

DUMPS ARENA

Acquia Certified Drupal Site Builder Exam for
Drupal 10, 11

Acquia Drupal-Site-Builder

Version Demo

Total Demo Questions: 6

Total Premium Questions: 63

Buy Premium PDF

<https://dumpsarena.co>

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Content Modeling	12
Topic 2, Site Display	17
Topic 3, Site Building	16
Topic 4, Module and Theme Management	4
Topic 5, Security and Performance	13
Topic 6, Mix Questions	1
Total	63

QUESTION NO: 1

Your site is experiencing high page load times. When looking at recent log messages, you noticed a high frequency of "page not found" messages from a single IP address. On further investigation, the IP address was tracked to a suspicious crawler.

Which step will you take to improve site performance?

- A. Enable "Ban" module and configure it to ban the IP address found in the logs.
- B. Visit the Performance administration page, select "enable fast 404s".
- C. Visit the Search administration page, select redirect 404s to the search page.
- D. No action needed, Drupal automatically blocks any such suspicious visitors.

ANSWER: A

Explanation:

Enable "Ban" module and configure it to ban the IP address found in the logs. This directly addresses the identified source of the excessive requests: a suspicious crawler operating from one known IP address. Drupal's core Ban module lets an administrator add an IP address to the banned list through the administration interface. Once banned, requests from that address receive an access-denied response rather than continuing through the normal request handling needed to resolve routes, render pages, and generate repeated "page not found" log entries. Reducing these unnecessary requests lowers avoidable application work and log volume, helping preserve resources for legitimate visitors.

This is an appropriate operational response when investigation has established that the traffic is unwanted and originates from a specific IP address. The ban should be reviewed as part of routine site operations, particularly if the crawler's source addresses change or if an address is later found to belong to legitimate traffic. Drupal includes the Ban module in core; it must be enabled before IP addresses can be managed. See the [Drupal Ban module documentation](#) and the [Drupal API documentation for BanMiddleware](#).

QUESTION NO: 2

Users are complaining that listing pages with a large number of images are loading very slowly. On reviewing this you found that full-sized images are loaded while displaying thumbnails.

How will you ensure that smaller sized images are used as thumbnails?

- A. Enable the "Optimize images" option on the Performance administration page.
- B. Use an image style to resize the images.
- C. Use JavaScript to resize the images.
- D. Use CSS to resize the images.

ANSWER: B

Explanation:

Use an image style to resize the images. Drupal image styles create and deliver derivative versions of an original image rather than requiring the browser to download the original full-sized file. An image style can combine image effects such as scale, crop, resize, desaturate, or rotate; for thumbnails, a scale-and-crop or scale effect can produce a consistently small image suitable for listing pages. When the styled image is first requested, Drupal generates the derivative in the public files directory and can reuse that derivative for subsequent requests. This reduces transferred bytes and rendering work for pages that show many images, improving perceived and actual load performance.

A site builder creates the required style under *Configuration* → *Media* → *Image styles*, adds the appropriate effects and dimensions, and then selects that style in the image field's display formatter or in the relevant View field configuration. This ensures the markup references the styled image URL, not merely a visually constrained original image. Drupal's Image module documentation describes image styles as named configurations that apply effects to produce image derivatives, and

the Field UI documentation covers configuring how image fields are displayed. See [Drupal.org: Image styles](#) and [Drupal.org: Working with images](#).

QUESTION NO: 3

You have created a custom block "Festival Offer" and placed this block in the header section of your site. Your site has a "Premium customer" user role. You need to show this block only on the front page and only to users who have a "Premium customer" user role.

Which two visibility settings you need to make to the "Festival Offer" block to achieve these requirements?

- A. Select the user role "Premium customer" from the Roles tab.
- B. Add in the Pages tab and select "Show for the listed pages".
- C. Remove < front > path in the Pages tab & select "Hide for the listed pages".
- D. Select the user role "Authenticated user" from the Roles tab.

ANSWER: A B

Explanation:

Selecting the user role "Premium customer" from the Roles tab applies Drupal's role-based block visibility condition. The block is then available only when the current user has that role. This condition is evaluated together with other configured visibility conditions, so it can be combined with a page requirement rather than functioning as an alternative to it.

Adding <front> in the Pages tab and selecting "Show for the listed pages" limits the block to the site's front page. Drupal uses the special <front> token for the home page rather than requiring the actual path configured as the front page. With both settings configured, the Festival Offer block displays only when the request is for the front page and the viewing account has the Premium customer role. This satisfies both requirements through the standard block configuration interface, without custom code or a contributed visibility module. Drupal's user guide documents page and role visibility controls for blocks, including the <front> value for the front page: [Drupal User Guide: Placing a Block](#). The configurable visibility conditions are also implemented by the core block form: [Drupal BlockForm API reference](#).

QUESTION NO: 4

You just installed Drupal 10 with a standard installation profile.

When you visit the User roles list, which three user roles are already available by default?

- A. Site builder
- B. Authenticated user
- C. Publisher
- D. Content editor
- E. Administrator

ANSWER: B D E

Explanation:

A Drupal 10 Standard-profile installation supplies a ready-to-use baseline of user roles for normal site operation and editorial work. **Authenticated user** is the built-in role that applies automatically to every user who has signed in. It is fundamental to Drupal's permissions model: permissions assigned to this role are inherited by all logged-in accounts, unless access is further controlled by other configuration.

Content editor is included by the Standard installation profile to support common content-management workflows without requiring the site builder to first create an editorial role. Its installed configuration provides the starting point for users responsible for creating and maintaining site content.

Administrator is also provided by the Standard profile as the privileged site-management role. It is intended for accounts that administer the site's configuration, users, structure, and operational settings. Drupal's user documentation describes roles as permission groupings assigned to users, while the Standard profile's install configuration includes the content-editor and administrator role definitions. These three listed roles are therefore available immediately after a Standard-profile installation. See the [Drupal User Guide: users, roles, and permissions](#) and the [Drupal Standard profile install configuration](#).

QUESTION NO: 5

You manage a local restaurants guide website. You are creating a page listing all the restaurants registered on your site. You've been asked to make sure that each restaurant in the list includes an image, a title, and cuisine style. When the user clicks the title or picture, the user will be taken to a detail page showing a full description of the restaurant.

Which 2 options will ensure that the listing page only contains the fields noted, without removing the description field from the detail page?

- A.** Use the Views module to create a list of all restaurants, and make sure it displays each restaurant row as fields. Add the image, title, and cuisine fields to the view's field list.
- B.** Create a second content type called "Restaurant Overview," which only contains the image, title, and cuisine fields. Use the Views module to create a list of all of these Restaurant Overview nodes, but link their titles to their corresponding Restaurant nodes.
- C.** Use custom display settings for the "Teaser" view mode and configure it to display only the image, title, and cuisine fields. Use the Views module to create a list of all restaurants, displaying each row as content using the "Teaser" view mode.
- D.** Use the Views module to create a list of all restaurants. Configure the view to hide the description field by adding a custom CSS class to its wrapper.

ANSWER: A C

Explanation:

Using the Views module to create a list of all restaurants and display each row as fields is an appropriate solution because a View can select exactly which fields are rendered for a listing. Adding the image, title, and cuisine fields gives the listing its required content while leaving the restaurant body or description field unchanged on the node itself. The image and title fields can also be configured to link to the restaurant's content page, preserving access to the full detail page. This is a standard use of Views field-based displays; see the [Drupal Views documentation](#).

Using custom display settings for the "Teaser" view mode and configuring it to display only the image, title, and cuisine fields is also correct. Drupal separates stored field data from presentation settings through view modes. The Full view mode can continue to render the description for the restaurant detail page, while the Teaser view mode renders the concise set of fields needed in a listing. A View configured to display rows as content using the Teaser view mode then reuses that display configuration consistently anywhere teasers are needed. Drupal's [Manage display modes documentation](#) describes this presentation mechanism.

QUESTION NO: 6

You wish to display the "Recent content" block in the sidebar region on all the article pages only. The block should be hidden on all other pages.

How do you accomplish this?

- A.** Add the Recent content block to all pages, and hide it from non-Article pages using CSS.
- B.** Edit Article Content type and check the "show recent content" option.
- C.** Check the "show recent content" option in the site branding block, and set it to be visible only for Article Content type.

D. In the Block layout settings, add the Recent content block to the Sidebar section and set the block to be visible for the "Article" Content type.

ANSWER: D

Explanation:

In the Block layout settings, add the Recent content block to the Sidebar section and set the block to be visible for the "Article" Content type. Drupal's block placement interface lets a site builder select the theme region where a block is rendered and configure visibility conditions independently for that specific block placement. After placing the Recent content block in the sidebar region, open its configuration form and use the Content types visibility condition to select Article. Drupal then evaluates the route's displayed node and renders the block when that node is an Article, while keeping it out of the sidebar for pages that do not display an Article node. This is a configuration-based solution, so the block is not merely concealed after rendering; its display is governed by Drupal's normal block visibility system. The approach also keeps the behavior maintainable: a site builder can later adjust the assigned region, change the selected content types, or add other visibility conditions through the same block configuration screen. Drupal's user guide documents placing blocks through the Block layout page and configuring their visibility, including content-type conditions. See [Drupal User Guide: Placing a block in a region](#) and [Drupal User Guide: Configuring block visibility](#).