Lombok Annotations 101

Learn the Basics in Minutes

Ivan Franchin
@ivanfranchin

@Getter and @Setter

Generates getter and setter methods for class fields, reducing the need for manual boilerplate code. They can be applied at the class or field level, allowing fine-grained control over access.

Ivan Franchin
@ivanfranchin

@ToString

Generates a toString() method for a class, including all fields. It facilitates debugging and logging by providing a concise representation of the object's state.

Ivan Franchin
@ivanfranchin

@EqualsAndHashCode

Generates equals() and hashCode() methods based on the class's fields. It ensures consistency between equality and hash code calculations.

Ivan Franchin
@ivanfranchin

@Data

It is a concise annotation that combines the functionalities of @ToString, @EqualsAndHashCode, @Getter/@Setter, and @RequiredArgsConstructor. Essentially, it automates the generation of common boilerplate code associated with simple Plain Old Java Objects (POJOs) and beans.

Ivan Franchin
@ivanfranchin

@Value

It is the immutable counterpart to @Data. It automatically sets all fields as private and final, eliminates setters, and defaults the class itself to final. Like @Data, it generates essential methods such as toString(), equals(), and hashCode(). Each field receives a getter method, and a constructor covering every argument is also provided (excluding final fields initialized in the field declaration).

Ivan Franchin
@ivanfranchin

@NonNull

Generates null-checks for a field or a parameter in a method. When applied to a field, Lombok generates null-checks in methods that use or set that field. When applied to a method parameter, Lombok generates a null-check at the beginning of the method.

Ivan Franchin
@ivanfranchin

@NoArgs Constructor

Generates a no-argument constructor for a class. This constructor initializes the fields with their default values, which is particularly useful when working with frameworks like Hibernate that require a default constructor.

Ivan Franchin
@ivanfranchin

@RequiredArgs Constructor

Generates a constructor that includes only the non-null and final fields of the class. It's useful when you have fields that are marked as final or are annotated with @NonNull.

Ivan Franchin
@ivanfranchin

@AllArgs Constructor

Generates a constructor with parameters for all fields in the class. It initializes all fields with the provided values.

Ivan Franchin
@ivanfranchin

@Builder

Implements the builder pattern for a class, providing a fluent API for object creation. It simplifies the instantiation of complex objects with many optional parameters.

Ivan Franchin
@ivanfranchin

@With

Generates a new instance of the class with modified values for specified fields. It is commonly used for immutable classes, providing a convenient way to create a new instance with changes to specific fields.

Ivan Franchin
@ivanfranchin

@Slf4j

Integrates with the Simple Logging Facade for Java (SLF4J), automatically creating a logger for the annotated class. It simplifies logging without the need for manual logger instantiation.

Ivan Franchin
@ivanfranchin

@Cleanup

Automatically closes resources (like streams or sockets) at the end of the block, reducing the likelihood of resource leaks.

Ivan Franchin
@ivanfranchin

@Sneaky Throws

Suppresses checked exceptions within a method without explicitly declaring them in the method signature. It simplifies code by avoiding the need for excessive try-catch blocks for checked exceptions that are not expected to occur. However, caution should be exercised as unchecked exceptions will not be caught by the compiler.

Ivan Franchin
@ivanfranchin

Share your thoughts!

Do you like Lombok? Which Lombok annotations do you like or not like? Let us know in the comments!

Ivan Franchin
@ivanfranchin