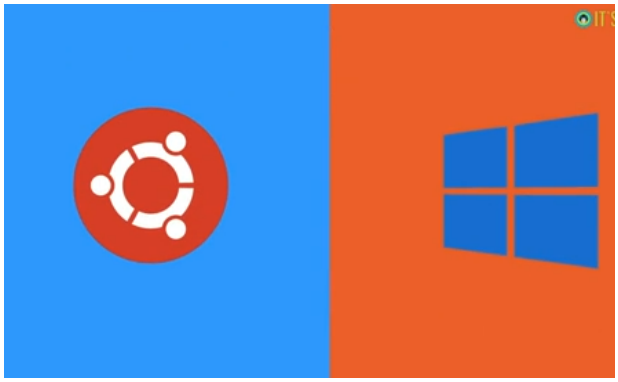
(/)

Setting up Lombok with Eclipse and Intellij



Last updated: June 20, 2022

Written by: baeldung (https://www.baeldung.com/author/baeldung)

DevOps (https://www.baeldung.com/category/devops)

Java (https://www.baeldung.com/category/java) +

Lombok (https://www.baeldung.com/tag/lombok)

1. Overview

Lombok (/intro-to-project-lombok) is a library that facilitates many tedious tasks and reduces Java source code verbosity.

Of course, we usually want to be able to use the library in an IDE, which requires additional setup.

In this tutorial, **we'll talk about configuring Lombok in two of the most popular Java IDEs — IntelliJ IDEA and Eclipse.**

Further reading:

Using Lombok's @Builder Annotation (/lombok-builder)

Learn how the @Builder annotation in Project Lombok can help you reduce boilerplate code when implementing the builder pattern to create instances of your Java classes.

Read more (/lombok-builder) →

Introduction to Project Lombok (/intro-to-project-lombok)

A comprehensive and very practical introduction to many useful usecases of Project Lombok on standard Java code.

Read more (/intro-to-project-lombok) →

2. Lombok in IntelliJ IDEA

As of IntelliJ version 2020.3 (https://www.jetbrains.com/idea/whatsnew/2020-3/#other), we don't need to configure the IDE to use Lombok anymore. The IDE comes bundled with the plugin. Also, the annotation processing will be enabled automatically.

In earlier versions of IntelliJ, we need to perform the below steps to use Lombok. Also, if we use the latest version and the IDE doesn't recognize the Lombok annotation, we need to verify that the below configuration was not disabled manually.

2.1. Enabling Annotation Processing

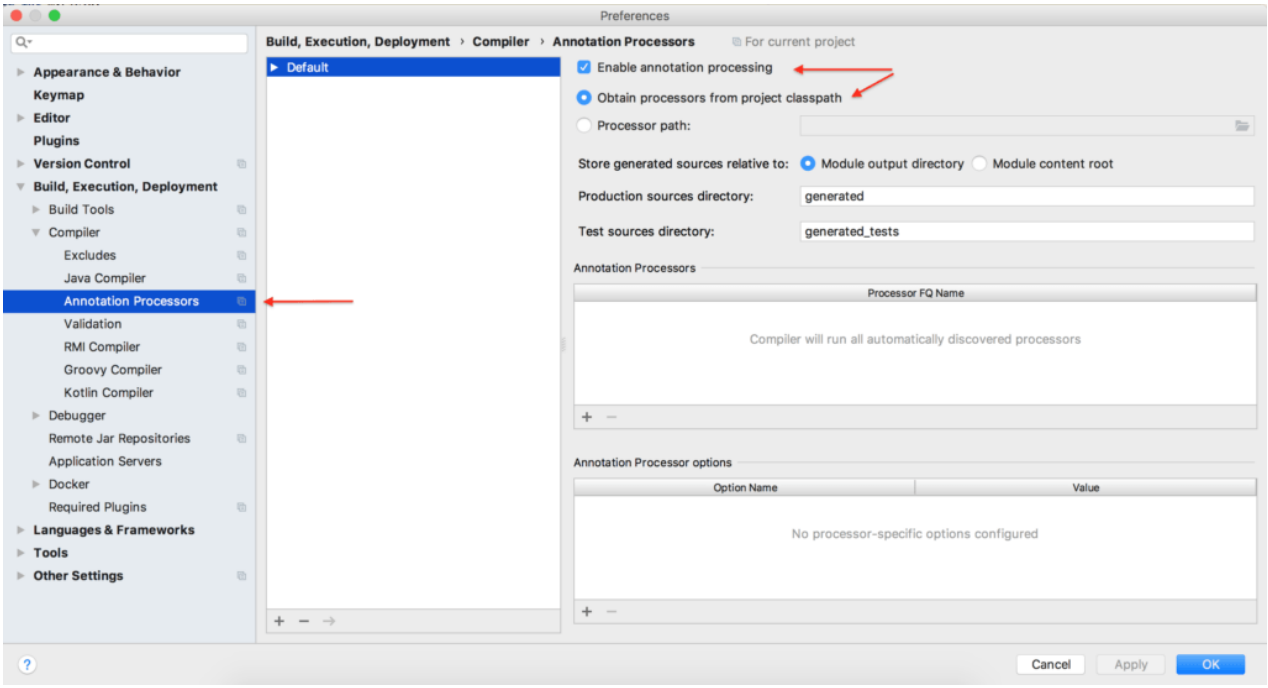
Lombok uses annotation processing through APT (https://docs.oracle.com/javase/7/docs/technotes/guides/apt/GettingStarted.html). So, when the compiler calls it, the library generates new source files based on annotations in the originals.

Annotation processing isn't enabled by default, though.

Therefore, the first thing to do is to enable annotation processing in our project.

We need to go to the *Preferences | Build, Execution, Deployment | Compiler | Annotation Processors* and make sure of the following:

- *Enable annotation processing* box is checked
- *Obtain processors from project classpath* option is selected



(/wp-content/uploads/2019/01/lombok1.png)

2.2. Installing the IDE Plugin

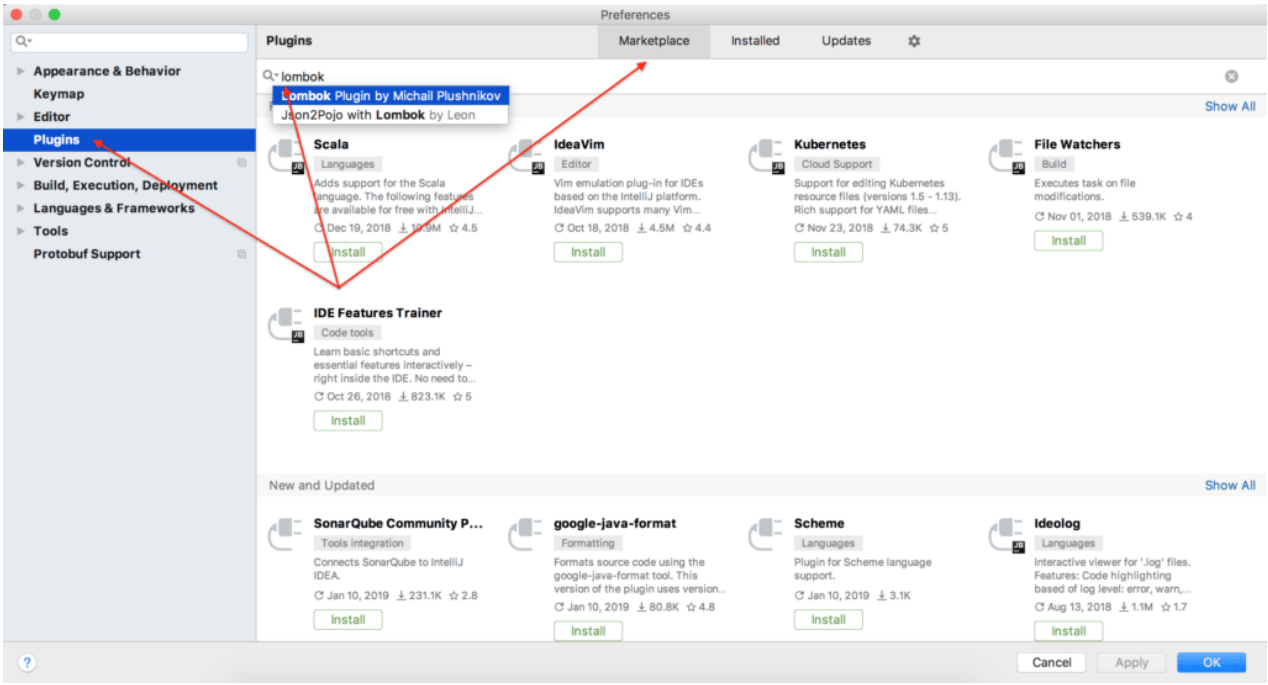
While Lombok generates code only during compilation, the IDE highlights errors in raw source code:

```
public class Start {
    public static void main(String[] args) {
        TestPojo pojo = new TestPojo();
        pojo.setMyField(12);
        System.out.println(pojo.getMyField());
    }
}

public static class TestPojo {

    @Getter
    @Setter
    private int myField;
}
```

(/wp-content/uploads/2019/01/lobom2.png)

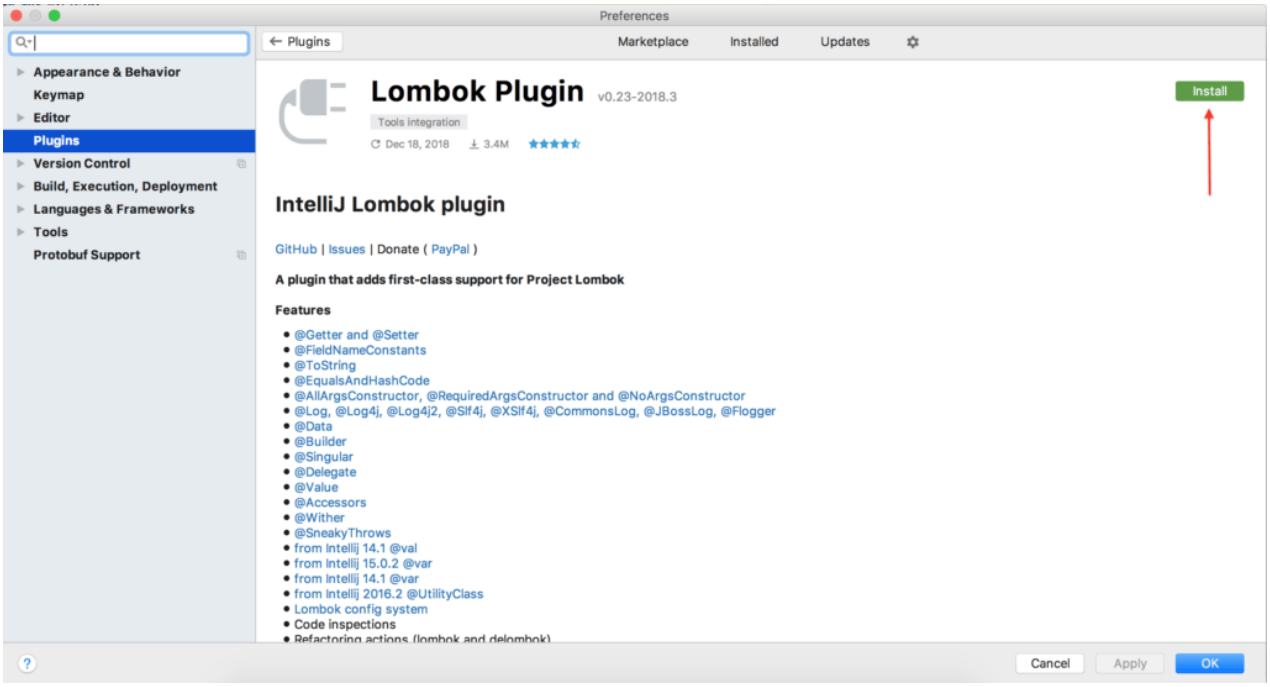
There is a dedicated plugin that makes IntelliJ aware of the source code to be generated. **After installing it, the errors go away and regular features like *Find Usages* and *Navigate To* start working.**

We need to go to the *Preferences | Plugins*, open the *Marketplace* tab, type "lombok" and choose *Lombok Plugin by Michail Plushnikov*.

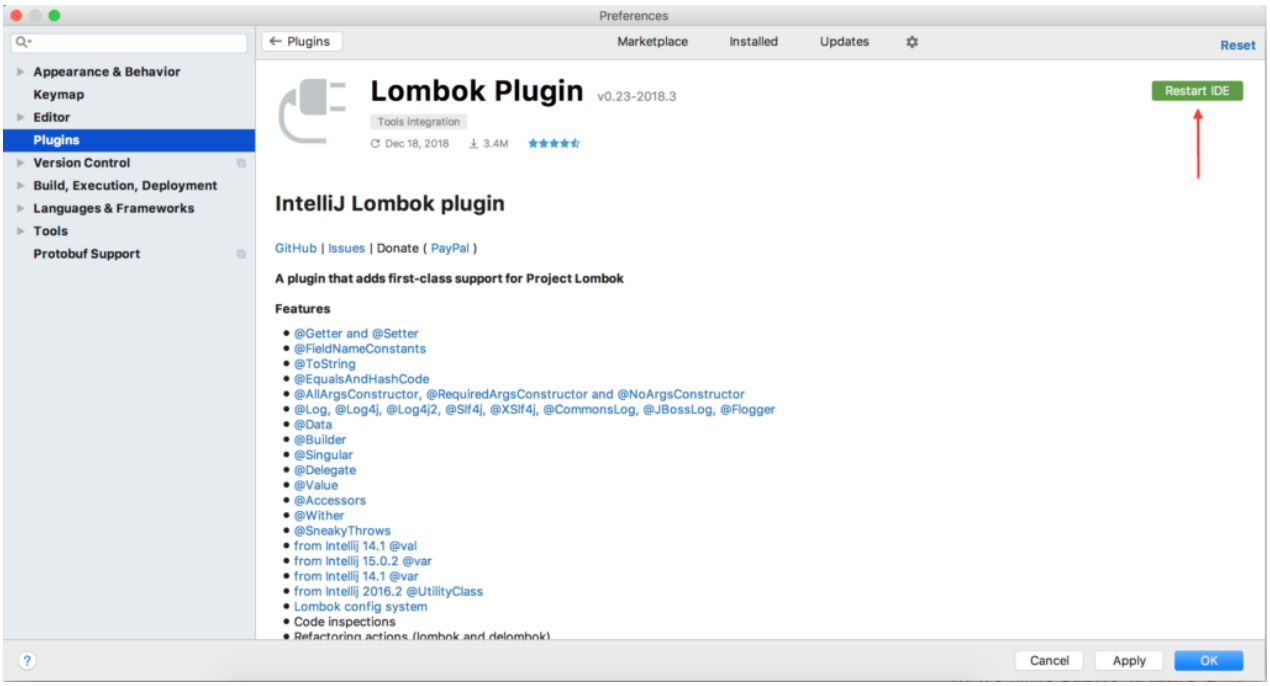


(/wp-content/uploads/2019/01/lombok3.png)

Next, click the *Install* button on the plugin page:



(/wp-content/uploads/2019/01/lombok4.png)
After the installation, click the *Restart IDE* button:



(/wp-content/uploads/2019/01/lombok5.png)

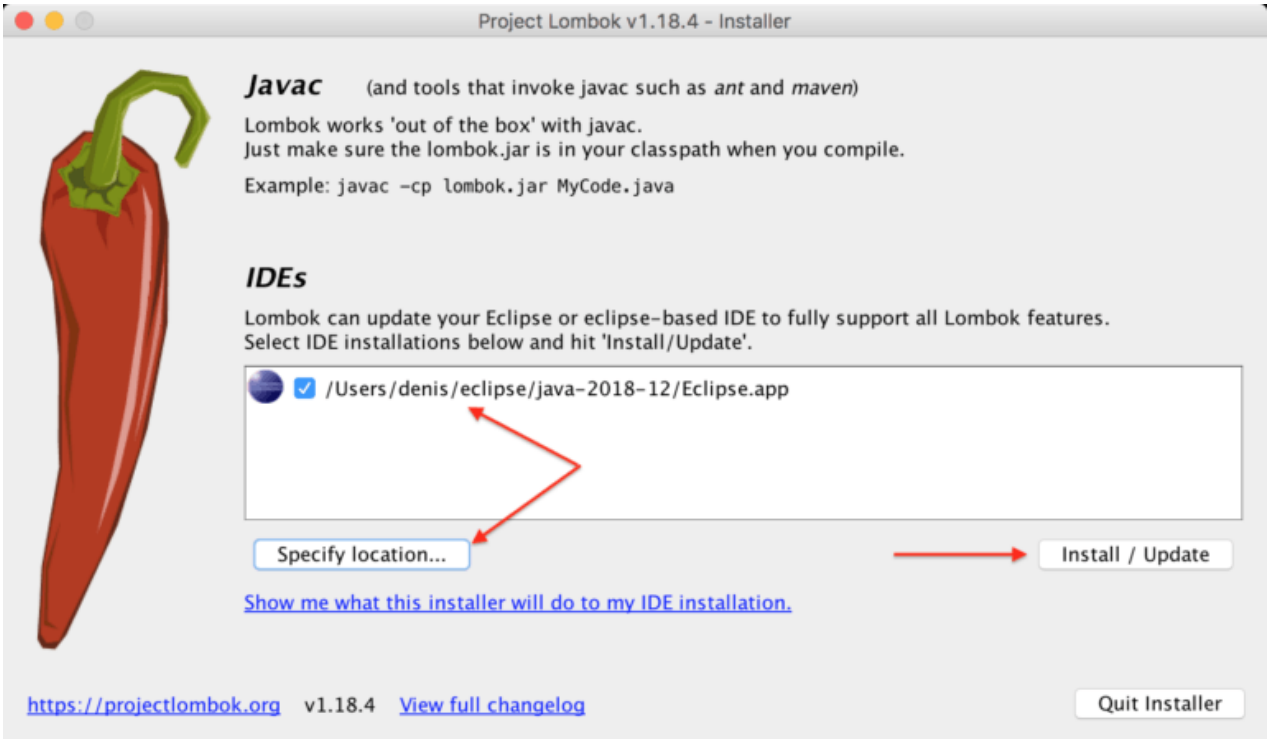
3. Lombok in Eclipse

If we're using Eclipse IDE, we need to get the Lombok jar first. The latest version is located on Maven Central (https://search.maven.org/search?q=g:org.projectlombok%20AND%20a:lombok&core=gav).

For our example, we're using lombok-1.18.4.jar (https://search.maven.org/remotecontent?filepath=org/projectlombok/lombok/1.18.4/lombok-1.18.4.jar).

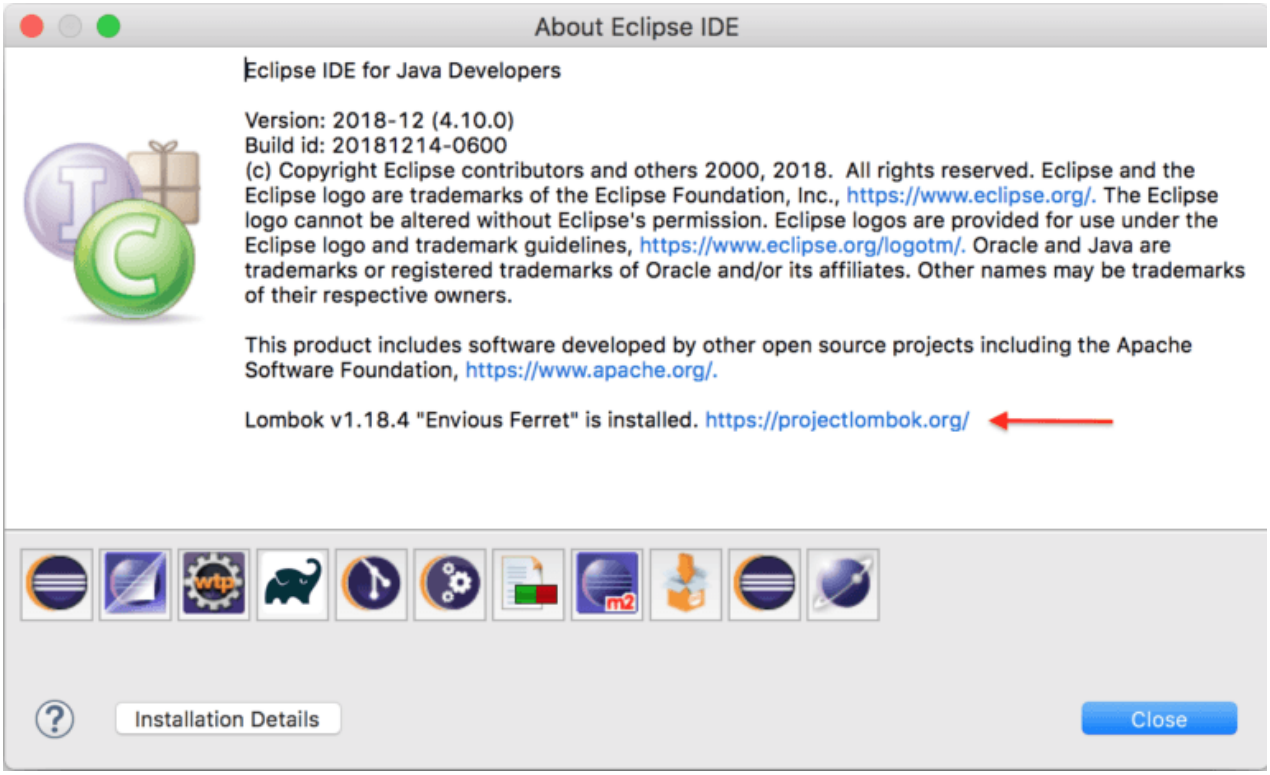
Next, we can run the jar via *java -jar* command, and an installer UI will open. This tries to automatically detect all available Eclipse installations, but it's also possible to specify the location manually.

Once we've selected the installations, we press the *Install/Update* button:



(/wp-content/uploads/2019/01/lombok6.png)
If the installation is successful, we can exit the installer.

After installing the plugin, we need to restart the IDE and ensure that Lombok is correctly configured. We can check this in the *About* dialog:



(/wp-content/uploads/2019/01/lombok7.png)

4. Adding Lombok to the Compile Classpath

The last remaining part is to ensure that Lombok binaries are on the compiler classpath. Using Maven, we can add the dependency to the *pom.xml*:

```
<dependencies>
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <version>1.18.20</version>
        <scope>provided</scope>
    </dependency>
</dependencies>
```

The most recent version is located on Maven Central (https://search.maven.org/search?q=g:org.projectlombok%20AND%20a:lombok&core=gav).

Everything should be fine now. The source code should be shown without errors in the IDE, correctly compiled and executed:

```
public class UserIntegrationTest {

    @Test
    public void givenAnnotatedUser_thenHasGettersAndSetters() {
        User user = new User();
        user.setFirstName("Test");
        assertEquals(user.gerFirstName(), "Test");
    }

    @Getter @Setter
    class User {
        private String firstName;
    }
}
```

5. Conclusion

Lombok does a great job of reducing Java verbosity and covering boilerplate stuff under the hood. In this article, we checked how to configure the tool for the two most popular Java IDEs.

The source code for the examples is available over on GitHub (https://github.com/eugenp/tutorials/tree/master/lombok-modules/lombok).

Comments are closed on this article!

COURSES

ALL COURSES (/ALL-COURSES)
ALL BULK COURSES (/ALL-BULK-COURSES)
ALL BULK TEAM COURSES (/ALL-BULK-TEAM-COURSES)
THE COURSES PLATFORM (HTTPS://COURSES.BAELDUNG.COM)

SERIES

JAVA "BACK TO BASICS" TUTORIAL (/JAVA-TUTORIAL)
JACKSON JSON TUTORIAL (/JACKSON)
APACHE HTTPCLIENT TUTORIAL (/HTTPCLIENT-GUIDE)
REST WITH SPRING TUTORIAL (/REST-WITH-SPRING-SERIES)
SPRING PERSISTENCE TUTORIAL (/PERSISTENCE-WITH-SPRING-SERIES)
SECURITY WITH SPRING (/SECURITY-SPRING)
SPRING REACTIVE TUTORIALS (/SPRING-REACTIVE-GUIDE)

ABOUT

ABOUT BAELDUNG (/ABOUT)
THE FULL ARCHIVE (/FULL_ARCHIVE)
EDITORS (/EDITORS)
JOBS (/TAG/ACTIVE-JOB/)