Answer:

---

A

### **Explanation:**

---

The Sitecore serialization plugin provides the serialization command with the short form ser1. This command handles the serialization of items in and out of a Sitecore instance1. You can use the watch subcommand to monitor changes to content items in a Sitecore instance and automatically serialize the changes to your file system1. This way, you can ensure that you don't forget to pull changes made within a developer-managed location1.

The CLI serialization command

Sitecore Content Serialization structural overview

## **Question 3**

---

**Question Type: MultipleChoice**

---

When deploying to XM Cloud, how are Sitecore Content Serialization (SCS) serialized items created in the Content Management (CM) instance?

**Options:**

---

- A-** The XM Cloud deploy process uses the SCS CLI to do a push once the CM instance is active.
- B-** The XM Cloud deploy process creates an Items as Resources (IAR) file and layers this into the CM instance.
- C-** A build pipeline must be created to use the SCS CLI push commands to push content after the XM Cloud Deploy app finishes.
- D-** The XM Cloud deploy process uses serialized item batches through REST APIs at the XM Cloud CM to speed up serialization.

**Answer:**

---

D

**Explanation:**

---

The Sitecore Content Serialization (SCS) system serializes content items in and out of a Sitecore instance using YAML files<sup>1</sup>. When deploying to XM Cloud, the XM Cloud deploy process uses serialized item batches through REST APIs at the XM Cloud CM to speed up serialization<sup>2</sup>. This way, you can serialize large amounts of content items without affecting the performance of the Content Management (CM) instance<sup>2</sup>.

[Sitecore Content Serialization structural overview](#)

[Sitecore Content Serialization in XM Cloud](#)

## Question 4

---

**Question Type: MultipleChoice**

---

What does the default scope for a serialization include if the scope parameter is not defined?

**Options:**

---

- A- DescendantsOnly
- B- Singleitem
- C- ItemAndDescendants
- D- ItemAndChildren

**Answer:**

---

C

**Explanation:**

---

The default scope for a serialization in Sitecore Content Serialization (SCS) includes all the content items under the specified path and all their descendants<sup>1</sup>. You can configure what and how content items are included and excluded from serialization in a module file such as `Project.module.json`<sup>1</sup>. You can also use rules to configure the serialization of content item trees with different scopes<sup>1</sup>.

## Question 5

---

**Question Type:** MultipleChoice

---

What is the sitecore.json file in a development solution?

**Options:**

---

- A- The environment variables file for the Sitecore Docker deployment
- B- The configuration file for the Sitecore development tooling
- C- The solution integration file needed for deploying code changes to the environment
- D- A custom configuration file defining a solution for headless environments

**Answer:**

---

C

## Explanation:

---

The sitecore.json file is located in the root folder of the solution and it comes with the XM Cloud Foundation Head starter kit1. In here we can configure what \*.module.json files should be considered for serialization configuration, the plugins with it's particular versions and some general settings for the serialization1.

[Setup Sitecore Content Serialization](#)

[Sitecore Content Serialization structural overview](#)

## Question 6

---

**Question Type:** MultipleChoice

---

A developer has set up a local XM Cloud development environment and would like to connect Sitecore Pages to their locally running XM Cloud instance. However, after following the necessary steps, they're facing issues with the connection. Of the options below, what is the most likely potential reason for this issue?

### Options:

---

**A-** The value of the local storage entry "Sitecore.Pages.LocalXMCloudUrl" should be set to <https://pages.sitecorecloud.io>.



- B-** The browser developer tools console does not support the local storage functionality required for this connection.
- C-** The value of the local storage entry 'Sitecore.PagesXmCloud' should be set to enabled.
- D-** The environment variable 'SITECORE\_Pages\_CORS\_Allowed\_Origins' is not correctly configured to include pages.sitecorecloud.io.

**Answer:**

---

D

## Question 7

---

**Question Type:** MultipleChoice

---

If a developer wants to limit the serialization of items under a Navigation item to just the item itself and then one step below, what property should the developer add to the includes to indicate this limitation?

**Options:**

---

- A-** Scope
- B-** Nothing---this is the default.
- C-** Limit

D- Name

### Answer:

---

A

### Explanation:

---

According to the Sitecore XM Cloud Documentation for Developers<sup>1</sup>, the scope property is used to control the depth of the serialization tree. The scope property can have one of the following values<sup>2</sup>:

Self -- only the item itself is serialized, not its descendants.

Children -- the item and its immediate children are serialized, not the grandchildren or further descendants.

Subtree -- the item and all its descendants are serialized, regardless of the depth.

To limit the serialization of items under a Navigation item to just the item itself and then one step below, the developer should add the scope property with the value Children to the includes section of the serialization configuration file. For example<sup>3</sup>:

includes:

- name: Navigation

path: /sitecore/content/MySite/Navigation

scope: Children

## Question 8

---

### Question Type: MultipleChoice

---

A developer is using the Content Editor to set placeholder restrictions on a placeholder called headless-footer. Because these restrictions have been set on the placeholder settings within /sitecore/Layout/Placeholder Settings, which of the following statements is correct?

#### Options:

---

- A-** The restrictions apply for all placeholders called headless-footer within a single site in the XM Cloud Content Management instance.
- B-** The restrictions apply for all placeholders called headless-footer on every page of every site that is in the XM Cloud Content Management instance.
- C-** The restrictions only apply for a placeholder called headless-footer on a designated page.
- D-** The restrictions only apply for subsequently created versions of a placeholder called headless-footer.

## Answer:

---

B

## Explanation:

---

According to the Sitecore XM Cloud Documentation for Developers<sup>1</sup>, in the Content Editor, you can set placeholder restrictions for your layout placeholders. These restrictions are applied on all pages in the site. For example, if you add restrictions for a footer, these apply for all footers on every page of the site. In this way, you can control the content that content authors can place in layout placeholders. To add a placeholder setting for a site, you need to create a placeholder item in the Layout/Placeholder Settings folder and enter the placeholder key and the allowed components<sup>2</sup>.

The other options are not correct:

A) The restrictions apply for all placeholders called headless-footer within a single site in the XM Cloud Content Management instance. This option is too narrow, as the placeholder restrictions apply to all sites that use the same placeholder key, not just one site.

C) The restrictions only apply for a placeholder called headless-footer on a designated page. This option is too specific, as the placeholder restrictions apply to all pages that use the same placeholder key, not just one page. However, you can create placeholder restrictions on specific pages in the Experience Editor, if you want to override the default settings<sup>3</sup>.

D) The restrictions only apply for subsequently created versions of a placeholder called headless-footer. This option is incorrect, as the placeholder restrictions apply to all versions of the placeholder, not just the new ones.

[2: Set placeholder restrictions | Sitecore Documentation](#) [3: Set placeholder restrictions | Sitecore Documentation](#) [1: XM Cloud Documentation for Developers - Sitecore](#)

## Question 9

---

**Question Type:** MultipleChoice

---

Users in XM Cloud Pages want to modify and add specific components at a given point on the page. What would a developer do to allow the users to accomplish this task?

### Options:

---

- A-** Enable the correct placeholder settings of the components to be modified to include the page template used for the content item users are editing.
- B-** Include a placeholder as part of the layout markup and create a corresponding placeholder settings item with the appropriate allowed components.
- C-** Set the standard values for the page template to include component modification options and make sure the template used by the page inherits appropriately.
- D-** Enable the insert options of the components to be included for the page template and set the appropriate allowed components.

### Answer:

---

B

## Explanation:

---

According to the [Sitecore XM Cloud Documentation for Developers1](#), a placeholder is a named location on a page where components can be inserted. A placeholder can have one or more components assigned to it, and the content author can add, remove, or reorder the components in the XM Cloud Pages editor. To create a placeholder for a page, you need to follow these steps2:

In the Content Editor, create or select a rendering item that defines the metadata and parameters for your component.

In the rendering item, in the Data section, in the Layout field, enter the HTML markup for your component. Include a placeholder tag with a unique name, such as `<sc-placeholder key='my-placeholder' />`.

In the Content Editor, create or select a placeholder settings item that defines the allowed components for your placeholder. You can also clone an existing placeholder settings item and modify it as needed.

In the placeholder settings item, in the Data section, in the Key field, enter the same name as the placeholder tag, such as my-placeholder.

In the placeholder settings item, in the Data section, in the Allowed Controls field, select the components that you want to allow in the placeholder. You can also use the Edit button to open the Allowed Controls dialog box and select the components from a tree view.

Register the component and the placeholder settings in the Components builder and add them to the Components library.

In the XM Cloud Pages editor, drag and drop the component to the page and use the placeholder to insert other components as needed.

2: [Create a placeholder for a component | Sitecore Documentation](#) 1: [XM Cloud Documentation for Developers - Sitecore](#)

## Question 10

---

**Question Type:** MultipleChoice

---

A developer needs to create a site for a company and must define the data structures in Sitecore to create items and content. What should the developer use to define the data structures?

**Options:**

---

- A- Templates
- B- Rendering parameters
- C- Renderings
- D- Components data sources

**Answer:**

---

A

**Explanation:**

---

According to the Sitecore XM Cloud Documentation for Developers<sup>1</sup>, templates are the fundamental concept in Sitecore that determine the structure and behavior of content items. Templates define the fields, sections, and parameters that make up each item. Templates also enable inheritance, which allows items to share common fields and properties from their base templates. There are different types of templates in Sitecore, such as data templates, branch templates, and command templates<sup>2</sup>.

To create the data structures for a site, the developer should use data templates, which define the fields used to control how data is entered. Data templates form the framework around which items are built. Sitecore associates a data template with every item in the content tree. The field types determine the editing controls shown to the user and they contain the raw values that are stored in the database<sup>2</sup>.

The other options are not correct:

Rendering parameters are custom properties for a component that can be changed by the content author in the XM Cloud Pages editor. Rendering parameters allow the content author to customize the appearance and behavior of the component, but they do not define the data structure of the content item<sup>3</sup>.

Renderings are items that define the metadata and parameters for a component. Renderings link a component to a data template, a headless variant, and a rendering parameter template. Renderings are used to create and manage components in the Components builder, but they do not define the data structure of the content item<sup>4</sup>.

Component data sources are content items that provide the content for a component. Component data sources can have fields, such as text, image, or link, that can be edited by the content author in the XM Cloud Pages editor. Component data sources are based on data templates, but they are not templates themselves<sup>4</sup>.

2: Data definition and template overview | Sitecore Documentation 3: Create a component with rendering parameters | Sitecore Documentation 4: Create a component that uses a data source item | Sitecore Documentation 1: XM Cloud Documentation for Developers - Sitecore



## Question 11

---

**Question Type:** MultipleChoice

---

A developer wants to create a new component to display content from a datasource. Of the options listed below, what are the optimal steps to do this?

### Options:

---

- A-** Create or reuse a SXA Module, insert a new rendering, and use the JSS CLI to scaffold out a new component.
- B-** Create the new component in the Next.js application and import it to XM Cloud.
- C-** Create or reuse a SXA Module, clone an existing component that uses a data source item, and create the React component based on the original cloned component.
- D-** Create or reuse a SXA Module, clone an existing component that uses a context item, and create the React component based on the original cloned component.

### Answer:

---

C

## **Explanation:**

---

According to the [Sitecore XM Cloud Documentation for Developers1](#), a data source item is a content item that provides the content for a component. A data source item can have fields, such as text, image, or link, that can be edited by the content author in the XM Cloud Pages editor. To create a new component that displays content from a data source item, you need to follow these steps2:

Create or reuse a SXA Module, which is a folder that contains the renderings, data templates, and media items for your components.

Insert a new rendering item in the SXA Module, which defines the metadata and parameters for your component. You can also clone an existing rendering item that uses a data source item and modify it as needed.

Use the JSS CLI to scaffold out a new React component in your Next.js application, which defines the layout, style, and logic for your component. You can also copy and paste an existing React component that uses a data source item and modify it as needed.

Register the React component in the Components builder and add it to the Components library.

In the XM Cloud Pages editor, drag and drop the component to the page and select or create a data source item for the component. You can also edit the data source item's fields in the Properties panel.

2: [Create a component that uses a data source item | Sitecore Documentation 1: XM Cloud Documentation for Developers - Sitecore](#)

**To Get Premium Files for Sitecore-XM-Cloud-Developer Visit**

**<https://www.p2pexams.com/products/sitecore-xm-cloud-developer>**

**For More Free Questions Visit**

**<https://www.p2pexams.com/sitecore/pdf/sitecore-xm-cloud-developer>**

