

Java 版本與 Spring Boot 相容性

理解錯誤

當你在 Spring Boot 3.4.3 中運行 `mvn spring-boot:run` 並遇到錯誤 “class file has wrong version 61.0, should be 52.0”，這表示存在 Java 版本相容性問題。在 Java 中，類文件有對應的版本號，這個版本號對應於它們被編譯的 Java 版本：

- **版本 52.0** 對應於 Java 8。
- **版本 61.0** 對應於 Java 17。

這個錯誤表示你的 Spring Boot 應用程序的類文件是用 Java 17（版本 61.0）編譯的，但嘗試運行它們的 Java 虛擬機（JVM）期望的是與 Java 8（版本 52.0）相容的類文件。由於 Java 8 的 JVM 無法執行用 Java 17 編譯的類文件（因為 Java 是向後兼容但不是向前兼容），你會遇到這個錯誤。

為什麼會發生這個問題

Spring Boot 3.4.3 要求 **Java 17 或更高版本** 作為其最低支持版本。這意味著你的項目應該同時用至少 Java 17 編譯和運行。這個不匹配可能是因為：

1. **編譯**：你的項目配置為用 Java 17 編譯，產生版本 61.0 的類文件。這通常在你的 `pom.xml` 中用 `maven-compiler-plugin` 或 `<java.version>` 屬性設置。
2. **運行時**：當你執行 `mvn spring-boot:run` 時，Spring Boot Maven 插件啟動一個新的 Java 進程來運行應用程序。然而，它使用的 `java` 可執行文件來自 Java 8 安裝（版本 52.0），這無法處理用 Java 17 編譯的類文件。

修復問題的步驟

要解決這個問題，你需要確保編譯和運行環境都使用 Java 17。以下是如何修復它的方法：

1. 驗證項目的 Java 版本 首先，確認你的項目設置為使用 Java 17。在你的 `pom.xml` 中檢查以下內容：

```
<properties>
  <java.version>17</java.version>
</properties>
```

這個屬性告訴 `maven-compiler-plugin` 用 Java 17 編譯你的代碼。Spring Boot 3.4.3 默認設置這個屬性，但最好驗證一下。如果它遺失或設置為不同的版本（例如 8），將其更新為 17。

2. 安裝 Java 17 確保你的系統上安裝了 Java 17。你可以從以下網站下載：

- Adoptium (Eclipse Temurin)
- Oracle JDK (如果你接受許可條款)

要檢查是否安裝了 Java 17，打開一個終端並運行：

```
java -version
```

如果它沒有顯示 Java 17（例如 `openjdk 17.x.x` 或類似），安裝它並繼續。

3. 更新環境以使用 Java 17 Spring Boot Maven 插件使用環境中的 `java` 可執行文件，通常由 `JAVA_HOME` 環境變量或系統 `PATH` 中的 `java` 命令決定。如果你當前的 `java` 命令指向 Java 8，你需要更新它。

選項 A：設置 `JAVA_HOME` 和 `PATH` 設置你的 `JAVA_HOME` 環境變量指向 Java 17 安裝，並確保 `PATH` 中的 `java` 命令使用它。

• **在 Linux/Mac 上：**

1. 找到你的 Java 17 安裝目錄（例如 `/usr/lib/jvm/java-17-openjdk` 或你安裝的位置）。
2. 在終端中設置 `JAVA_HOME` 並更新 `PATH`：

```
export JAVA_HOME=/path/to/java-17
export PATH=$JAVA_HOME/bin:$PATH
```

3. 驗證：

```
java -version
```

它應該現在顯示 Java 17。

要使其永久生效，將 `export` 行添加到你的 shell 配置文件（例如 `~/.bashrc`，`~/.zshrc`）。

• **在 Windows 上：**

1. 找到 Java 17 安裝目錄（例如 `C:\Program Files\Java\jdk-17`）。
2. 設置 `JAVA_HOME`：
 - 打開系統屬性窗口（在開始菜單中搜索“環境變量”）。
 - 在“系統變量”下，添加或更新 `JAVA_HOME` 為 `C:\Program Files\Java\jdk-17`。
3. 更新 `Path`：
 - 在“系統變量”下，編輯 `Path` 並添加 `%JAVA_HOME%\bin`（確保它在頂部或移除舊的 Java 路徑如 Java 8）。
4. 打開新的命令提示符並驗證：

```
java -version
```

它應該顯示 Java 17。

選項 B：為 Maven 使用特定的 Java 版本 如果你安裝了多個 Java 版本並不想更改系統範圍內的 `JAVA_HOME`，你可以在運行 Maven 時使用特定的 Java 版本：

- 在運行 `mvn spring-boot:run` 之前，僅為該命令設置 `JAVA_HOME`：

```
JAVA_HOME=/path/to/java-17 mvn spring-boot:run
```

這樣可以確保 Spring Boot Maven 插件使用 Java 17，而不改變你的系統設置。

4. 再次運行應用程序 設置好 Java 17 後，運行：

```
mvn spring-boot:run
```

插件現在將使用 Java 17 來執行你的應用程序，匹配編譯的類文件版本（61.0），錯誤應該消失。

額外檢查

- **Maven 的 Java 版本**：運行 `mvn -version` 以查看 Maven 使用的 Java 版本。它應該與你的 `JAVA_HOME` 匹配。如果不匹配，確保你的 `PATH` 和 `JAVA_HOME` 正確配置。
- **依賴項**：如果錯誤持續，某個依賴項可能是用 Java 17 編譯的，而你的運行時仍然配置不當。由於 Spring Boot 3.4.3 強制使用 Java 17，確保你的運行時是 Java 17 應該解決這個問題。

為什麼不使用 Java 8？

Spring Boot 3.4.3 不支持 Java 8。如果你需要使用 Java 8，你將不得不降級到較舊的 Spring Boot 版本（例如 2.x），但除非你有特定的約束，這並不推薦。由於你的類文件是版本 61.0，保持使用 Java 17 符合 Spring Boot 3.4.3 的要求。

最終解決方案總結

要修復在運行 `mvn spring-boot:run` 時遇到的“class file has wrong version 61.0, should be 52.0”錯誤，使用 Spring Boot 3.4.3：

1. 確保 `<java.version>17</java.version>` 在你的 `pom.xml` 中。
2. 如果尚未安裝，安裝 Java 17。
3. 將 `JAVA_HOME` 設置為你的 Java 17 安裝目錄。
4. 更新你的 `PATH`，使 `java -version` 顯示 Java 17。
5. 再次運行 `mvn spring-boot:run`。

這樣可以確保你的運行時與編譯的 Java 17 類文件匹配，解決兼容性問題。