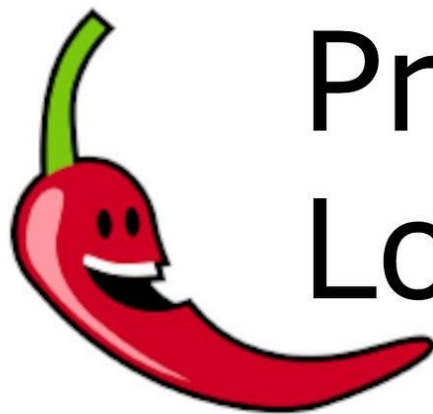


Project Lombok



Project  
Lombok

**Project Lombok** is a Java library that automatically plugs into your editor and build tools, spicing up your java. Never write another getter or equals method again, with one annotation your class has a fully featured builder, Automate your logging variables, and much more.

#### - Team Lombok

Project Lombok is a popular Java library that is designed to reduce the amount of boilerplate code that developers need to write in their Java applications. It achieves this by providing a set of annotations that can be added to Java classes to automatically generate common code structures, such as getters and setters, constructors, equals(), hashCode(), and toString() methods. Lombok helps make Java code more concise, readable, and less error-prone.

Here are some of the key features and annotations provided by Lombok:

**@Getter and @Setter:** These annotations generate getter and setter methods for class fields, eliminating the need to write them manually.

**@ToString:** Generates a **toString()** method that includes all the class's fields for easy debugging and logging.

**@EqualsAndHashCode:** Generates **equals()** and **hashCode()** methods based on the class's fields for object equality comparisons.

**@NoArgsConstructor, @RequiredArgsConstructor, and @AllArgsConstructor:** Generate constructors with no arguments, constructors with required fields, and constructors with all fields, respectively.

**@Data:** A shortcut annotation that combines **@Getter**, **@Setter**, **@ToString**, **@EqualsAndHashCode**, and **@RequiredArgsConstructor** into a single annotation.

**@Builder:** Generates a builder pattern for creating instances of a class with a fluent API, which can be particularly useful for creating complex objects with many optional parameters.

**@Slf4j** and other logging annotations: Simplifies the integration of logging frameworks like SLF4J by generating a logger field for the class.

**Custom Annotations:** Lombok allows you to create custom annotations for code generation, enabling you to automate repetitive tasks specific to your application.

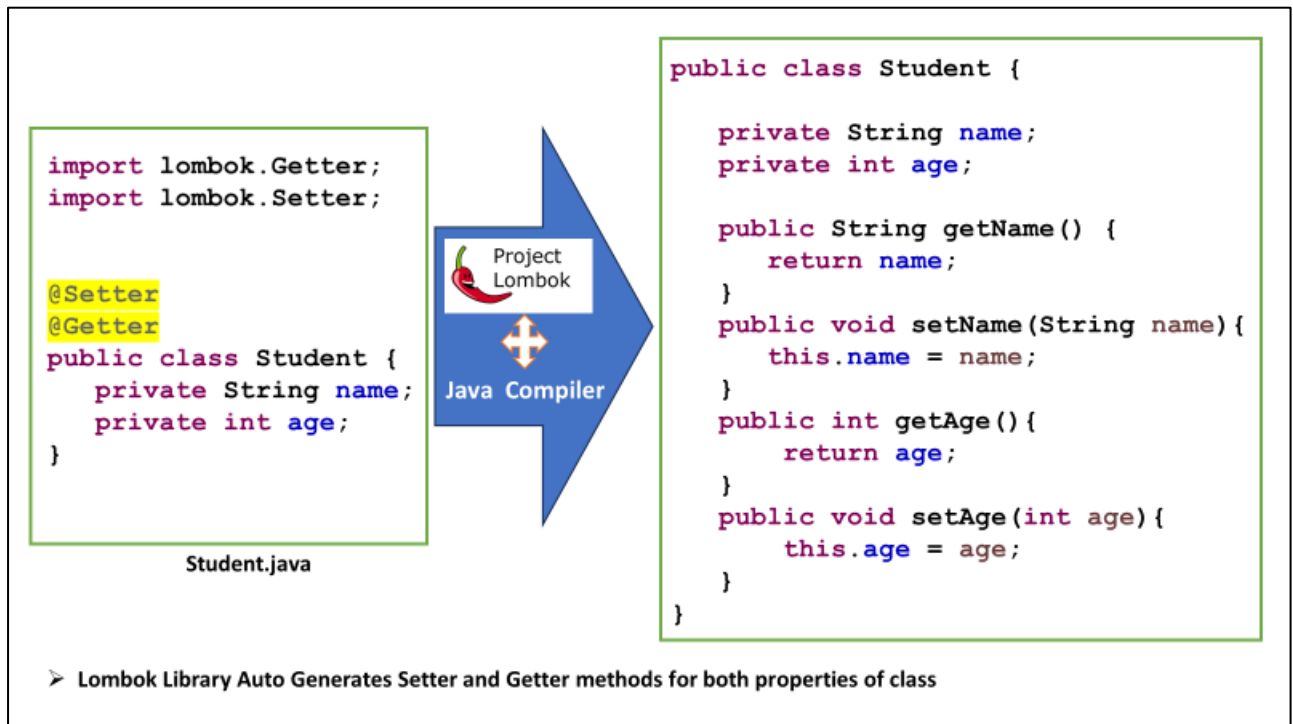
To use Lombok, you typically include it as a dependency in your project's build configuration (e.g., using Maven or Gradle) and enable annotation processing in your IDE. Lombok's annotations are processed at compile-time, which means that the generated code is automatically added to your classes during compilation. Lombok can significantly reduce the amount of boilerplate code you need to write, making your Java code cleaner and more maintainable.

## Enable Lombok Annotation Processing in IDE's:

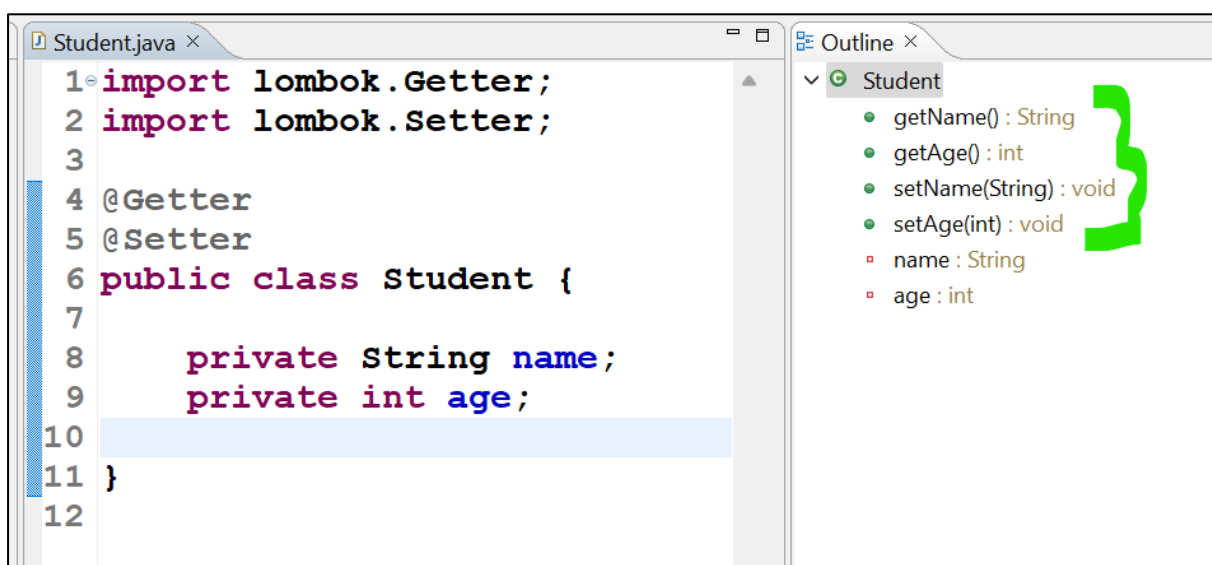
Please follow below link which will guide you to enable Lombok annotation processing.

<https://projectlombok.org/setup/eclipse>

## Annotation Processing by Lombok Example:



## Same we can see inside Outline View of a class in IDE:



➤ Similarly all other Annotations will be processed by Lombok from java Source Code.

➤ Here are some common features and annotations provided by Lombok:

**@Getter and @Setter:** These annotations generate getter and setter methods for class fields. You can apply them at the field or class level.

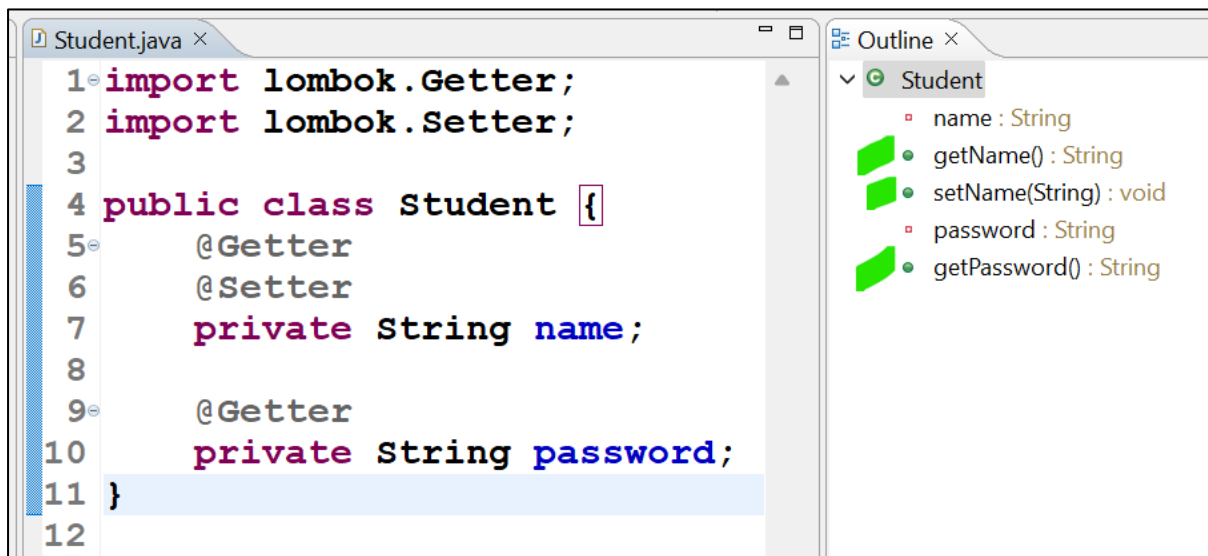
```
import lombok.Getter;
import lombok.Setter;

public class Student {
    @Getter
    @Setter
    private String name;

    @Getter
    private String password;
}
```

From Above Declaration,

- both setter and getter will be generated for property **name**
- Only Getter Method will Be Generated for **password**



**@ToString:** Generates a **toString()** method that includes all fields of the class.

```
import lombok.ToString;

@ToString
public class MyClass {
    //Properties
}
```

**@EqualsAndHashCode**: Generates equals and hashCode methods based on the fields of the class.

```
import lombok.EqualsAndHashCode;

@EqualsAndHashCode
public class MyClass {
    //Properties
}
```

**@NoArgsConstructor**, **@RequiredArgsConstructor**, and **@AllArgsConstructor**: Generate constructors with no arguments, constructors with required fields, and constructors with all fields, respectively.

```
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;
import lombok.RequiredArgsConstructor;

@NoArgsConstructor
@RequiredArgsConstructor
@AllArgsConstructor
public class MyClass {
    //Properties
}
```

**@Data**: Combines @Getter, @Setter, @ToString, @EqualsAndHashCode, and @RequiredArgsConstructor into a single annotation.

```
import lombok.Data;

@Data
public class MyClass {
    //Properties
}
```

**@Builder**: Generates a builder pattern for your class, allowing you to create instances with a fluent API.

```
import lombok.Builder;

@Builder
public class MyClass {
    //Properties
}
```

We do have many other Annotations from Project Lombok. For More and clear Details please refer followed Link:

<https://objectcomputing.com/resources/publications/sett/january-2010-reducing-boilerplate-code-with-project-lombok>