**what is war file and jar files in java\_8 explain with example and command?**



what is war file and jar files in java\_8 exmpalin with example and command ?

In Java, both JAR (Java ARchive) and WAR (Web Application ARchive) files are used to package Java applications, but they serve different purposes and are used in different contexts.

**JAR File (Java ARchive)**

**Purpose:** A JAR file is used to bundle Java class files, libraries, resources (like images, property files), and metadata into a single file for distribution. It is typically used for standalone Java applications or libraries.

**Structure:**

* META-INF/: Contains metadata about the JAR file, including the MANIFEST.MF file.
* \*.class: Compiled Java class files.
* Other resources: Any other files needed by the application, like configuration files or images.

**Creating a JAR File:** To create a JAR file, you can use the jar command-line tool that comes with the JDK.

Example directory structure:

arduino

Copy code

myapp/

├── com/

│ └── example/

│ └── MyApp.class

└── META-INF/

└── MANIFEST.MF

**Example Command:**

sh

jar cf myapp.jar -C myapp/ .

This command creates a myapp.jar file from the contents of the myapp directory.

**WAR File (Web Application ARchive)**

**Purpose:** A WAR file is used to package web applications that run on a server. It contains Java class files, libraries, web pages (JSP, HTML), images, and other resources necessary for a web application.

**Structure:**

* WEB-INF/: Contains all the server-side components of the web application, including:
  + classes/: Contains the compiled Java classes.
  + lib/: Contains JAR files that the application depends on.
  + web.xml: The deployment descriptor.
* Other files: Static resources such as HTML, JSP, images, etc.

**Creating a WAR File:** You can use the jar tool to create a WAR file as well.

Example directory structure:

vbnet

Copy code

mywebapp/

├── WEB-INF/

│ ├── classes/

│ │ └── com/

│ │ └── example/

│ │ └── MyServlet.class

│ ├── lib/

│ │ └── some-library.jar

│ └── web.xml

├── index.html

└── style.css

**Example Command:**

sh

jar cf mywebapp.war -C mywebapp/ .

This command creates a mywebapp.war file from the contents of the mywebapp directory.

**Example Java Code and Commands**

**Example Java Class (for JAR):**

java

// File: com/example/MyApp.java

package com.example;

public class MyApp {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

**Compiling and Creating JAR:**

sh

javac -d myapp/classes com/example/MyApp.java

jar cf myapp.jar -C myapp/classes .

**Example Servlet (for WAR):**

java

// File: com/example/MyServlet.java

package com.example;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class MyServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.getWriter().println("Hello, Web World!");

}

}

**web.xml:**

xml

<!-- File: WEB-INF/web.xml -->

<web-app>

<servlet>

<servlet-name>MyServlet</servlet-name>

<servlet-class>com.example.MyServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>MyServlet</servlet-name>

<url-pattern>/hello</url-pattern>

</servlet-mapping>

</web-app>

**Compiling and Creating WAR:**

sh

javac -d mywebapp/WEB-INF/classes com/example/MyServlet.java

jar cf mywebapp.war -C mywebapp/ .

These commands create the respective archive files, which can be distributed and deployed accordingly. A JAR file can be run using the java -jar command if it’s an executable JAR, and a WAR file can be deployed to a servlet container like Apache Tomcat.