



Sitecore-XM-Cloud-Developer Practice Questions

Shared by Long on 12-10-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page



Question 1

Question Type: MultipleChoice

A developer needs to configure a rendering in order to use dynamic placeholders. Which of the following steps is required? Select all that apply.

Options:

- A- Include the IDynamicPlaceholder base template in the Rendering Parameters template.
- B- Link the placeholder settings item to the rendering item.
- C- Define the placeholder key using a question mark (?) in the placeholder settings item.
- D- In the component TSX file, set a unique placeholder key value that has not yet been defined.

Answer:

A, B, C

Explanation:

To configure a rendering for the use of dynamic placeholders in Sitecore XM Cloud, the following steps are required:

Include the IDynamicPlaceholder base template in the Rendering Parameters template to enable dynamic assignment of IDs to the placeholder key.

Link the placeholder settings item to the rendering item to ensure that the dynamic placeholders are correctly associated with the rendering.

Define the placeholder key using a wildcard in the placeholder settings item, which allows for the generation of unique placeholder keys for every component on the page.

Question 2

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 Developers¹, 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 components².

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 settings³.
- 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 3

Question Type: MultipleChoice

Which of the following is part of the out-of-the-box technology stack for XM Cloud?

Options:

- A- jQuery
- B- Python
- C- Svelte Kit
- D- React

Answer:

D

Explanation:

XM Cloud includes a modern technology stack that supports various frameworks and libraries for development. React is part of this stack, as it is commonly used in conjunction with Sitecore's headless services and JSS (JavaScript Services) for building dynamic user interfaces.

Question 4

Question Type: MultipleChoice

Where can a developer create and edit data templates?

Options:

- A- Pages or Explorer
- B- Explorer or Portal
- C- Content Editor or Pages
- D- Content Editor

Answer:

C

Explanation:

According to the Sitecore XM Cloud Documentation for Developers¹, you create and edit data templates in the Template Manager or the Template Builder in the Content Editor, which provide the same functionality with the only difference being the root item of the content tree that is shown in the user interface. In the Template Manager, the root item of the content tree is /Sitecore/Templates.

You can also create additional data templates to be used in XM Cloud Pages², which are web pages that can be customized with layouts and content.

Question 5

Question Type: MultipleChoice

Which of these options best describes the purpose of the following query to the Experience Edge GraphQL schema?

```
query {  
  
  layout(site: "experienceedge", routePath: "/", language: "en") {  
  
    item {  
  
      homeltemPath: path  
  
      contentRoot: parent {  
  
        id  
  
        path  
  
      }  
  
    }  
  
  }  
  
}
```

Options:

- A- To get an item by ID
- B- To get the root item of a site
- C- To get the item layout for a URL
- D- To get information about a specific content site

Answer:

C

Explanation:

The query to the Experience Edge GraphQL schema is designed to retrieve the layout information for a specific URL, which in this case is the root path ("/") of a site named "experienceedge". This allows developers to access the Layout Service JSON for the item, which is essential for rendering the page in a headless setup.



Question 6

Question Type: MultipleChoice

A developer is working with Sitecore's Authoring and Management API to manage their Sitecore content using GraphQL. They want to explore and interact with the API using the GraphQL integrated development environment (IDE). Which option best statements is correct about using the GraphQL IDE?

Options:

- A- A developer needs to be in the sitecore\Admin role to access the GraphQL IDE.
- B- A developer needs to be in the sitecore\Sitecore Client Users role to access the GraphQL IDE.
- C- The GraphQL IDE provides read-only access to the API.
- D- The GraphQL IDE is only available for non-production environments to ensure secure interactions.



Answer:

B

Explanation:

Access to the GraphQL IDE for exploring and managing Sitecore content via the Authoring and Management API requires a developer to have the sitecore\Sitecore Client Users role. This role grants the necessary permissions to use the IDE for various operations, not just read-only access.

Question 7

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 8

Question Type: MultipleChoice

A developer can reference an external code base that includes components so they can be used in the Components builder interface. Which option best are possible results of adding React components to the Components builder? Select all that apply.

Options:

- A- The registered component is scoped to only one environment.

- B- A content author can enter values for the defined input parameters.
- C- The registered component is part of the Components library.
- D- The registered component can be previewed in the Component builder.

Answer:

B, C, D

Explanation:

According to the Sitecore XM Cloud Documentation for Developers¹, the Components builder is a tool that allows you to create and manage components for your XM Cloud Pages. You can use the Components builder to register React components from an external code base, such as a GitHub repository or a local folder. When you add a React component to the Components builder, the following results are possible:

- B) A content author can enter values for the defined input parameters. When you register a React component, you can define input parameters that allow the content author to customize the component's appearance and behavior. For example, you can define a text parameter that lets the content author enter the title of the component. The input parameters are displayed in the Properties panel of the XM Cloud Pages editor².
- C) The registered component is part of the Components library. When you register a React component, it becomes available in the Components library, which is a collection of components that you can use in your XM Cloud Pages. You can access the Components library from the Components builder or the XM Cloud Pages editor. You can also organize your components into categories and subcategories in the Components library³.
- D) The registered component can be previewed in the Component builder. When you register a React component, you can preview how it looks and behaves in the Component builder. You can also edit the component's code, input parameters, and metadata in the Component builder⁴.

The following result is not possible:

- A) The registered component is scoped to only one environment. When you register a React component, it is available in all the environments of your XM Cloud project. You cannot limit the component's availability to a specific environment. However, you can control the component's visibility and accessibility by using the Publish and Archive actions in the Component builder⁴.

2: Register a React component | Sitecore Documentation 3: Manage the Components library | Sitecore Documentation 4: Manage a component in the Component builder | Sitecore Documentation 1: XM Cloud Documentation for Developers - Sitecore

Question 9

Question Type: MultipleChoice

Which of the following tasks are performed by the publishing pipeline to create an accurate snapshot of the content available for publishing from the Content Management instance? Select all that apply.

Options:

- A- Calculating additional entities to publish
- B- Calculating dependencies
- C- Archiving expired content
- D- Resolving dependencies

Answer:

A, B, D

Explanation:

The publishing pipeline in Sitecore XM Cloud performs several tasks to ensure an accurate snapshot of content is available for publishing from the Content Management instance. These tasks include calculating additional entities to publish, calculating dependencies, and resolving dependencies. These steps are crucial for maintaining the integrity and consistency of the published content.

Question 10

Question Type: MultipleChoice

A developer is working on managing environments within the XM Cloud Deploy app. They have created a new environment for the project and linked it to a specific repository branch. However, they realize that they need to change the linked repository branch due to new developments in the project. Which steps should they follow to achieve this?

Options:

- A- Delete the current environment and create a new one with the desired repository branch.
- B- Unlink the current repository branch and then relink the desired branch to the environment through the project's 'Options' menu.
- C- Link the desired branch to the environment directly from the repository settings. The XM Cloud Deploy app will automatically update the linked branch.

D- Go to the project page, click the environment, choose 'Options,' and then 'Edit environment details.' In the dialog, select the desired branch from the "Link to branch" drop-down menu and save the changes.

Answer:

D

Explanation:

According to the Sitecore XM Cloud Documentation for Developers¹, you can link an environment to a specific branch of your repository in the XM Cloud Deploy app. This allows you to deploy different versions of your code to different environments. To change the linked branch for an existing environment, you need to follow these steps²:

On the navigation pane of the XM Cloud Deploy app, click Projects.

On the Projects page, click the project that contains the environment where you want to change the linked branch.

On the project page, click the environment where you want to change the linked branch.

On the environment page, click Options, then click Edit environment details.

In the Edit environment details dialog, select the desired branch from the Link to branch drop-down menu.

Click Save to confirm your changes.

2: Manage an environment in the XM Cloud Deploy app | Sitecore Documentation 1: XM Cloud Documentation for Developers - Sitecore

Question 11

Question Type: MultipleChoice

A developer is creating a component that has the following requirements:

- * Image Left, Title, Call to action
- * Image Right, Title, Call to action
- * Image Centered, Call to action

The component should be able to display the same content in all three options. What is the best way to build this?

Options:

- A- Three separate components with the compatible renderings set
- B- One component with logic in the component to change the view
- C- One component with three Headless variants
- D- One component with rendering parameters to control the layout

Answer:

D

Explanation:

According to the Sitecore XM Cloud Documentation for Developers¹, rendering parameters are a way to define custom properties for a component that can be changed by the content author in the XM Cloud Pages editor. Rendering parameters allow you to create flexible and reusable components that can adapt to different scenarios and layouts. To create a component with rendering parameters, you need to follow these steps²:

Create a data template that defines the rendering parameters for your component. For example, you can create a data template with a field called Layout that has three options: Left, Right, and Center.

Create a component that uses the data template as its rendering parameter template. For example, you can create a component called Image and Text that has an image, a title, and a call to action button.

In the component's code, use the rendering parameter value to control the layout of the component. For example, you can use conditional logic or CSS classes to change the position of the image based on the Layout value.

Register the 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 use the Properties panel to change the rendering parameter value. For example, you can select Left, Right, or Center from the Layout drop-down menu.

2: Create a component with rendering parameters | Sitecore Documentation 1: XM Cloud Documentation for Developers - Sitecore

Question 12

Question Type: MultipleChoice

A developer wants to deploy to XM Cloud, but the project is not using a source code provider

that has an out-of-the-box connector for XM Cloud. How can they deploy the site to XM Cloud?

Options:

- A- Use the Sitecore Cloud CLI to create a project and deployment.
- B- Create a Sitecore Update Package using SCS to include items and files.
- C- Use Vercel to connect to their source code provider and deploy.
- D- Create a custom connector to be used with the XM Cloud Deploy app.

Answer:

A

Explanation:

According to the Sitecore XM Cloud Documentation for Developers¹, the Sitecore Cloud CLI is a command-line tool that allows you to perform operations against local or remote XM Cloud instances. You can use the Sitecore Cloud CLI to deploy your existing XM Cloud-compatible Sitecore XM solution in the cloud, regardless of the source code provider you are using. To deploy your solution to XM Cloud using the Sitecore Cloud CLI, you need to follow these steps²:

Install the Sitecore Cloud CLI on your workstation.

Authorize the Sitecore Cloud CLI with your Sitecore Cloud Portal organization.

Create a project in the XM Cloud Deploy app using the Sitecore Cloud CLI.

Connect your local environment to the project using the Sitecore Cloud CLI.

Push your serialized items to the project using the Sitecore Cloud CLI.

Deploy your solution to an XM Cloud environment using the Sitecore Cloud CLI.

2: Walkthrough: Creating an XM Cloud project using the Sitecore CLI 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>

20%
DISCOUNT

P2P
exams