

使用 Lombok 於 Java

要在 Java 中使用 Lombok，你需要在項目中設置它，以簡化代碼並減少樣板代碼，如 getter、setter 和構造函數。以下是逐步操作方法：

首先，你需要將 Lombok 添加為依賴項。如果你使用 Maven，請在 `pom.xml` 中包含以下內容：

```
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.32</version> <!-- 檢查最新版本 -->
  <scope>provided</scope>
</dependency>
```

對於 Gradle，請在 `build.gradle` 中添加以下內容：

```
provided 'org.projectlombok:lombok:1.18.32'
// 也添加這行以確保註解處理工作
annotationProcessor 'org.projectlombok:lombok:1.18.32'
```

接下來，如果你使用的是像 IntelliJ IDEA 或 Eclipse 這樣的 IDE，你需要啟用註解處理並安裝 Lombok 插件：

- **IntelliJ**：轉到 File > Settings > Plugins，搜索“Lombok”，安裝它並重新啟動。然後確保在 Settings > Build, Execution, Deployment > Compiler > Annotation Processors 中勾選 Enable annotation processing。
- **Eclipse**：從官方網站（projectlombok.org）下載 Lombok JAR，使用 `java -jar lombok.jar` 運行它，並按照安裝程序將其指向你的 Eclipse 安裝。

現在，你可以在 Java 代碼中開始使用 Lombok 注釋。以下是一個簡單的例子：

```
import lombok.Getter;
import lombok.Setter;
import lombok.NoArgsConstructor;
import lombok.AllArgsConstructor;

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Person {
    private String name;
    private int age;
}
```

在這個例子中：- @Getter 和 @Setter 自動生成 name 和 age 的 getter 和 setter 方法。- @NoArgsConstructor 創建一個無參數的構造函數。- @AllArgsConstructor 創建一個包含所有字段作為參數的構造函數。

當你編譯或運行這個程序時，Lombok 會處理這些註解並為你添加樣板代碼。你可以這樣測試它：

```
public class Main {  
    public static void main(String[] args) {  
        Person person = new Person("Alice", 30);  
        System.out.println(person.getName()); // 輸出: Alice  
        person.setAge(31);  
        System.out.println(person.getAge()); // 輸出: 31  
    }  
}
```

一些你可能會發現有用的流行註解：- @Data：是 @Getter、@Setter、@ToString、@EqualsAndHashCode 和一個構造函數（如 final 字段）的快捷方式。- @Builder：為對象創建提供建造者模式。- @Slf4j：為你的類添加一個日誌記錄器（例如，來自 SLF4J）。

例如，使用 @Data 和 @Builder：

```
import lombok.Data;  
import lombok.Builder;  
  
@Data  
@Builder  
public class User {  
    private String username;  
    private String email;  
}
```

// 使用

```
public class Main {  
    public static void main(String[] args) {  
        User user = User.builder()  
            .username("john_doe")  
            .email("john@example.com")  
            .build();  
        System.out.println(user); // 輸出: User(username=john_doe, email=john@example.com)  
    }  
}
```

請注意：- 總是檢查 Maven Central 或官方網站上的最新版本的 Lombok。- 如果你在團隊中工作，請確保每個人都 IDE 中設置了 Lombok，以避免因生成的代碼在源代碼中不可見而引起的混淆。