

Reflection API

- ⇒ Reflection API is the set of classes and interfaces in `java.lang.reflect` package
- ⇒ It allows us to inspect, analyze and modify the behaviour of classes, methods, fields or constructors at runtime.
- ⇒ Use :-
 - It is internally used in frameworks (spring, springboot, ORM etc)
 - Code analysis and debugging tools
 - Serialization / Deserialization
 - etc
- ⇒ Classes & Interfaces in Reflection API
 - 1. `Class` (`java.lang`)
 - 2. `Constructor`
 - 3. `Field`
 - 4. `Method`
 - 5. `Array`
 - etc

Interview Questions

- ⇒ Basics of Reflection
 - 1. What is the Reflection API in Java?
 - 2. In which package is the Reflection API available?
 - 3. Why do we use reflection in Java?
 - 4. What are some common use cases of reflection?
 - 5. What are the drawbacks or disadvantages of using reflection?
- ⇒ Class Class
 - 6. What is the role of the `Class` class in reflection?
 - 7. How can you obtain the `Class` object of a given class? (3 ways)
 - 8. What methods are available in the `Class` class to get class details?
 - 9. How do you check whether a `Class` object represents an interface or a primitive type?
 - 10. Can you create an instance of a class using reflection? How?
 - 11. Accessing Fields, Methods, and Constructors
 - 12. How can you get the list of fields of a class using reflection?
 - 13. What is the difference between `getField()` and `getDeclaredField()`?
 - 14. How can you access private fields of a class using reflection?
 - 15. How do you get all methods of a class?
 - 16. What is the difference between `getMethod()` and `getDeclaredMethod()`?
 - 17. How can you invoke a method using reflection?
 - 18. How can you obtain constructors of a class via reflection?
 - 19. How do you create an object using a specific constructor with reflection?
- ⇒ Annotations and Modifiers
 - 20. How can you retrieve annotations of a class or method using reflection?
 - 21. What is the use of the `Modifier` class in reflection?
 - 22. How can you check if a method or field is public, private, or static using reflection?
- ⇒ Advanced & Practical
 - 23. How does reflection break encapsulation?
 - 24. Can reflection be used to change the value of a final field?
 - 25. How does reflection impact performance?
 - 26. Is reflection used internally in frameworks like Spring or Hibernate? If yes, how?
 - 27. What are the security concerns of using reflection?
 - 28. Can reflection be disabled in Java?