



链滴

IDEA 插件 Lombok 基本介绍和使用

作者: [Mckaymk](#)

原文链接: <https://ld246.com/article/1576329466405>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

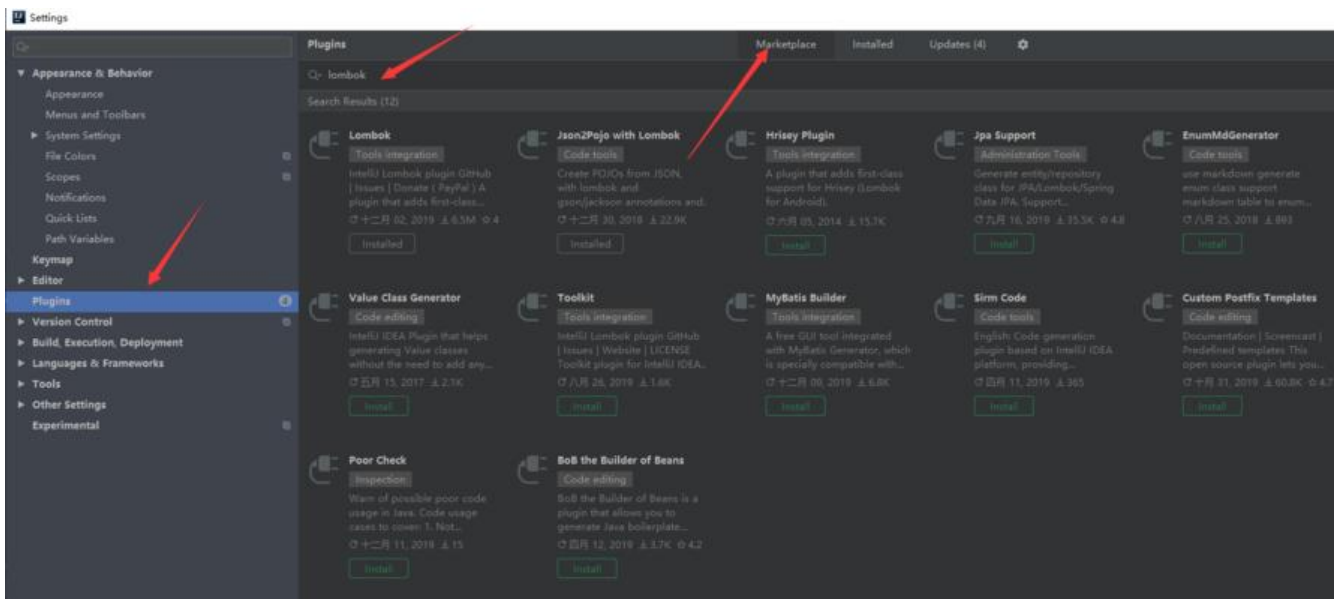
1、基本介绍

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.

以上为[官网介绍](#)，简而言之，Lombok通过注解的方式，简化代码，进行快速开发，通过注解帮你在译的时候生成相应的代码！如getter等方法！

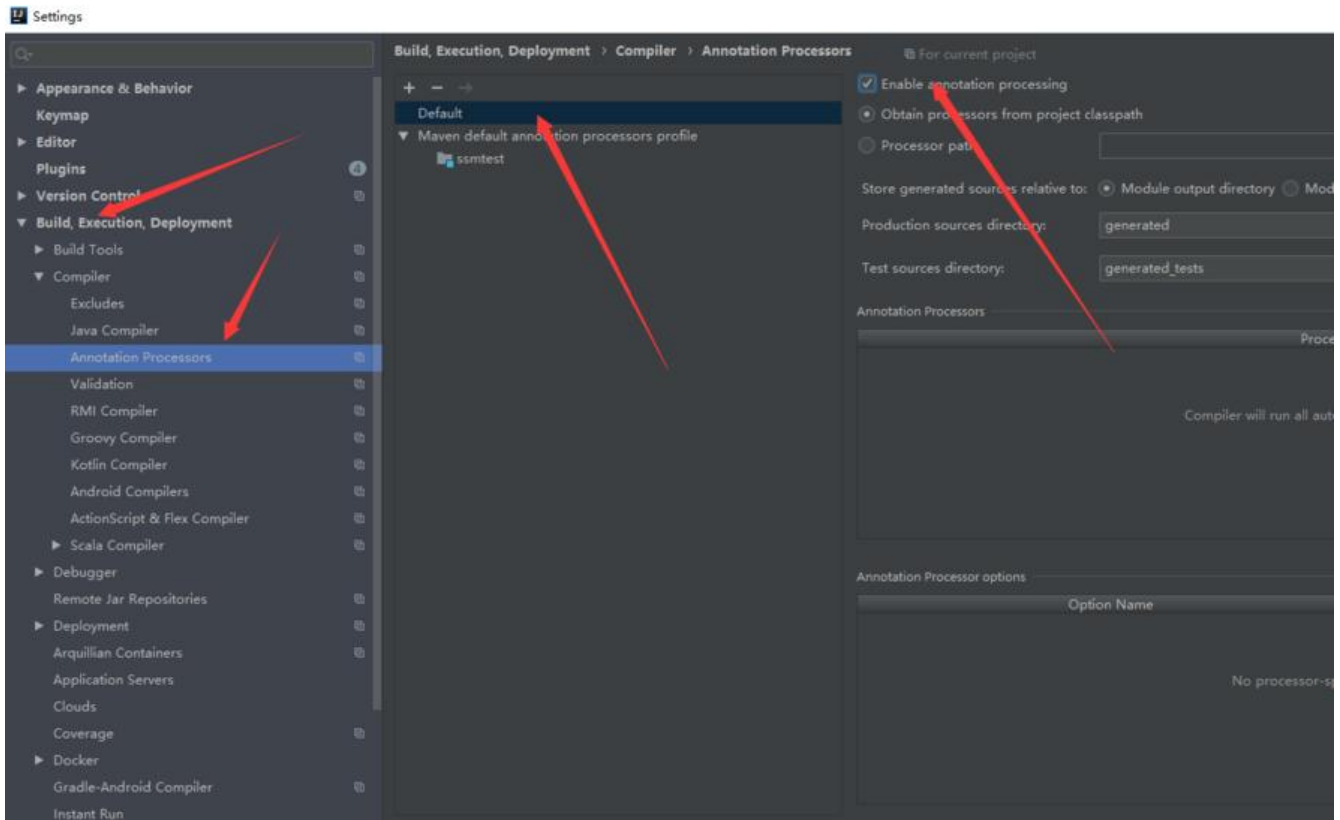
2、插件安装



在IDEA的插件中心搜索lombok，点击安装，完成后，重启即可！

3、简单使用

1、开启Annotation Processors



如上图所示，开启IDEA的Annotation Processors。

2、添加依赖

```
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.4</version>
  <scope>provided</scope>
</dependency>
```

在maven的pom.xml文件中添加上面的依赖，便可以使用

4、相关注解简单介绍

- @Setter/@Getter：注解在属性上；为属性提供 setter/getter方法
- @ToString：为类生成toString方法，输出类名和所有属性
- @EqualsAndHashCode：生成equal和hashCode方法，可以通过参数，排除一些属性
- @NoArgsConstructor：注解在类上；为类提供一个无参的构造方法
- @RequiredArgsConstructor：会生成一个包含常量（final修饰），和标识了NotNull的变量的构造方法。
- @AllArgsConstructor：注解在类上；为类提供一个全参的构造方法
- @Data；注解在类上相当于@Getter、@Setter、@ToString 等几个注解的合集。
- @Log4j：注解在类上，为类提供一个属性名为log 的 log4j 日志对象
- @Cleanup：帮助关闭资源

- @Synchronized : 方法级别, 添加同步锁
 - @SneakyThrows : 等同于try/catch 捕获异常
 - @NonNull : 非null注解, 参数为null会抛出空指针异常
 - @Value : @Data类似, 区别在于它会把所有成员变量默认定义为private final修饰, 并且不会生成get/set方法
 - @Accessors注解: 类级别, 控制各种方法操作。
 - fluent: boolean值, 默认为false。此字段主要为控制生成的getter和setter方法前面是否带get/set; false带, true不带。
 - chain : boolean值, 默认false。如果设置为true, setter返回的是此对象, 可以链式调用方法。
 - prefix : 设置前缀 例如: @Accessors(prefix = "abc") private String abcAge 当生成get/set方法时, 会把此前缀去掉
-

以下是编译前的代码

```
import lombok.Data;
import lombok.experimental.Accessors;

/**
 * @author Mckay
 * @create 2019-12-14
 * @description lombok简单示例, 编译前
 **/
@Data
@Accessors(chain = true)
public class Student {
    private String name;
    private int age;
    private String gender;

    public static void main(String[] args) {
        Student student = new Student();
        student.setName("小明").setAge(24).setGender("男");
        System.out.println(student);
    }
}
```

以下是编译后的代码

```
//
// Source code recreated from a .class file by IntelliJ IDEA
// (powered by Fernflower decompiler)
// lombok简单示例, 编译后
```

```

public class Student {
    private String name;
    private int age;
    private String gender;

    public static void main(String[] args) {
        Student student = new Student();
        student.setName("小明").setAge(24).setGender("男");
        System.out.println(student);
    }

    public Student() {
    }

    public String getName() {
        return this.name;
    }

    public int getAge() {
        return this.age;
    }

    public String getGender() {
        return this.gender;
    }

    public Student setName(String name) {
        this.name = name;
        return this;
    }

    public Student setAge(int age) {
        this.age = age;
        return this;
    }

    public Student setGender(String gender) {
        this.gender = gender;
        return this;
    }

    public boolean equals(Object o) {
        if (o == this) {
            return true;
        } else if (!(o instanceof Student)) {
            return false;
        } else {
            Student other = (Student)o;
            if (!other.canEqual(this)) {
                return false;
            } else {
                label39: {
                    Object this$name = this.getName();
                    Object other$name = other.getName();

```

```

        if (this$name == null) {
            if (other$name == null) {
                break label39;
            }
        } else if (this$name.equals(other$name)) {
            break label39;
        }

        return false;
    }

    if (this.getAge() != other.getAge()) {
        return false;
    } else {
        Object this$gender = this.getGender();
        Object other$gender = other.getGender();
        if (this$gender == null) {
            if (other$gender != null) {
                return false;
            }
        } else if (!this$gender.equals(other$gender)) {
            return false;
        }

        return true;
    }
}
}
}

protected boolean canEqual(Object other) {
    return other instanceof Student;
}

public int hashCode() {
    int PRIME = true;
    int result = 1;
    Object $name = this.getName();
    int result = result * 59 + ($name == null ? 43 : $name.hashCode());
    result = result * 59 + this.getAge();
    Object $gender = this.getGender();
    result = result * 59 + ($gender == null ? 43 : $gender.hashCode());
    return result;
}

public String toString() {
    return "Student(name=" + this.getName() + ", age=" + this.getAge() + ", gender=" + this
getGender() + ")";
}
}
}

```

更多的详细的信息可以查看[官网介绍](#)