Free Questions for Sitecore-XM-Cloud-Developer

Shared by Chan on 04-10-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

In the content tree, an administrator-level user cannot see the Layouts or Templates nodes. Which of the following would explain the missing nodes?

Options:

A- A template has not been established for the Content Editor.

B- A layout has not been assigned for the content tree in the Content Editor.

C- The Layouts and Templates folders need to be created by a developer.

D- The 'Hidden items' check box has been unselected on the View tab of the ribbon.

Answer:

D

Explanation:

In Sitecore XM Cloud, if an administrator-level user cannot see the Layouts or Templates nodes in the content tree, it is likely because the "Hidden items" option is not selected in the Content Editor's View tab. These nodes are typically marked as hidden to streamline the

interface for users.

Question 2

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 {
homeItemPath: 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 3

Question Type: MultipleChoice

When an item is published, the Experience Edge for XM Connector publishes a static snapshot of the Layout Service output of that item. If a change is made to a data source item that is referenced on the page, how is that content made visible on the website?

Options:

A- A developer must publish the data source item.

B- A developer must publish the related page items.

C- A developer must publish to the web database.

D- A developer must reconnect to the Experience Edge Connector module.

Answer:

A

Explanation:

When a change is made to a data source item in Sitecore XM Cloud, the updated content becomes visible on the website after the data source item itself is published. This is because the Experience Edge for XM Connector publishes a static snapshot of the Layout Service

output, and any changes to the data source items require republishing to reflect on the website.

Question 4

Question Type: MultipleChoice

A developer wants to create a webhook that sends an HTTP request to a specified endpoint when the workflow moves to the approved state. What type of webhook should they use?

Options:

A- Submit handler

B- Event handler

C- Submit action

D- Validation action

Answer:

C

Explanation:

In Sitecore XM Cloud, a webhook submit action is used to send an HTTP request to a specified endpoint when an item changes workflow state or a workflow command runs. Therefore, for a developer wanting to create a webhook that triggers when the workflow moves to the approved state, a webhook submit action would be the appropriate choice.

Question 5

Question Type: MultipleChoice

A developer is tasked with creating an item using the Sitecore Authoring and Management GraphQL API. Which of the following GraphQL mutations is the correct way to create a new item?

Options:

A- createOrUpdateItem

B- create TemplateItem

C- createItem

D- updateItem

Answer:

C

Explanation:

The correct GraphQL mutation to create a new item in Sitecore XM Cloud is createItem. This mutation allows developers to specify the necessary details such as the item's name, template ID, parent ID, language, and fields to create a new content item within the Sitecore content tree.

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 of the following 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

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 8

Question Type: MultipleChoice

What information can be found in the deployment logs?

Options:

A- Content Management and Content Delivery deployment logs

B- Content Management instance and Rendering Hosting logs

C- Provisioning, Deployment, and Publishing information

D- Provisioning, Build, Deployment, and Post Action information

Answer:

D

Explanation:

According to the Sitecore XM Cloud Documentation for Developers1, the deployment log displays the progress, status, warnings, and errors of an XM Cloud deployment. From the deployment log, you can also cancel a running deployment, rerun a failed deployment, promote the deployment to another environment, download logs, or open the XM Cloud Dashboard. The logs include information about:

Provisioning -- the process of creating and configuring the resources required for the environment, such as the Content Management and Content Delivery instances, the database, the storage, and the network.

Build -- the process of compiling the source code, running tests, and generating the deployment package.

Deployment -- the process of deploying the package to the environment and applying any configuration changes.

Post Action -- the process of performing any additional tasks after the deployment, such as publishing, indexing, or clearing caches.

1: XM Cloud Documentation for Developers - Sitecore

Question 9

Question Type: MultipleChoice

A developer wants to add a new language to a headless SXA site. Which steps are required to add a new language in XM Cloud?

Options:

A- Install a language pack on the Content Management instance and then add the language in /sitecore/system/languages.

B- Add language in /sitecore/system/languages. Then, on the content item, change the language dropdown to the new language and add a new version.

C- Add language in /sitecore/system/languages, right click the site root, and go to scripts -> Add Site language.

D- They must enter the country code in the Language field. Then, on the content item, change the language dropdown to the new

language and add a new version.

Answer:

C

Question 10

Question Type: MultipleChoice

To connect Sitecore Pages to your local XM Cloud instance, which steps are involved?

Options:

A- Open the local Content Management instance and set the predefined rendering host field value in Sitecore to https://pages.sitecorecloud.io

B- Open Pages and create an entry in Local Storage for the key 'Sitecore.Pages.LocalXmCloudUrl' with the following value: https://xmcloudcm. localhost

C- Open Pages and create an entry in Local Storage for the key 'Sitecore.Pages.LocalXmCloudUrl' with the following value: https://pages.sitecorecloud.io

D- Open the local Content Management instance and set the predefined rendering host field value in Sitecore to

https://xmcloudcm.localhost

Answer:

B

Explanation:

According to the Sitecore XM Cloud Documentation for Developers1, after setting up your local XM Cloud development environment, Sitecore Pages is not connected to your locally running XM Cloud instance. You can connect your local XM Cloud application to Sitecore Pages by setting a local storage key in your browser. This allows you to use Sitecore Pages to edit content and layouts on your local front-end application. To connect Sitecore Pages to your local XM Cloud instance, you need to follow these steps2:

In your browser, navigate to the Sitecore Pages application.

Open the developer tools console of your browser and find the local storage section. The location differs between browsers:

For Chrome and Edge, open the Application panel. In the Storage section, expand the Local Storage section.

For Firefox, open the Storage panel, and expand the Local Storage section.

In the Local Storage panel, click the entry for Sitecore Pages 9 to reveal the local storage entries.

Create a new entry by adding a new row with the following values:

Key: Sitecore.Pages.LocalXmCloudUrl.

Value: the URL of your local XM Cloud instance. For example, 10.

Refresh the 9 page to see your local front-end application.

2: Connect Sitecore Pages to your local XM Cloud instance 1: XM Cloud Documentation for Developers - Sitecore

Question 11

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