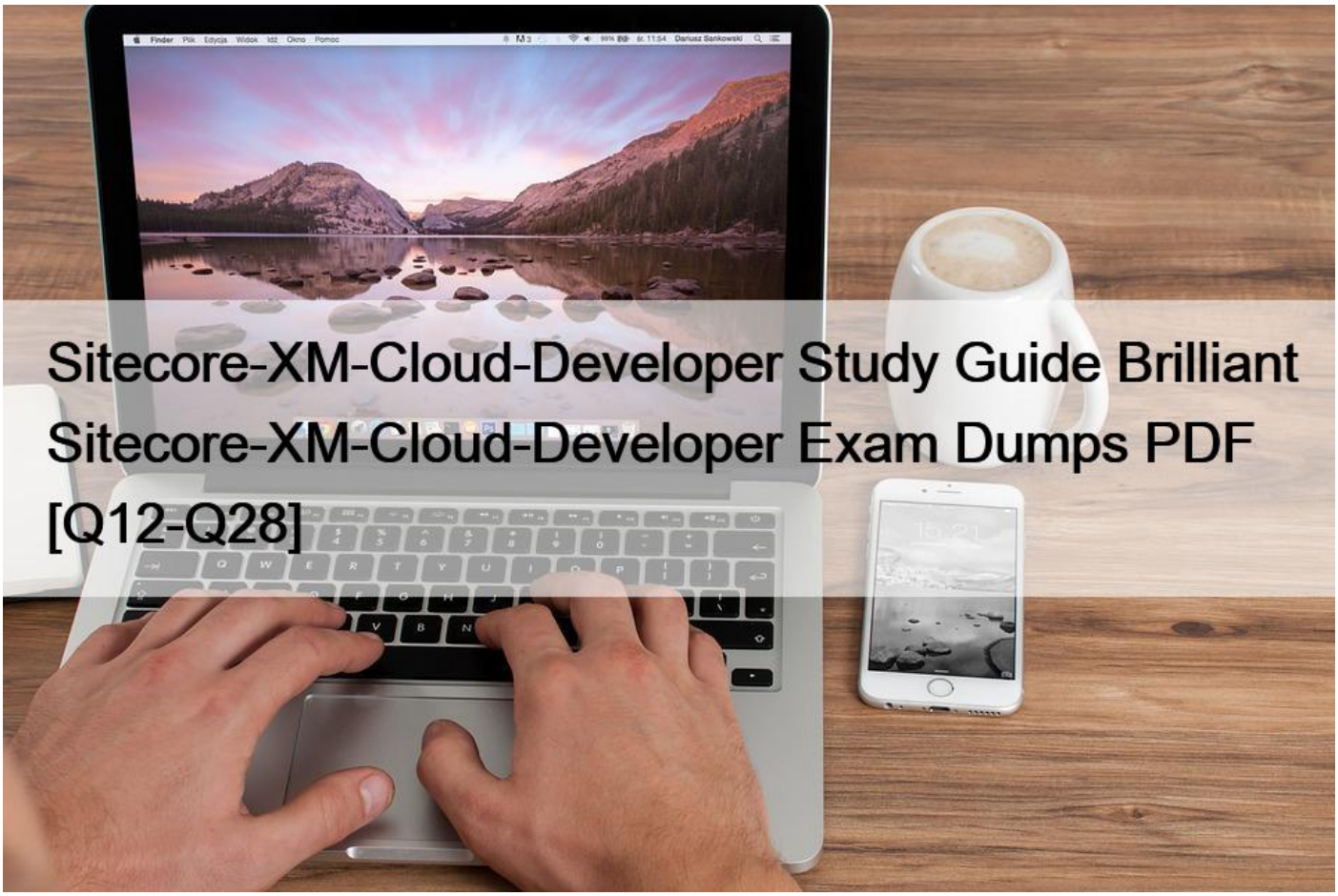
Sitecore-XM-Cloud-Developer Study Guide Brilliant Sitecore-XM-Cloud-Developer Exam Dumps PDF [Q12-Q28



Sitecore-XM-Cloud-Developer Study Guide Brilliant Sitecore-XM-Cloud-Developer Exam Dumps PDF View Sitecore-XM-Cloud-Developer Exam Question Dumps With Latest Demo NEW QUESTION 12

Which of the following is part of the out-of-the-box technology stack for XM Cloud?
* jQuery
* Python
* Svelte Kit
* React

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.

References:The Sitecore XM Cloud documentation mentions the use of modern frameworks and libraries, including React, as part of its technology stack1. This aligns with the industry’s move towards more flexible and modular front-end development practices.

NEW QUESTION 13

A developer needs to log in to the XM Cloud Deploy app to create a project using a starter template but requires the correct permissions. Which of the following should happen?
* An Organization Admin or Organization Owner must update their role.
* An Organization Admin or Organization Owner must update their role using the XM Cloud Deploy app.
* Only the Organization Owner can update their role in the Sitecore Cloud Portal.
* Only the Organization Owner can update their role using the XM Cloud Deploy app.
According to the Sitecore XM Cloud Documentation for Developers1, to use the XM Cloud Deploy app, you must be an Organization Admin or Organization Owner in your Sitecore Cloud Portal organization.An Organization Admin or Organization Owner can grant a team member access to the XM Cloud Deploy app by changing their organization role to Admin2. This can be done in the Sitecore Cloud Portal, not in the XM Cloud Deploy app.

2:Invite team members to your Sitecore Cloud Portal organization1:XM Cloud Documentation for Developers

– Sitecore

NEW QUESTION 14

What does the default scope for a serialization include if the scope parameter is not defined?
* DescendantsOnly
* Singleitem
* ItemAndDescendants
* ItemAndChildren
The default scope for a serialization in Sitecore Content Serialization (SCS) includesall the content itemsunder the specified path andall their descendants1.You can configure what and how content items are included and excluded from serialization in a module file such as Project.module.json1.You can also use rules to configure the serialization of content item trees with different scopes1.

References:

Sitecore Content Serialization structural overview

The default serialization format

NEW QUESTION 15

A build to XM Cloud fails. Which options are available for a developer to diagnose the error?
* The Deploy app system status page
* The Content Management instance logs for the environment
* The project details page in the Deploy app
* The deployment log files for the environment
According to the Sitecore XM Cloud Documentation for Developers1, the deployment log files contain the progress, status, warnings, and errors of an XM Cloud deployment. You can access the deployment log files from the XM Cloud Deploy app by clicking the Logs button on the environment page. The log files include information about the provisioning, build, deployment, and post action phases of the deployment.You can also download the log files for offline analysis or troubleshooting2.

The other options are not as useful for diagnosing a build failure:

The Deploy app system status page shows the overall health and availability of the XM Cloud Deploy app, not the status of individual deployments3.

The Content Management instance logs for the environment show the runtime logs of the Content Management instance, not the build logs of the deployment4.

The project details page in the Deploy app shows the general information and settings of the project, not the details of the deployment1.

2:View the deployment log files | Sitecore Documentation3:Check the system status of the XM Cloud Deploy app | Sitecore Documentation4:Access the Content Management instance logs | Sitecore Documentation1:Manage a project in the XM Cloud Deploy app | Sitecore Documentation

NEW QUESTION 16

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?
* Templates
* Rendering parameters
* Renderings
* Components data sources

According to the Sitecore XM Cloud Documentation for Developers1, 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 templates2.

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 database2.

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 item3.

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 item4.

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 themselves4.

2:Data definition and template overview | Sitecore Documentation3:Create a component with rendering parameters | Sitecore Documentation4:Create a component that uses a data source item | Sitecore Documentation1:XM Cloud Documentation for Developers – Sitecore

NEW QUESTION 17

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?
* Install a language pack on the Content Management instance and then add the language in

/sitecore/system/languages.
* Add language in /sitecore/system/languages. Then, on the content item, change the language dropdown to the new language and add a new version.
* Add language in /sitecore/system/languages, right click the site root, and go to scripts -> Add Site language.
* 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.

NEW QUESTION 18

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?
* Use the Sitecore Cloud CLI to create a project and deployment.
* Create a Sitecore Update Package using SCS to include items and files.
* Use Vercel to connect to their source code provider and deploy.
* Create a custom connector to be used with the XM Cloud Deploy app.
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 CLI1:XM Cloud Documentation for Developers – Sitecore

NEW QUESTION 19

Which of the following statements accurately describes the purpose of Headless variants in Sitecore XM Cloud?
* Headless variants define how a rendering appears and what content it displays.
* Headless variants determine the order in which renderings are displayed on a webpage.
* Headless variants enable the creation of custom rendering items for components from scratch.
* Headless variants allow renderings to be edited and customized in the Content Editor.
According to the Sitecore XM Cloud Documentation for Developers1, headless variants are a way to create different versions of a rendering that can be used in different contexts or scenarios. A headless variant defines the layout, style, and content of a rendering using HTML, CSS, and JavaScript. You can create headless variants for your components in the Components builder and use them in your XM Cloud Pages.Headless variants allow you to reuse the same component with different appearances and contents without creating multiple renderings2.

2:Create a headless variant for a component | Sitecore Documentation1:XM Cloud Documentation for Developers – Sitecore

NEW QUESTION 20

What is the sitecore.json file in a development solution?
* The configuration file for the Sitecore development tooling
* The environment variables file for the Sitecore Docker deployment
* A custom configuration file defining a solution for headless environments
* The solution integration file needed for deploying code changes to the environment

NEW QUESTION 21

A developer changed the CSS grid system after a site was created. Which of the following can happen due to references to the previous grid system?
* The changes break the layout of the site.
* The Grid Setup item and its dependencies were automatically updated.
* A new custom theme with a grid definition item was created.
* The new grid was automatically added as one of the grid options.

Changing the CSS grid system after a site has been created can lead to broken layouts due to references to the previous grid system. This is because grid systems have specific rules, classes, and options that are not automatically updated or compatible with the new system, requiring manual changes to ensure the layout functions correctly.

References:The Sitecore XM Cloud documentation provides insights into the implications of changing the grid system, emphasizing the need for manual updates to avoid breaking the site layout1.

NEW QUESTION 22

The XM Cloud Pages editor comes with a set of devices (for example, Desktop Regular and Mobile) with predefined settings. Which of the following fields must a developer specify to add and configure an additional device?
* Device height, visibility, an icon to represent the device
* Device height, visibility, responsive breakpoints
* Device width, visibility, an icon to represent the device
* Device width, visibility, responsive breakpoints

When adding and configuring an additional device in the XM Cloud Pages editor, a developer must specify the device width in pixels, the visibility options (Disabled, Fixed, Optional), and an icon to represent the device.

These settings are crucial for simulating how content will appear on different devices within the Pages editor.

References:The Sitecore XM Cloud documentation provides detailed instructions on adding and configuring devices in the Pages editor, including specifying the device width, visibility options, and selecting an icon1.

NEW QUESTION 23

Why should a developer assign security to roles instead of users when setting up security on a site?
* Security roles can assign individual rights and permissions to be configured by users.
* Role-based security allows multiple users to have the same username.
* Security can be managed more efficiently as the rights and permissions apply to a group of users.
* Individual users with a role can set and configure their own item security permissions.

According to the Sitecore XM Cloud Documentation for Developers1, security roles define how different users access different types of records in Sitecore. To control access to data and resources, you can create or modify security roles and change the security roles that are assigned to your users. A user can have multiple security roles. Security role privileges are cumulative.Users are granted the privileges that are available in each role that’s assigned to them2.

This makes it easier for you to manage your security system because you do not necessarily have to assign access rights for each item in your content tree.Instead, you only need to assign and manage the access rights on the parent items and then specify whether their descendants can inherit these access rights3.

NEW QUESTION 24

Which of the following must be done to delete or remove a user from an administrator role?
* Delete the user or remove their administrator role from the Sitecore Cloud Portal.
* Create a new user account and assign a new role from the XM Cloud Deploy app.
* Delete the user or remove their administrator role from the XM Cloud Deploy app.
* Create a new user account and assign a new role from the Sitecore Cloud Portal.

According to the Sitecore XM Cloud Documentation for Developers1, the Sitecore Cloud Portal is where you can manage access to your DXP products, create users and manage their access to all your Sitecore DXP products.To delete or remove a user from an administrator role, you need to follow these steps2:

Sign in to the Sitecore Cloud Portal using your administrator account.

In the left navigation pane, click Users.

In the Users list, find the user you want to delete or remove from the administrator role and click the Edit icon.

In the Edit user dialog box, you can either delete the user by clicking the Delete user button or remove the administrator role by deselecting the Administrator checkbox.

Click Save to confirm your changes.

1:XM Cloud Documentation for Developers – Sitecore2:Manage users in the Sitecore Cloud Portal – Sitecore

NEW QUESTION 25

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?
* Create or reuse a SXA Module, insert a new rendering, and use the JSS CLI to scaffold out a new component.
* Create the new component in the Next.js application and import it to XM Cloud.
* 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.
* 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.

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 Documentation1:XM Cloud Documentation for Developers – Sitecore

NEW QUESTION 26

When a developer selects the base templates for a new data template, what happens if those base templates contain a field with the same name?
* The field will not be displayed on the data item.
* Duplicate field names will appear on the data item.
* XM Cloud will add a suffix to the field names to identify them.
* The fields will be merged on the data item.

According to the Sitecore XM Cloud Documentation for Developers1, a template inherits sections and fields from its base templates. When there are several base templates associated with a data template, fields and sections from inherited templates aremerged.To configure the base templates for a data template, you can use theBase Templatesdialog box in theTemplate Manageror theContent Editor1.

NEW QUESTION 27

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

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 asmy-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 Documentation1:XM Cloud Documentation for Developers

– Sitecore

NEW QUESTION 28

A developer needs to configure a rendering in order to use dynamic placeholders. Which of the following steps is required? Select all that apply.
* Include the IDynamicPlaceholder base template in the Rendering Parameters template.
* Link the placeholder settings item to the rendering item.
* Define the placeholder key using a question mark (?) in the placeholder settings item.
* In the component TSX file, set a unique placeholder key value that has not yet been defined.

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

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

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

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

References:The Sitecore XM Cloud documentation provides a walkthrough on configuring components to use dynamic placeholders, detailing the steps required to set up dynamic placeholders1.This includes adding theIsRenderingsWithDynamicPlaceholdersproperty to the component item, including theIDynamicPlaceholderbase template in the Rendering Parameters template, defining the placeholder key using a wildcard, and linking the placeholder settings item to the component1.