



Free Questions for AD0-E123 by [braindumpscollection](#)

Shared by [Garza](#) on 24-05-2024

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

A developer is using a Sling Model to apply some business logic to the values stored in the JCR before outputting them to the page, and all injected fields are optional for this model.

What is the proper way to apply this behavior on the class level?

Options:

- A- 'default InjectionStrategy = Default InjectionStrategy.OPTIONAL'
- B- 'requiredInjectionStrategy = FALSE'
- C- 'injectionStrategy = InjectionStrategy.OPTIONAL'

Answer:

A

Explanation:

In Sling Models, the `@DefaultInjectionStrategy` annotation is used to specify the strategy for injecting fields into the model. When setting it to `DefaultInjectionStrategy.OPTIONAL`, it indicates that all injected fields in the Sling Model are optional and the model will not fail to

instantiate if a field cannot be injected. This is a class-level annotation that affects the default injection behavior of all fields within the Sling Model.

Option B is incorrect because `requiredInjectionStrategy` is not a valid annotation property in Sling Models.

Option C is incorrect because `injectionStrategy` is not specified at the class level; it is used on the field level with the `@Inject` annotation.

Question 2

Question Type: MultipleChoice

A developer must use `print-314%` in a webpage.

Assuming `piValue = -3.14`

Which HTL approach would be used?

Options:

A- `${'-#.##%' @ format=piValue}`

B- `${'#%' @ formal=piValue. percentages 100}`

C- `#{#%' @ format=piValue}`

Answer:

A

Explanation:

In HTL, the format option in the data-sly-use or data-sly-test attribute can be used to format values according to a pattern. The pattern `'-###%'` indicates that the number should be formatted as a percentage with up to two decimal places. The `piValue` variable would be the number to format, and in this example, it would convert `-3.14` to `-314%`. The `@` symbol is used to denote the use of an option with the variable in HTL.

Options B and C are incorrect because they do not use the correct syntax for the format option in HTL. The use of `@` is to specify options and not part of the variable or pattern itself.

Question 3

Question Type: MultipleChoice

A developer has created a transient workflow, but a specific step defeats the purpose of making the workflow transient. The step creates a sling job to proceed further and generates error messages in log files.

Which step must be avoided to eliminate this problem?

Options:

A- Container Step

B- Goto Step

C- Process Step

Answer:

C

Explanation:

In Adobe Experience Manager (AEM), a transient workflow is designed for short-lived workflows that do not persist their state information in the repository, which means they are faster and do not leave a footprint in the repository. However, if a Process Step in a transient workflow creates a sling job, it defeats the purpose of the transient workflow because sling jobs are persisted in the repository and could generate error messages in the log files if they are not executed immediately or if there are issues with the job queue.

Therefore, to maintain the transient nature of the workflow, Process Steps that create sling jobs should be avoided.

The Container Step (Option A) is used to group other workflow steps and does not inherently create sling jobs.

The Goto Step (Option B) is used for branching logic and also does not create sling jobs.

Question 4

Question Type: MultipleChoice

A developer wants to replace the element name of the host element with titleLevel property of the model in HTL.

Which action would be taken to complete this task?

Options:

A-

C- `<h1 data-sly-element='${titleLevel}' @context='unsafe'></h1>`

Answer:

C

Explanation:

In HTL (HTML Template Language), data-sly-element is used to dynamically set the name of a DOM element. The provided option C correctly demonstrates the use of data-sly-element to replace the element name with the value of the titleLevel property of the model. The context 'unsafe' is specified because changing the element name can be a security risk if the value is not properly sanitized since it could lead to script injection vulnerabilities if used with user-provided data.

Options A and B are incorrect because they don't use the data-sly-element attribute, which is necessary to dynamically change the tag name of the element in HTL.

Question 5

Question Type: MultipleChoice

Multiple Experience fragments have been created as a part of the ongoing AEM Project. These Experience fragments can be exported to Adobe Target using the "Export to Adobe Target" configuration and by providing Cloud Configuration for Adobe Target.

What is the default format used to complete this export?

Options:

A- HTML

B- XML

C- JSON

Answer:

A

Explanation:

Experience Fragments in AEM can be exported to Adobe Target, and the default format for this export is HTML. When Experience Fragments are exported to Adobe Target, they are typically used for personalization and A/B testing. HTML is the format that is readily usable within Adobe Target to create variations of a page or a component for these testing purposes.

XML (Option B) and JSON (Option C) are not the default formats for exporting Experience Fragments to Adobe Target. While they can represent structured data, for the purposes of visual personalization in Adobe Target, the HTML format is used because it can directly represent the markup required for the content to be rendered in a browser.

Question 6

Question Type: MultipleChoice

A developer needs to produce various content fragments depending on some dynamic names and decides to use JAVA code.

Which code snippet should be used to create a new content fragment?

Options:

A- resource.createFragment()

B- resource, adapt To(Fragment class)

C- FragmentTemplate.createFragment()

Answer:

C

Explanation:

To create new content fragments programmatically in AEM, a developer would typically use an API provided by AEM for content fragment operations. Among the options given, `FragmentTemplate.createFragment()` seems to be the most fitting API method call, as it implies the use of a template to create a fragment. In practice, the Content Fragment Java API provides methods to create and manage content fragments in AEM by interacting with the underlying JCR (Java Content Repository). However, without specific API documentation, I'm relying on provided patterns that align with typical AEM Java API usage.

Question 7

Question Type: MultipleChoice

An AEM sites developer is working on the development of a 'Hello World' component and wants to facilitate the mapping of data from the JCR to Java variables.

Which feature should the developer use to provide this facilitation?

Options:

- A- HTL and Dialogs
- B- Client-Side Libraries
- C- Sling Models

Answer:

C

Explanation:

Sling Models is an AEM feature that allows developers to map JCR data to Java objects. It is a part of the Sling framework, which AEM is built on top of. Sling Models facilitate the decoupling of content from the rendering layer, making it easier to work with data in AEM components. By defining simple POJOs (Plain Old Java Objects) with annotations, developers can map properties from the JCR (Java Content Repository) to fields in a Java class, enabling a straightforward and type-safe way to handle the content. This is especially useful when creating components such as 'Hello World', where dynamic data needs to be fetched from the JCR and rendered on the page.

HTL and Dialogs (Option A) are used for rendering components and creating editable fields in the component dialogs, respectively, but do not facilitate data mapping to Java variables.

Client-Side Libraries (Option B) are used to manage client-side resources like JavaScript and CSS but are not involved in JCR to Java variable mapping.

Question 8

Question Type: MultipleChoice

A developer needs to use the AEM Core Components. What is the recommended way to complete this task?

Options:

- A- By referencing them directly from the content
- B- By changing their component group to use them directly in editor
- C- By using proxy components

Answer:

C

Explanation:

The recommended approach to use AEM Core Components is by creating proxy components. Proxy components are essentially wrapper components within your own project that reference the core components. This practice allows developers to inherit functionality from the core components while being able to override and extend it as necessary. By using proxy components, developers can ensure that any custom styles and configurations are preserved even when the core components are updated in the future.

Question 9

Question Type: MultipleChoice

A developer needs to remove all the selectors from the linkValue.

Assuming linkValue = 'path/page.woo.too.html', which two HTL approaches would be used to complete this task? (Choose two.)

Options:

A- `${linkValue@ removeSelectors-foo, bar, woo}`

B- `${ linkValue @ temoveSeleclots=[foo', bar']}`

C- `${linkValue @ selectors='}'`

D- `${ linkValue @ selectors}`

Answer:

C, D

Explanation:

In HTL (HTML Template Language), the syntax to manipulate the URI path does not provide direct methods for removing selectors. However, within Sling and HTL, the Link Handler can be extended to support such operations. For out-of-the-box HTL, the proper approach to manipulate the link would be done in the Java layer (e.g., a Sling Model or a Use-Class in HTL), and then passed to the template for rendering. Nevertheless, if such a functionality were to be implemented within HTL, it would require custom code and not use the indicated syntax options. It is important to note that options A and B with 'removeSelectors' are not standard HTL syntax.

Question 10

Question Type: MultipleChoice

A developer has to allow third-party applications to access the content of the Experience Fragment directly, using only the URL. It should be available through the browser.

Which selector would be used for this purpose?

Options:

A- text

B- noCss

C- plain

Answer:

A

Explanation:

When it comes to rendering the content of an Experience Fragment for third-party applications to access directly via a URL, and in a browser-friendly format, the text selector is used. This selector causes AEM to render the Experience Fragment as plain text, which is typically what a third-party application would require if it needs to consume the content directly without any associated styling or scripts.

Option B, noCss, is not a standard selector for this purpose.

Option C, plain, is not the correct selector for rendering Experience Fragment content in AEM. The plain selector might suggest a text/plain MIME type, but in AEM, the correct selector for retrieving the text content of an Experience Fragment is text.

Question 11

Question Type: MultipleChoice

Review the following HTL code snippet:

```
use.testComponent=${com.adobe.core.models.TestComponent}@  
testParam='testValue'}>
```

Which two approaches correctly read the value of testParam in Sling model? (Choose two.)

Options:

- A- @Inject @Source('request-attributes') @Named('testParam')
- B- @RequestAttribute(name = 'testParam')
- C- @Inject(name = 'testParam')
- D- @Inject @Prop('testParam')

Answer:

B, C

Explanation:

In Sling Models, annotations are used to map request parameters, attributes, and properties to fields in Java classes. When an HTL script passes parameters to a Sling Model, the Sling Model can use annotations to access these parameters.

Option B uses the `@RequestAttribute` annotation which allows the Sling Model to retrieve an attribute that has been set on the Sling HTTP request object. This is a valid approach to read the 'testParam' if it has been set as a request attribute.

Option C uses the `@Inject` annotation with the name attribute specifying the name of the parameter to inject. This is another valid approach to read the 'testParam' from the request.

Option A is incorrect because `@Named` is not the correct annotation for retrieving request attributes or parameters.

Option D is incorrect because there is no `@Prop` annotation in Sling Models; it's likely confused with the `@Property` annotation which is not the correct one for this context either.

To Get Premium Files for AD0-E123 Visit

<https://www.p2pexams.com/products/ad0-e123>

For More Free Questions Visit

<https://www.p2pexams.com/adobe/pdf/ad0-e123>

