

JAR vs WAR

在 Java 打包中，**JAR**（Java ARchive）和 **WAR**（Web ARchive）都是基於 ZIP 結構的檔案格式，但它們有不同的用途和特性。以下是它們的詳細差異說明：

1. 用途

- **JAR**：JAR 檔案是一種通用的檔案，用於打包 Java 類別、庫或獨立應用程式。它通常用於分發可重用的庫（例如工具類別）或可在 Java 虛擬機（JVM）上運行的可執行 Java 程式。
 - **WAR**：WAR 檔案專門設計用於網頁應用程式。它將網頁應用程式所需的所有組件（例如 Servlets、JSPs（JavaServer Pages）、HTML、CSS、JavaScript 和配置檔案）打包成一個單一單元，以便在網頁伺服器或應用程式伺服器（例如 Apache Tomcat、JBoss）上部署。
-

2. 內容

- **JAR**：包含 Java 類別檔案（.class）、META-INF 目錄下的 MANIFEST.MF 檔案（提供元數據，例如可執行 JAR 的主類別）以及可選的資源，如配置檔案、圖像或屬性檔案。
 - **WAR**：包含特定於網頁的組件，具有定義的結構：
 - **WEB-INF/**：必需的目錄，包含：
 - * web.xml（用於配置 Servlets、映射等的部署描述符）
 - * classes/（編譯的 Java 類別）
 - * lib/（網頁應用程式使用的依賴 JAR 檔案）
 - 靜態資源（例如 HTML、CSS、JavaScript）通常位於根目錄或 WEB-INF 外的子目錄，雖然 JSPs 可能放在 WEB-INF 內以限制直接訪問。
-

3. 結構

- **JAR**：具有平坦結構，主要由類別檔案和資源組成，並且清單檔案指定元數據。範例：

```
myapp.jar
META-INF/
    MANIFEST.MF
com/
    example/
```

```
    MyClass.class
resources/
    config.properties
```

- **WAR**：遵循專為網頁應用程式設計的分層結構。範例：

```
mywebapp.war
  index.html
  css/
    style.css
  WEB-INF/
    web.xml
    classes/
      com/
        example/
          MyServlet.class
    lib/
      dependency.jar
```

4. 部署和使用

- **JAR**：
 - 包含在應用程式的類別路徑中，以提供庫或可重用的程式碼。
 - 如果可執行（在清單中指定 `Main-Class`），可以直接使用 `java -jar myapp.jar` 運行。
 - **WAR**：
 - 部署到 Servlet 容器或應用程式伺服器（例如 Tomcat），該伺服器會解壓縮它並使網頁應用程式在特定 URL（上下文路徑）可訪問。
 - 不能獨立執行；它依賴伺服器來管理其生命週期。
-

5. 依賴處理

- **JAR**：不內建包含其他 JAR 檔案；依賴項由外部管理（例如通過類別路徑或構建工具如 Maven）。
 - **WAR**：可以在 `WEB-INF/lib/` 目錄中打包依賴 JAR 檔案，確保網頁應用程式使用特定版本的庫，與同一伺服器上的其他應用程式隔離。
-

6. 創建

JAR 和 WAR 檔案都可以使用 JDK 的 `jar` 命令行工具或現代構建工具如 Maven 或 Gradle 創建：- **JAR 範例**：`jar cvf myapp.jar -C classes .` - **WAR 範例**：`jar cvf mywebapp.war .`（假設目錄結構包含 WEB-INF 等）。- 在 **Maven** 中，打包類型在 `pom.xml` 中指定：- `<packaging>jar</packaging>` 適用於 JAR。- `<packaging>war</packaging>` 適用於 WAR。

摘要表

特性	JAR	WAR
全名	Java ARchive	Web ARchive
用途	程式庫、獨立應用程式	網頁應用程式
內容	類別、清單、資源	Servlets、JSPs、HTML、WEB-INF 等.
結構	平坦	分層（WEB-INF 等）
部署	類別路徑或 <code>java -jar</code>	網頁伺服器（例如 Tomcat）
依賴	外部	打包在 WEB-INF/lib/

關鍵要點

總的來說，**JAR** 是一種通用的 Java 程式碼檔案，而 **WAR** 是一種專門用於網頁應用程式的檔案，具有專門為網頁伺服器部署設計的結構和組件。了解它們的不同角色有助於在 Java 開發和部署情境中正確使用。