PassQueen
a.k.a dumpsqueen.com

Acquia Certified Site Builder - Drupal 8

Acquia Acquia-Certified-Site-Builder-D8

Version Demo

Total Demo Questions: 10

Total Premium Questions: 101

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Planning	7
Topic 2, Content Modeling	14
Topic 3, Site Display	15
Topic 4, Content Authoring	12
Topic 5, Site Building	36
Topic 6, Maintaining and Troubleshooting	17
Total	101

QUESTION NO: 1

What is meant by headless Drupal? Choose 2 answers

A. Just another phrase for "decoupled Drupal"

B. Applying security updates automatically to Drupal installation

C. Exposing Drupal content to other systems such as native applications.

D. Publishing content automatically in Drupal.

ANSWER: A C

Explanation:

Headless Drupal is commonly used interchangeably with decoupled Drupal: Drupal remains responsible for managing structured content, users, workflows, and editorial administration, while the presentation layer is separated from Drupal. In a fully headless implementation, Drupal does not provide the visitor-facing rendered website; a separate client or application is responsible for displaying the content. This separation lets teams use Drupal's mature content-management capabilities without requiring the consumer experience to be built with Drupal's theming layer.

Exposing Drupal content to other systems such as native applications is a core outcome of this architecture. Drupal can make content available through web-service interfaces, including JSON:API and REST, so a native mobile app, JavaScript frontend, kiosk, or another external consumer can request and render the same managed content. Drupal's documentation describes decoupled Drupal as using Drupal as a content repository while another system handles presentation, and its JSON:API documentation explains the built-in mechanism for exposing Drupal entities as an API. See Drupal's Decoupled Drupal guide and Drupal's JSON:API module documentation.

QUESTION NO: 2

Your team uses development, staging, and production environments. A site builder has added a new vocabulary, fields, and a View in development. The changes have been tested and are ready to move to staging.

Which TWO statements describe an appropriate configuration-management workflow? Choose 2 answers

A. Export the tested configuration from development and commit the exported files to version control.

B. Deploy the configuration files to staging and import the configuration there.

C. Copy the development database over the staging database to transfer the new View and fields.

D. Recreate the vocabulary, fields, and View manually in each environment.

ANSWER: A B

Explanation:

The tested configuration should be exported from development into the site's configuration sync directory and committed to version control. The same configuration files can then be deployed to staging and imported there, allowing staging to receive the vocabulary, fields, and View definitions that were tested in development.

Configuration deployment should not be performed by copying a development database over a staging or production database, because database copies can overwrite changing content, users, comments, and other active data. Drupal's configuration system is designed to deploy configuration separately from content. See the Drupal Configuration Management documentation.

QUESTION NO: 3

A marketing team needs a reusable promotional block to appear only on Article pages tagged with the taxonomy term "Featured." The block must not appear on other content types or on Articles without that term.

Which TWO block visibility conditions can be used together to meet this requirement? Choose 2 answers

A. Use the Content types visibility condition and select Article.

B. Use the Taxonomy term visibility condition and select Featured.

C. Use the Pages visibility condition and enter the path "/article".

D. Use the Roles visibility condition and select Content editor.

E. Use the Language visibility condition and select English.

ANSWER: A B

Explanation:

The *Content types* visibility condition can limit a block placement to Article nodes. The *Taxonomy term* visibility condition can further limit that same block to nodes tagged with the Featured term. When both conditions are configured, Drupal evaluates the placement so that the promotional block appears only where the required content type and classification apply.

Page-based visibility can restrict a block by URL path, but it does not automatically identify all current and future Articles assigned to a taxonomy term. Role visibility controls who sees a block, not which content is being viewed. Drupal block placements support configurable visibility conditions through the Block layout interface. See the Drupal User Guide: Configuring block visibility.

QUESTION NO: 4

Your site review team reports that the new Product pages look good, but takes a long time to load.

What are two possible causes? Choose 2 answers

A. images are being loaded without using Image styles to load correctly sized versions

B. Internal Dynamic Page Cache is installed, but the reviewers are testing while logged in

C. The permission "View cached pages" is not set for Anonymous users.

D. The Performance setting for Caching is set to "no caching."

ANSWER: A C

Explanation:

Using images without Image styles can substantially increase Product-page load time because Drupal may deliver the original, full-resolution image files even when only smaller display dimensions are needed. Image styles create derivative images at the required size, reducing transferred bytes and browser image-processing work. For image-heavy product listings or detail pages, this is a common and significant performance concern. Drupal's Image module documentation explains that image styles generate transformed image derivatives for display: Drupal image documentation.

The *View cached pages* permission is also important for anonymous-page caching. Internal Page Cache is intended to serve cached responses to anonymous visitors; anonymous users must be allowed to view cached pages for that cache to be usable for their requests. If that permission is absent, anonymous visitors cannot benefit from cached page delivery and Drupal must perform more work to build responses, which can make page loads slower. Drupal documents Internal Page Cache as the cache used to improve performance for anonymous users: Drupal Internal Page Cache documentation.

QUESTION NO: 5

You are asked to implement a "employee of the month" block into your corporate website The block should show name and picture of the employee and should be editable over the normal block layout All the employees are users of the website

How do you implement the block?

A. Create a user view mode with user name and picture. Add a user reference field to a custom block type and configure the field to use the new user view mode.

B. Add a custom block and select the type "user account information", activate the user name and picture fields

C. Install the user_blocks module from Drupal org, select the fields you need and place block via block layout

D. Create a new Users View with user name field and user picture and an exposed filter on uid Add a block display to the view and place the block in the block layout

ANSWER: A

Explanation:

Create a user view mode with user name and picture, then add a user reference field to a custom block type and configure that field to render the referenced user in the new view mode. This uses Drupal's standard content-modeling and block-placement workflow. A custom block type provides an editor-friendly block instance that can be created and placed through the normal Block layout administration page. The user reference field lets an editor select the employee from existing website user accounts without duplicating the employee's name or image in the block.

A dedicated user view mode controls exactly how the selected account is presented in this context. Configure the view mode so that it displays the user name and picture, then set the reference field's display formatter to render the referenced entity using that view mode. This keeps presentation configuration reusable and separate from the selected employee data. If the user's displayed name or picture changes later, the block automatically reflects the current user-account data. Drupal's display-mode system is intended for defining different presentations of the same entity in different contexts; see Drupal's display modes documentation. Custom block types and their fields are also managed through the standard block system described in the Block module overview.

QUESTION NO: 6

You have installed a custom theme for your website, and you notice the theme displays a Druplicon as the logo in the upper left corner As much as you love Drupal, you would like to remove this logo ankdisplay your company's logo instead

How would you do this in the Drupal admin interface? Choose 2 answers

A. Go to Appearance > Settings, select the active theme, and upload your new logo.

B. Create a custom block. Include your logo in the block and place it in the appropriate region Uncheck the "Use the logo supplied by the theme" option.

C. This cannot be accomplished through the admin interface You must update the theme in code

D. Use a contributed module to inject CSS to hide the Druplicon and display your logo instead

ANSWER: A

Explanation:

Go to Appearance > Settings, select the active theme, and upload the company logo because Drupal's theme settings form includes built-in logo configuration. From this administrative page, a site builder can disable the setting to use the logo supplied by the theme, then select and upload a replacement image or specify the path to an existing image in the site's public files. Saving the theme settings makes Drupal render the replacement logo through the theme's normal logo variable and templates, without requiring code changes or an additional module.

Theme settings are configured per theme, so it is important to open the settings page for the theme currently used by the site rather than only changing a global-looking appearance setting. The uploaded file should be an appropriate web image and be accessible through Drupal's configured file system. This approach is the standard Drupal administrative workflow for replacing a theme-provided default logo. Drupal's user documentation describes configuring theme settings, including logo images, at Drupal User Guide: Themes. The relevant administrative route and theme configuration concepts are also documented in the Drupal ThemeSettingsForm API documentation.

QUESTION NO: 7

A Drupal site is configured for English and French. The site builder needs translated labels for a custom taxonomy vocabulary and wants each taxonomy term to have its own French name.

Which TWO Drupal multilingual capabilities are needed? Choose 2 answers

A. Enable Configuration Translation to translate the vocabulary label and description.

B. Enable Content Translation and configure the taxonomy vocabulary so its terms can be translated.

C. Enable Interface Translation to create French versions of each taxonomy term.

D. Enable browser language detection to translate vocabulary labels automatically.

E. Create a separate French vocabulary with duplicate terms and no translation relationship.

ANSWER: A B

Explanation:

The Configuration Translation module is used to translate configuration such as the label and description of a taxonomy vocabulary. Content Translation is used to enable translation of content entities, including taxonomy terms when the vocabulary is configured as translatable. Together, these capabilities allow the vocabulary's administrative configuration and the individual term names to be translated appropriately.

Interface Translation provides translations for Drupal interface strings supplied by modules and themes, but it does not create translated taxonomy terms. Language detection chooses a language for a request; it does not provide translated values. See the Drupal multilingual overview and the Drupal documentation on translating configuration.

QUESTION NO: 8

The marketing department has decided that a call-to-action button currently in the site footer should be moved to the sidebar for higher visibility The button is currently implemented in a custom block.

How should you make the requested changes?

A. From the block layout admin page, drag the existing block from the Footer region to the Sidebar region

B. Using the "Basic page" content type's "Manage display" interface, drag the existing block from the Footer region to the Sidebar region.

C. For each content type, use the Layout Builder module's 'manage layout" feature to reassign the block from the Footer region to the Sidebar region

D. From the "Appearance" admin page, reconfigure the site's theme to place the block in the Sidebar region instead of the Footer.

ANSWER: A

Explanation:

From the block layout admin page, drag the existing block from the Footer region to the Sidebar region is correct because custom blocks are placed into theme regions through Drupal's Block layout administration interface. A block's content and its placement are managed separately: the custom block remains the same reusable block, while its region assignment determines where the active theme renders it. An administrator can open *Appearance* > *Block layout*, locate the call-to-action block in the Footer region, and move it to an available Sidebar region using the region selector or the drag handle, then save the block layout. This change applies wherever that theme's sidebar region is rendered, subject to any visibility conditions configured for the block. No editing of the underlying custom block content is needed merely to change its location. Drupal's user guide describes theme regions as the areas where blocks can be placed and explains that the Block layout page is used to place and arrange blocks in those regions. See the Drupal User Guide: Concept—Regions in a Theme and Drupal User Guide: Placing a Block in a Region.

QUESTION NO: 9

The design team wants article teaser images to load at an appropriate size on phones, tablets, and desktop screens. Original images should remain available for other uses on the site.

What is the best Drupal core solution?

A. Create image styles for the required sizes, create a responsive image style, and use the Responsive image formatter in the Teaser view mode.

B. Upload a separate image field for phone, tablet, and desktop on every Article node.

C. Set the image field's maximum resolution to the largest desktop size.

D. Use CSS to resize the original image differently at each screen width.

ANSWER: A

Explanation:

Create image styles for the required image sizes, create a responsive image style that maps those styles to breakpoints, and select the Responsive image formatter for the image field in the Teaser view mode. Drupal can then provide browser-appropriate image derivatives while retaining the original uploaded image and allowing other display modes to use different styles.

The Responsive Image module works with image styles and breakpoint configuration to provide responsive image output. See the Drupal Responsive Image module documentation.

QUESTION NO: 10

The site has a number of older URLs that have been shared in newsletters and printed materials. A page has been moved from `/about-us` to `/about`, but visitors using the older URL should be sent automatically to the new page.

How can you configure this without custom code?

A. Install and configure the Redirect module, then create a redirect from "/about-us" to "/about".

B. Add "/about-us" as a second menu link to the page, then hide that menu link from navigation.

C. Create a URL alias of "/about-us" for the new page and remove the alias of "/about".

D. Add a custom block to the old page path containing a link to "/about".

ANSWER: A

Explanation:

Install and configure the Redirect module, then create a redirect from `/about-us` to `/about`. Redirect is the standard contributed Drupal solution for managing redirects from obsolete internal paths to current URLs. It can issue an appropriate permanent redirect, preserving access for visitors who use the old address and helping search engines recognize the preferred location.

Creating a new URL alias alone does not redirect an old path; it only defines an additional alias for a route. A menu link also does not control requests made directly to an old URL. The Redirect module provides a user interface for managing redirect records. See the Redirect module project page.