

1. Explain Spring AOP.
2. What is Aspect in Spring AOP?
3. Explain the differences between concern and cross-cutting concern in spring AoP.
4. Explain Join point in spring AOP.
5. What is Advice in Spring AOP?
6. What is Pointcut in Spring AOP?
7. What is Introduction in Spring AOP?
8. What is Target object in Spring AOP?
9. What is a Proxy in Spring AOP?
10. What are the different types of AutoProxying in Spring AOP?
11. What is weaving in Spring AOP? What are the different points where weaving can be applied?
12. Explain XML schema-based aspect implementation in spring AOP.
13. Explain annotation-based (@AspectJ based) aspect implementation in spring AOP.
14. What are the types of Advice in Spring AOP?
15. What is AOP Alliance?
16. What do the proxy-target-class attributes do in Spring AOP?
17. How spring AOP works?
18. Which design pattern is used in Spring AOP?
19. Difference between Spring AOP and aspectJ.

Could not find what you were looking for? [send us the question](#) and we would be happy to answer your question.

Explain Spring AOP.
Aspect-oriented programming (AOP) refers to a programming technique that allows programmers to modularize crosscutting concerns or behavior that cuts across the typical divisions of responsibility, such as logging and transaction management.
What is Aspect in Spring AOP?
The core construct of AOP is the aspect that encapsulates behaviors affecting multiple classes into reusable modules. It is a module which has a set of APIs providing cross-cutting requirements. For example, a logging module would be called AOP aspect for logging. An application can have any number of aspects depending on the requirement. In spring AOP, aspects are implemented using regular classes annotated with @Aspect annotation (@AspectJ style).
Explain the differences between concern and cross-cutting concern in spring AoP.

The concern is behavior that we want to have in a particular module of an application. A concern may be defined as a functionality we want to implement.

The cross-cutting concern is a concern which is applicable throughout the application and it affects the entire application. For example, logging, security and data transfer is applicable for every module of an application hence they are cross-cutting concerns.

Explain Join point in spring AOP.

The Join point represents a point in an applicaiton where we can plug-in an AOP aspect. It is the actual place in the application where an action will be taken using Spring AOP framework.

What is Advice in Spring AOP?

The advice is the actual action that will be taken either before or after the method execution. This is the actual piece of code that is invoked during the program execution by the Spring AOP framework.

Spring aspects can work with five different advices.

- **before:** Run advice before the a method execution.
- **after:** Run advice after the a method execution regardless of its outcome.
- **after-returning:** Run advice after the a method execution only if method completes successfully.
- **after-throwing:** Run advice after the a method execution only if method exits by throwing an exception.
- **around:** Run advice before and after the advised method is invoked.

What is Pointcut in Spring AOP?

The pointcut is a set of one or more joinpoints where an advice should be executed. You can specify pointcuts using expressions or patterns.

What is Introduction in Spring AOP?

An Introduction allows us to add new methods or attributes to existing classes.

What is Target object in Spring AOP?

The target object is an object being advised by one or more aspects. It will always be a proxy object. It is also referred to as the advised object.

What is a Proxy in Spring AOP?

A proxy is an object that is created after applying advice to a target object. When you think of client objects, the target object and the

A proxy is an object that is created after applying advice to a target object. When you think of client objects the target object and the proxy object are the same.

What are the different types of AutoProxying in Spring AOP?

- BeanNameAutoProxyCreator,
- DefaultAdvisorAutoProxyCreator,
- and Metadata autoproxying.

What is weaving in Spring AOP? What are the different points where weaving can be applied?

Weaving is the process of linking aspects with other application types or objects to create an advised object. Weaving can be done at compile time, at load time, or at runtime.

Explain XML schema-based aspect implementation in Spring AOP.

In this implementation case, aspects are implemented using regular classes along with XML based configuration.

Explain annotation-based (@AspectJ based) aspect implementation in Spring AOP.

This implementation case (@AspectJ based implementation) refers to a style of declaring aspects as regular Java classes annotated with Java 5 annotations.

What are the types of Advice in Spring AOP?

There are 5 types of Advice.

- **Before Advice.** Advice that is executed prior to a joinpoint.
- **After returning advice.** Advice that is executed after the normal completion of a joinpoint.
- **After throwing advice.** Advice that is executed only if a method exits abnormally by throwing an exception.
- **Finally advice.** Advice that is executed irrespective of how a joinpoint exits.
- **Around advice.** Advice that borders a joinpoint, for example, a method invocation. Advice can execute before and after the invocation of method.

What is AOP Alliance?

AOP alliance is an open-source project which is aimed at promoting adoption of AOP. The AOP alliance's goal is to define a common set of components and interfaces so as to improve interoperability among different AOP implementations.

What do the proxy-target-class attributes do in Spring AOP?

To force the use of CGLIB proxies set the value of the proxy-target-class attribute of the element to **true**.

```
<aop:config proxy-target-class="true">
  <!-- other beans defined here... -->
</aop:config>
```

How spring AOP works?

Spring AOP is proxy-based. Spring uses either JDK proxy, preferred whenever the proxied target implements at least one interface or CGLIB proxy when the target object does not implement any interfaces, to create the proxy for a given target bean.

Spring AOP performs run-time weaving. You can however set up Spring to do load-time weaving through AspectJ.

Which design pattern is used in Spring AOP?

AOP is mainly based on Proxy design pattern. It also uses remoting, Singleton, template method and decorator design pattern.

Difference between Spring AOP and aspectJ.

Spring AOP does runtime weaving by default while Aspect is compile time weaving.

Spring AOP doesn't work with final class or final/static methods because it may not be able to create proxy by subclassing. AspectJ works with final class as it does compile time weaving and not depend on proxy pattern.

Custom Search