

Liferay Front End Certification Practice test (Sample)

Study Guide



Everything you need from our exam experts!

Copyright © 2025 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain from reliable sources accurate, complete, and timely information about this product.

SAMPLE

Questions

SAMPLE

- 1. What is the role of the `journal` folder in Liferay?**
 - A. It stores user-generated content securely.**
 - B. It includes web content such as articles and templates.**
 - C. It archives any outdated site templates.**
 - D. It manages third-party integrations.**
- 2. Which of the following elements can be added to the `_liferay-look-and-feel.xml` file to modify the theme?**
 - A. Custom CSS**
 - B. Application Decorators**
 - C. Custom JavaScript**
 - D. HTML Metadata**
- 3. Which file types are NOT commonly used in Liferay development?**
 - A. HTML files**
 - B. Cascading Style Sheets (.css)**
 - C. Executable files (.exe)**
 - D. JavaScript files (.js)**
- 4. In terms of technical capabilities, what does Lexicon Design primarily offer?**
 - A. A set of visual styles only**
 - B. A framework for developing rules and patterns**
 - C. A coding language for new developers**
 - D. A library for external integration**
- 5. What parameter specifies how long a request can take before it times out in an SPA?**
 - A. Request Timeout Time**
 - B. Cache Expiration Time**
 - C. User Notification Time**
 - D. Navigation Exception Selectors**

- 6. What class is required for all column containers in Bootstrap?**
- A. portlet-column**
 - B. grid-column**
 - C. layout-column**
 - D. column-container**
- 7. How do you define a layout template in the liferay-look-and-feel.xml?**
- A. By using the layout-template tag**
 - B. By setting up a CSS class**
 - C. By including a thumbnail path**
 - D. By adding custom JavaScript functionality**
- 8. Which syntax is used to set an AUI column to replicate a Bootstrap col-6?**
- A. <aui:col cssClass="col-6">**
 - B. <aui:column width="50">**
 - C. <aui:column cssClass="half-width">**
 - D. <aui:col width="60">**
- 9. What does the lexicon experience language provide?**
- A. A common design framework for Liferay**
 - B. A coding language for Liferay themes**
 - C. Guidelines for content management**
 - D. A plugin for site administration**
- 10. What is defined in the assets.json file?**
- A. Security credentials for asset management.**
 - B. The overall layout of the site.**
 - C. The metadata of the imported assets.**
 - D. The configuration settings for themelets.**

Answers

SAMPLE

- 1. B**
- 2. B**
- 3. C**
- 4. B**
- 5. A**
- 6. A**
- 7. A**
- 8. A**
- 9. A**
- 10. C**

SAMPLE

Explanations

SAMPLE

1. What is the role of the `journal` folder in Liferay?

- A. It stores user-generated content securely.
- B. It includes web content such as articles and templates.**
- C. It archives any outdated site templates.
- D. It manages third-party integrations.

The `journal` folder in Liferay plays a vital role in managing web content. This includes storing various types of content such as articles, templates, and other web content structures. The organization of these elements within the `journal` folder allows for efficient handling and retrieval, which is crucial for displaying dynamic content on a Liferay site. Web content in Liferay is primarily managed through the Journal Content API, which provides the necessary functionalities for creating, updating, and displaying content. By having a dedicated location for articles and templates, Liferay ensures that developers and content managers have a streamlined approach to working with web content, facilitating consistent content delivery and management across different parts of the website. The other choices do not accurately represent the primary function of the `journal` folder. For example, while user-generated content may be stored, the `journal` folder specifically focuses on structured web content rather than general user-generated content. Archiving outdated templates and managing third-party integrations are more related to other functionalities within Liferay and do not fall under the purview of what the `journal` folder is designed to support.

2. Which of the following elements can be added to the `_liferay-look-and-feel.xml` file to modify the theme?

- A. Custom CSS
- B. Application Decorators**
- C. Custom JavaScript
- D. HTML Metadata

The choice of application decorators is accurate because the `_liferay-look-and-feel.xml` file is specifically designed to customize certain aspects of a Liferay theme, including the structure and layout adjustments that can be achieved through decorators. Application decorators allow developers to wrap the entire portal's output with a specific HTML structure, which can be vital for implementing distinct visual presentations or overriding default output with custom elements. This capability to extend the functionality by changing the layout and styling of sections of the site is pivotal in theme development within Liferay. Other choices like custom CSS, custom JavaScript, and HTML metadata are indeed important elements in theme creation and manipulation within Liferay, but they are not specifically configured in the `_liferay-look-and-feel.xml` file. Custom CSS and JavaScript are generally included in the theme's stylesheet or script files instead and can be linked from other configuration files or folders. Meanwhile, HTML metadata is usually specified in the head section of HTML files rather than centralized in the `_liferay-look-and-feel.xml`. Hence, the focus on application decorators as the correct answer highlights the particular purpose and functionality bestowed by the `_liferay-look-and-feel.xml` configuration in customizing themes.

3. Which file types are NOT commonly used in Liferay development?

- A. HTML files
- B. Cascading Style Sheets (.css)
- C. Executable files (.exe)**
- D. JavaScript files (.js)

In Liferay development, the focus is primarily on web technologies that are essential for building user interfaces and enhancing user experiences. HTML files, CSS files, and JavaScript files are fundamental components of web development. They are commonly used within Liferay to create pages, style content, and add interactivity, which are crucial for developing themes and applications within the platform. Executable files, such as those with the .exe extension, are not utilized in Liferay development. These files are typically associated with Windows applications and are not relevant to the creation of web applications or the deployment of features in a portal environment like Liferay. The Liferay framework leverages web standards, and executable files do not fit into this context, as they are intended for execution in a local operating system rather than serving content over the web. Understanding the roles of different file types in web development is key. Liferay extensively employs HTML, CSS, and JavaScript, while executable files do not serve a purpose in the development and deployment of Liferay applications.

4. In terms of technical capabilities, what does Lexicon Design primarily offer?

- A. A set of visual styles only
- B. A framework for developing rules and patterns**
- C. A coding language for new developers
- D. A library for external integration

Lexicon Design primarily offers a framework for developing rules and patterns, which plays a crucial role in ensuring consistency and coherence in user interface design across various applications. This framework encompasses guidelines for typography, spacing, colors, and components, all aimed at creating a unified experience for users. By establishing these design principles, Lexicon helps developers and designers understand how to implement visual styles effectively, ensuring that their work aligns with the overall Liferay ecosystem. The framework is particularly beneficial as it allows for the systematic application of design standards across different projects, facilitating collaboration among teams and enhancing usability. This approach goes beyond merely providing visual styles or aesthetics; it revolves around the consistent application of design methodologies that enhance the user experience and interface functionality. The other options, while related to design and coding, do not capture the essence of what Lexicon Design contributes. For instance, a set of visual styles alone would lack the comprehensive structure needed for cohesive design across applications. A coding language for new developers would not encompass the design-focused nature of Lexicon, and a library for external integration would detract from the core focus on user interface design principles that Lexicon is intended to provide.

5. What parameter specifies how long a request can take before it times out in an SPA?

- A. Request Timeout Time**
- B. Cache Expiration Time**
- C. User Notification Time**
- D. Navigation Exception Selectors**

The parameter that specifies how long a request can take before it times out in a Single Page Application (SPA) is indeed related to the concept of a request timeout. A request timeout is essential in web applications to avoid hanging indefinitely if a server does not respond, which would lead to a poor user experience. When defining timeouts, developers set a specific duration for which the application will wait for a response before concluding that the request has failed, ensuring that users receive timely feedback. In the context of SPAs, this timeout setting enables developers to manage how long the application should attempt to retrieve information from the backend before taking alternative actions, such as displaying an error message or retrying the request. Properly configured request timeout parameters contribute to the reliability and usability of web applications by preventing unresponsive behavior. The other options are unrelated to request timing. Cache expiration time deals with how long data should be stored in a cache before being considered stale. User notification time refers to the duration for displaying notifications to users. Navigation exception selectors would pertain to handling errors or exceptions in navigation processes, rather than defining the timeout for requests. Thus, the specified option reflects a crucial aspect of request management in SPAs.

6. What class is required for all column containers in Bootstrap?

- A. portlet-column**
- B. grid-column**
- C. layout-column**
- D. column-container**

The requirement for all column containers in Bootstrap is to use the class 'portlet-column'. This class is integral to the Bootstrap grid system, which is designed to create responsive layouts that adapt to various screen sizes. When building an interface using Bootstrap, using the appropriate classes allows developers to define how columns behave within a row. The 'portlet-column' class specifically indicates that an element is part of a column layout, enabling better alignment and spacing in conjunction with other Bootstrap utilities. The other options do not correspond with Bootstrap's established class system for creating columns. While Bootstrap has a variety of classes for layouts, 'portlet-column' stands out as the key class to denote a column container specifically within Bootstrap's framework. Understanding this concept is essential for building well-structured and responsive web applications using Bootstrap.

7. How do you define a layout template in the liferay-look-and-feel.xml?

- A. By using the layout-template tag**
- B. By setting up a CSS class**
- C. By including a thumbnail path**
- D. By adding custom JavaScript functionality**

Defining a layout template in the liferay-look-and-feel.xml file is done by using the layout-template tag. This tag specifically allows developers to create tailored layouts for their portlets and pages within the Liferay framework. It provides the framework with the necessary structure and semantics to recognize what constitutes a layout for a given theme. When the layout-template tag is employed, it can encompass various attributes, such as specifying the template's name and design features, which can then be referenced when configuring pages in the Liferay Portal. This provides flexibility in theme design, enabling developers to implement diverse layouts that enhance user experience. Other methods like setting up a CSS class, including a thumbnail path, or adding custom JavaScript functionality, do not specifically define layout templates. While these elements can be vital in theme development, they serve different purposes—styling, visual representation, or functionality enhancement—rather than the foundational layout design established by the layout-template tag. Thus, the correct approach to defining a layout template involves directly using the layout-template tag in the configuration file.

8. Which syntax is used to set an AUI column to replicate a Bootstrap col-6?

- A. <alui:col cssClass="col-6">**
- B. <alui:column width="50">**
- C. <alui:column cssClass="half-width">**
- D. <alui:col width="60">**

The syntax `<alui:col cssClass="col-6">` is correct because it directly utilizes the Bootstrap grid system, which employs specific class names to define the width of columns in a responsive layout. In Bootstrap, the class "col-6" divides the available horizontal space into 12 equal parts, with "col-6" taking up 6 of those parts, representing a half-width column. By using the `cssClass` attribute in the AUI tag to specify "col-6", you effectively instruct the system to apply the corresponding Bootstrap styles to that column, ensuring proper alignment and responsiveness in the layout. Other options, while they reference column settings, do not effectively align with Bootstrap's grid system. The use of width percentages or non-standard class names fails to directly replicate the intended Bootstrap behavior or styling conventions, which is based on predefined classes. Thus, opting for the use of predefined Bootstrap classes through the `cssClass` attribute is the most straightforward and accurate method for achieving the desired layout in AUI.

9. What does the lexicon experience language provide?

- A. A common design framework for Liferay**
- B. A coding language for Liferay themes**
- C. Guidelines for content management**
- D. A plugin for site administration**

The lexicon experience language in Liferay provides a common design framework that helps ensure consistency and cohesion across different applications and interfaces within the platform. This framework includes a set of design principles, patterns, and components that all Liferay themes and applications can adhere to, promoting a unified user experience. It encompasses aspects such as typography, color, spacing, and modular components that developers and designers can leverage to create interfaces that are not only visually appealing but also functionally integrated. By using this common design framework, organizations can enhance usability and accessibility for their users, while also streamlining the development process across their various Liferay projects. This feature is fundamental to maintaining a high standard for design within the Liferay ecosystem.

10. What is defined in the assets.json file?

- A. Security credentials for asset management.**
- B. The overall layout of the site.**
- C. The metadata of the imported assets.**
- D. The configuration settings for themelets.**

The assets.json file is a crucial component in the context of Liferay that specifically holds the metadata for imported assets. This metadata includes valuable information about the assets, such as their types, descriptions, and other descriptive attributes that are essential for the correct identification and management of these assets within the Liferay platform. By organizing and storing this information in a JSON format, Liferay allows for effective asset management, ensuring that all necessary details pertaining to various assets are easily accessible and modifiable. This structured approach to asset data supports better organization and facilitates interactions with the assets, whether for presentation, manipulation, or future import/export activities. In contrast, the other choices refer to different aspects of Liferay's functionality and configurations. Security credentials relate to user authentication and access management, overall site layout pertains to page design and structure, and themelets are related to themes and visual representation but do not encompass asset-specific metadata.