

Quiz Questions

(1) What is default Spring Boot Server?

← Classroom

Introduction To Spring Boot
What is default Spring Boot Server? ▾

?

V

Problem Submissions Doubts

✔ What is default Spring boot server?

Easy • Score 20/20

Send feedback

Problem statement

You are a software developer tasked with developing a web application using the Spring boot web framework. Your project manager has asked him to choose an appropriate application server. Which is the default server used with the Spring web framework?

Options: Pick one correct answer from below

☒ Apache Tomcat

☐ NGINX

☐ Microsoft IIS

☐ Lighttpd

Solution description

Apache Tomcat: Open-source web server and servlet container commonly used as the default server for Spring web applications.
NGINX: High-performance web server widely used as a reverse proxy and load balancer.
Microsoft IIS: Web server designed for Windows-based environments.
Lighttpd: Lightweight and high-performance web server widely used for serving static content.

(2) Difference between Spring Boot and Spring

← Classroom

Introduction To Spring Boot
Difference between Spring Boot and Spring ▾

?

V

Problem Submissions Doubts

✔ Difference between Spring Boot and Spring

Easy • Score 20/20

Send feedback

Problem statement

How does Spring Boot differ from Spring applications?

Options: One or more answers may be correct

☒ Spring Boot provides a preconfigured environment for developing Spring applications.

☒ Spring Boot does not require any XML configuration files.

☐ There is no difference between Spring and Spring Boot.

☐ Spring Boot only supports microservices architecture.

Solution description

Spring Boot does not require XML configuration files. Spring Boot provides auto-configuration that simplifies the configuration of Spring applications. It automatically configures the application based on the dependencies it finds on the classpath, reducing developers' need to configure the application manually.

(3) Introduction to access Application Context

← Classroom

Introduction To Spring Boot
Instruction to access ApplicationContext ▾

?

V

Problem Submissions Doubts

✔ Instruction to access ApplicationContext

Easy • Score 20/20

Send feedback

Problem statement

Which instruction is used to access ApplicationContext in the IOC containers?

Options: Pick one correct answer from below

☒ ApplicationContext context = new ClassPathXmlApplicationContext("application.xml");

☐ ApplicationContext new context = ClassPathXmlApplicationContext("application.xml");

☐ ApplicationContext context = new ClassPathXmlApplicationContext(application.xml);

☐ ApplicationContext context = new ClassPathXmlApplicationContext("application");

Solution description

ClassPathXmlApplicationContext is an implementation of ApplicationContext, which is a central interface to provide configuration for an application. 'new ClassPathXmlApplicationContext("application.xml")' creates a new instance of the ApplicationContext, which reads the "application.xml" configuration file and initialises the beans defined in it. This instance can then be used to access the beans and their properties that are defined in the configuration file.
The incorrect options are:
• ApplicationContext new context = ClassPathXmlApplicationContext("application.xml");
(Syntax error due to wrong placement of "new" keyword)
• ApplicationContext context = new ClassPathXmlApplicationContext(application.xml);
(Syntax error due to missing quotation marks around "application.xml")
• ApplicationContext context = new ClassPathXmlApplicationContext("application");
(error in finding the file because it is missing the ".xml" extension)
The correct syntax to create the instance is
ApplicationContext context = new ClassPathXmlApplicationContext("application.xml");

(4) IOC Implementation Steps

← Classroom

Introduction To Spring Boot
IOC Implementation Steps

80%

− +

Reset

Problem Submissions Doubts

Easy • Score: 20/20

Send feedback

Problem statement
You are given a Vehicle Interface, and its implementations Car and Truck as follows:
Vehicle Interface:

```
interface Vehicle {  
    void name();  
}
```

Car Class:

```
class Car implements Vehicle {  
    void name() {  
        System.out.println("This is a car");  
    }  
}
```

Truck Class:

```
class Truck implements Vehicle {  
    void name() {  
        System.out.println("This is a truck");  
    }  
}
```

Options: Pick one correct answer from below

☐ A)

☐ B)

☐ C)

☒ D)

Solution description

IOC allows the decoupling of objects, so there is no need to create an object of Car explicitly. We can make use of the object returned from ClassPathXmlApplicationContext.

A)Configure spring beans in application.xml

```
<bean id="car" class="com.codingninjas.example.Car"/>  
<bean id="truck" class="com.codingninjas.example.Truck"/>
```

C)Retrieve Bean from Container

```
Vehicle car = context.getBean("car", Vehicle.class);
```

B)Create a Spring Container

```
ClassPathXmlApplicationContext context = new  
ClassPathXmlApplicationContext("application.xml");
```

D)Create object from Class

```
Vehicle car = new Car();
```

(5) XML Tag

← Classroom

Introduction To Spring Boot
XML Tag

Problem Submissions Doubts

✓ XML Tag

Easy • Score 20/20

Problem statement

What is the error in the XML code shown above?

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="Car"
class="com.example.Car">
</bean>

<bean id="Bus"
class=com.example.Bus>
</bean>

<bean id="Bicycle"
class="com.example.Bicycle">
</bean>

</beans>
```

Options: Pick one correct answer from below

☐ The xsi namespace declaration is missing from the beans element.

☒ The class in the Bus bean is not declared in string format.

☐ The Car bean definition is missing class definition.

☐ The Bicycle bean definition is incorrect

Solution description

The class in the bean definition should be enclosed in "" mark i.e in string format.

(6) IOC Exception – Intro to Spring Boot

← Classroom

Introduction To Spring Boot
IOC Exception

Problem Submissions Doubts

✓ IOException - Introduction to Spring Boot

Easy • Score 20/20

Problem statement

The main method below gives an exception at runtime. "IOException parsing XML document from class path resource". What could be the reason for this error?

```
// MainApp.java file-
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {
    public static void main(String[] args) {
        ClassPathXmlApplicationContext context = new
        ClassPathXmlApplicationContext("ApplicationContext.xml");

        Car myCar = (Car) context.getBean("car");
        myCar.drive();
    }
}
```

Options: Pick one correct answer from below

☐ The bean is not defined correctly in the ApplicationContext.xml.

☐ ClassPathXmlApplicationContext is not the correct Class to get context from the XML configuration file.


☒ The application cannot find the ApplicationContext.xml file.

☐ None of the above.

Solution description

This type of exception usually happens when something in the application points to an XML resource that doesn't exist or is not placed where it should be.

(7) Factory Design Pattern

 Classroom

Introduction To Spring Boot
Factory Design Pattern

Problem Submissions Doubts

✓ Factory Design Pattern

Easy • Score 20/20

Problem statement

What is the primary purpose of the Factory Design Pattern?

Send feedback

Options: Pick one correct answer from below

☒ To encapsulate object creation logic

☐ To implement polymorphism in object-oriented programming


☐ To provide a way to manage database connections

☐ To improve code maintainability and flexibility

Solution description

- The Factory Design Pattern promotes encapsulation by hiding the object creation process from the client code. This allows for easier maintenance and flexibility in changing the implementation details of object creation without affecting the client code.
- It also facilitates loose coupling as clients depend on the Factory interface rather than concrete classes, making it easier to switch between different implementations or extend the system with new types of objects.

(8) Spring Container

 Classroom

Introduction To Spring Boot
Spring Container

Problem Submissions Doubts

✓ Spring Container

Easy • Score 20/20

Problem statement

What is the spring container responsible for?

Send feedback

Options: Pick one correct answer from below

☐ Executing business logic

☐ Configuring bean definitions

☒ Creating and managing spring beans

☐ None of the above

Solution description

- The Spring framework provides a powerful feature called the Spring container, which plays a central role in managing Spring Beans. The Spring container acts as a runtime environment for the application, responsible for creating, configuring, and managing the application's objects.
- The Spring container is primarily represented by two implementations: the BeanFactory and the ApplicationContext.
- The BeanFactory is the core interface for accessing the Spring container, providing basic functions such as bean instantiation, wiring dependencies, and lifecycle management.
- On the other hand, the ApplicationContext extends the BeanFactory interface and adds additional features and functionalities, making it the preferred choice for most applications.