Get Free VMware 2V0-72.22 Dumps PDF Questions

Why risk failure? Download updated VMware Professional Develop VMware Spring exam PDF questions today. Practice with real 2V0-72.22 dumps and verified answers designed to help you ace your certification quickly using PrepBolt 2V0-72.22 exam pdf questions and answers.

Thank you for Downloading 2V0-72.22 exam PDF Demo

https://prepbolt.com/2V0-72.22.html

QUESTIONS & ANSWERS

DEMO VERSION

(LIMITED CONTENT)

Question 1

Question Type: MultipleChoice

Which two statements are true concerning the BeanPostProcessor Extension point? (Choose two.)

Options:

A- BeanPostProcessors are called before the dependencies have been injected.
B- Custom BeanPostProcessors can be implemented for Spring applications.
C- BeanPostProcessors are called before the BeanFactoryPostProcessors.
D- BeanPostProcessors are called during the initialization phase of a bean life cycle.
E- BeanPostProcessors cannot be ordered in a Spring Boot application.

Answer:

B, D

Explanation:

B . Custom BeanPostProcessors can be implemented for Spring applications.

This is true because the BeanPostProcessor interface defines callback methods that you can implement to provide your own (or override the container's default) instantiation logic, dependency resolution logic, and so forth1.You can configure multiple BeanPostProcessor instances, and you can control the order in which these BeanPostProcessor instances run by setting the order property1. For example, inthis tutorial, a custom BeanPostProcessor is implemented to integrate Guava's EventBus with Spring beans.

D . BeanPostProcessors are called during the initialization phase of a bean life cycle.

This is true because the BeanPostProcessor interface consists of exactly two callback methods: postProcessBeforeInitialization and postProcessAfterInitialization.When such a class is registered as a post-processor with the container, for each bean instance that is created by the container, the post-processor gets a callback from the container both before container initialization methods (such as InitializingBean.afterPropertiesSet() or any declared init method) are called, and after any bean initialization callbacks1.

Question 2

Question Type: MultipleChoice

Which is the correct approach to register for a bean destruction callback?

Options:

A- Annotate the callback method with @PostDestroy.
B- Annotate the callback method with @PreDestroy.
C- Add the @Lazy annotation to the bean configuration.
D- Configure the bean instance to use prototype scope.

Answer:

B

Question 3

Question Type: MultipleChoice

What's the password storage format when using the DelegatingPasswordEncoder?

Options:

A- encodedPassword{id}, where {id} is an identifier used to look up which PasswordEncoder should be used.
B- {id}encodedPassword, where {id} is an identifier used to look up which PasswordEncoder should be used.
C- {timestamp}encodedPassword, where {timestamp} is the time when the password was encoded.
D- {id}{salt}encodedPassword, where {id} is an identifier used to look up which PasswordEncoder should be used and {salt} a randomly generated salt.

Answer:

B

Explanation:

https://docs.spring.io/spring-security/site/docs/current/api/org/springframework/security/crypto/password/DelegatingPasswordEncoder.html

Question 4

Question Type: MultipleChoice

Refer to the exhibit.

Which two statements are correct regarding the HelloAutoConfig auto-configuration class when it is specified in the META-INF/spring.factories file? (Choose two.)

Options:

A- A HelloService bean will be created from the helloService() method even if the HelloService.class is not in the classpath.
B- A HelloService bean will be created from the helloService() method only when there is no other HelloService bean in the ApplicationContext.
C- This auto-configuration class is used only when the HelloService.class is not on the classpath.
D- This auto-configuration class is used only when the HelloService.class is on the classpath.
E- A HelloService bean will be created from the helloService() method and will replace existing a HelloService bean in the ApplicationContext.

Answer:

B, D

Explanation:

B . A HelloService bean will be created from the helloService() method only when there is no other HelloService bean in the ApplicationContext.

This is true because the @ConditionalOnMissingBean annotation on the helloService() method indicates that this method will only be invoked to create a HelloService bean if there is no existing HelloService bean in the application context. This is a common way to provide a default bean that can be overridden by the user's own bean definition.

D . This auto-configuration class is used only when the HelloService.class is on the classpath.

This is true because the @ConditionalOnClass annotation on the HelloAutoConfig class indicates that this class will only be loaded and processed by Spring Boot if the HelloService.class is present on the classpath. This is a common way to enable auto-configuration for a specific library or feature only when it is available.

Question 5

Question Type: MultipleChoice

Which two statements are true regarding Spring Security? (Choose two.)

Options:

A- Access control can be configured at the method level.
B- A special Java Authentication and Authorization Service (JAAS) policy file needs to be configured.
C- Authentication data can be accessed using a variety of different mechanisms, including databases and LDAP.
D- In the authorization configuration, the usage of permitAll () allows bypassing Spring security completely.
E- It provides a strict implementation of the Java EE Security specification.

Answer:

A, C

Explanation:

Spring Security is a framework that provides comprehensive security services for Java applications, such as authentication, authorization, encryption, session management, and more. One of its features is method security, which allows applying access control rules at the method level using annotations or XML configuration. Another feature is authentication, which is the process of verifying the identity of a user or a system. Spring Security supports various authentication mechanisms, such as username and password, tokens, certificates, etc., and can access authentication data from different sources, such as databases, LDAP directories, in-memory stores, etc.

Question 6

Question Type: MultipleChoice

Which three types can be used as @Controller method arguments? (Choose three.)

Options:

A- Locale
B- Principal
C- Language
D- Session
E- Request
F- HttpSession

Answer:

A, B, F

Explanation:

A . Locale

This is true because the Locale argument can be used to resolve the current locale that the client is using, based on the Accept-Language header or a cookie1.

B . Principal

This is true because the Principal argument can be used to access the currently authenticated user, if any1.

F . HttpSession

This is true because the HttpSession argument can be used to access the current session, if any1. Note that this argument is not required and will be null if a session does not exist.

Question 7

Question Type: MultipleChoice

Which two statements are correct when @SpringBootApplication is annotated on a class? (Choose two.)

Options:

A- It causes Spring Boot to enable auto-configuration by default.
B- Component scanning will start from the package of the class.
C- All other annotations on the class will be ignored.
D- Methods in the class annotated with @Bean will be ignored.
E- A separate ApplicationContext will be created for each class annotated with @SpringBootApplication.

Answer:

A, B

Explanation:

A . It causes Spring Boot to enable auto-configuration by default.

This is true because the @SpringBootApplication annotation includes the @EnableAutoConfiguration annotation, which enables Spring Boot's auto-configuration mechanism1.Auto-configuration attempts to automatically configure your Spring application based on the jar dependencies that you have added2.

B . Component scanning will start from the package of the class.

This is true because the @SpringBootApplication annotation includes the @ComponentScan annotation, which enables component scanning on the package where the application is located1.Component scanning allows Spring to automatically detect and register beans annotated with @Component, @Controller, @Service, @Repository, etc3.

Question 8

Question Type: MultipleChoice

Refer to the exhibit.

Which two statements are correct regarding auto-configuration of DataSource and JdbcTemplate beans given a Spring Boot application with only these two dependencies? (Choose two.)

Options:

A- A DataSource bean will not be auto-configured.
B- A JdbcTemplate bean will not be auto-configured.
C- A JdbcTemplate bean will be auto-configured.
D- A DataSource bean will be auto-configured only if a JdbcTemplate bean is explicitly defined.
E- A DataSource bean will be auto-configured.

Answer:

C, E

Thank You for trying 2V0-72.22 PDF Demo

To try our 2V0-72.22 practice exam software visit link below

https://prepbolt.com/2V0-72.22.html

Start Your 2V0-72.22 Preparation

Use Coupon “SAVE50” for extra 50% discount on the purchase of Practice Test Software. Test your 2V0-72.22 preparation with actual exam questions.