

# WebSphere Liberty - Aplicación Hola Mundo

Aquí tienes un ejemplo sencillo para ejecutar una aplicación “Hello World” en tu servidor WebSphere Liberty, asumiendo que ya lo has configurado con la característica `javaee-8.0` habilitada según las instrucciones anteriores. Este ejemplo utiliza un archivo JSP (JavaServer Pages) básico y lo despliega utilizando el directorio `dropins` por simplicidad.

## Guía Paso a Paso

**1. Crear el Directorio y Archivo de la Aplicación** Crearás una pequeña aplicación web colocando un archivo JSP en el directorio `dropins` de tu servidor Liberty. El directorio `dropins` permite que Liberty detecte y despliegue aplicaciones automáticamente.

- **Localizar el directorio `dropins`:** Navega al directorio `dropins` dentro de la carpeta del servidor. Si tu instalación de Liberty está ubicada en `/opt/ibm/wlp` y tu servidor se llama `myServer`, la ruta es:

```
/opt/ibm/wlp/usr/servers/myServer/dropins
```

Reemplaza `/opt/ibm/wlp` con tu directorio de instalación de Liberty real y `myServer` con el nombre de tu servidor.

- **Crear un directorio WAR descomprimido:** Dentro del directorio `dropins`, crea un directorio llamado `myApp.war`. La extensión `.war` le indica a Liberty que debe tratar esto como una aplicación web. Usa este comando:

```
mkdir -p /opt/ibm/wlp/usr/servers/myServer/dropins/myApp.war
```

- **Crear el archivo `index.jsp`:** Dentro de `myApp.war`, crea un archivo llamado `index.jsp` con el siguiente contenido para mostrar “Hello World!”:

```
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
```

Puedes crearlo directamente con un comando como:

```
echo '<html><body><h2>Hello World!</h2></body></html>' > /opt/ibm/wlp/usr/servers/myServer/dropins/myApp.war/index.jsp
```

Alternativamente, usa un editor de texto para crear `index.jsp` y guárdalo en esa ubicación.

**2. Iniciar el Servidor (si No Está Ya en Ejecución)** Si tu servidor no está en ejecución, necesitas iniciararlo para que pueda desplegar y servir la aplicación.

- **Navegar al directorio bin:** Ve al directorio bin en tu instalación de Liberty:

```
cd /opt/ibm/wlp/bin
```

- **Iniciar el servidor:** Ejecuta el servidor en modo foreground para ver la salida directamente:

```
./server run myServer
```

Alternativamente, inicia en segundo plano:

```
./server start myServer
```

Si el servidor ya está en ejecución, salta este paso; Liberty detectará la nueva aplicación automáticamente.

**3. Verificar el Despliegue de la Aplicación** Liberty desplegará la aplicación myApp.war automáticamente cuando la detecte en el directorio dropins.

- **Revisar la salida de la consola:** Si iniciaste el servidor en modo foreground, busca un mensaje como:

```
[AUDIT    ] CWWKT0016I: Web application available (default_host): http://localhost:9080/myApp/
```

Esto confirma que la aplicación está desplegada y disponible.

- **Revisar los registros (si se ejecuta en segundo plano):** Si el servidor se está ejecutando en segundo plano, busca en el archivo de registro en:

```
/opt/ibm/wlp/usr/servers/myServer/logs/console.log
```

para el mismo mensaje de despliegue.

**4. Acceder a la Aplicación** Ahora que la aplicación está desplegada, puedes verla en un navegador web.

- **Abrir tu navegador:** Navega a:

```
http://localhost:9080/myApp/
```

La raíz del contexto /myApp proviene del nombre del directorio myApp.war, y index.jsp se sirve por defecto como archivo de bienvenida.

- **Resultado esperado:** Deberías ver una página que muestra:

Hello World!

en texto grande.

## Notas

- **Número de puerto:** Este ejemplo usa el puerto 9080, el puerto HTTP predeterminado para Liberty. Si has configurado un puerto diferente, ajusta la URL en consecuencia (por ejemplo, `http://localhost:YOUR_PORT/myApp/`).
- **Permisos:** Asegúrate de tener permisos de escritura en el directorio `dropins`.
- **Soporte JSP:** La característica `javaee-8.0` incluye soporte para JSP, por lo que no se necesita configuración adicional.
- **Solución de problemas:** Si la página no se carga, revisa los registros del servidor en `/opt/ibm/wlp/usr/servers/` para errores, o asegúrate de que el servidor esté en ejecución y los archivos estén en la ubicación correcta.

Este ejemplo sencillo demuestra cómo desplegar y ejecutar una aplicación “Hello World”en tu servidor WebSphere Liberty utilizando el directorio `dropins` para el despliegue automático. ¡Disfruta de tu primera aplicación Liberty!

---

Aquí tienes una explicación de cómo funciona una aplicación “Hello World”cuando se despliega en WebSphere Liberty:

### 1. El Directorio Dropins

WebSphere Liberty proporciona una característica conveniente para desplegar aplicaciones a través de su directorio `dropins`. Este es un folder especial donde puedes colocar simplemente tu aplicación, ya sea como un archivo WAR (Web Application Archive) o un directorio WAR descomprimido, y Liberty lo detectará y desplegará automáticamente. A diferencia de los métodos de despliegue tradicionales, no necesitas configurar manualmente la aplicación en el archivo `server.xml`. Cuando Liberty inicia o nota un cambio en el folder `dropins` (como agregar una nueva aplicación), inicia el proceso de despliegue automáticamente.

### 2. Usar un Directorio WAR Descomprimido

En este ejemplo, la aplicación se despliega como un directorio WAR descomprimido llamado `myApp.war` en lugar de un archivo WAR empaquetado único. Un WAR descomprimido es simplemente una carpeta que contiene todos los contenidos de una aplicación web (como HTML, archivos JSP y otros recursos) en forma descomprimida. Liberty trata este directorio exactamente como lo haría con un archivo WAR, desplegándolo como una aplicación web completamente funcional. Este método es especialmente útil durante el desarrollo porque puedes editar los archivos directamente (por ejemplo, ajustar el HTML o JSP) sin necesidad de volver a empaquetar todo en un archivo WAR.

### **3. El Archivo JSP**

El corazón de esta aplicación “Hello World” es un archivo llamado `index.jsp`, una página JavaServer (JSP). Este archivo contiene HTML básico para mostrar “Hello World!” en la pantalla y podría incluir código Java si es necesario (aunque en este caso se mantiene simple). Cuando accedes a la aplicación, Liberty compila dinámicamente el JSP en un servlet, un pequeño programa Java que genera la página web, y lo sirve a tu navegador.

### **4. Habilitar Características de Java EE**

Para que todo esto funcione, Liberty depende de características específicas habilitadas en su archivo de configuración, `server.xml`. Aquí, la característica `javaee-8.0` está activada, lo que proporciona soporte para tecnologías como JSP, servlets y otros componentes de la plataforma Java Enterprise Edition (EE) 8. Esta característica asegura que Liberty tenga las bibliotecas y configuraciones necesarias cargadas para ejecutar la aplicación sin problemas.

### **5. Proceso de Despliegue Automático**

Una vez que colocas el directorio `myApp.war` en el folder `dropins` e inicias Liberty (o si ya está en ejecución), el servidor detecta y despliega automáticamente la aplicación. Verás mensajes en la salida de Liberty indicando que la aplicación ha comenzado y está disponible en una URL específica. Este proceso de despliegue sin intervención manual hace que sea rápido y fácil poner una aplicación en funcionamiento.

### **6. Acceder a la Aplicación: Raíz del Contexto**

La URL donde puedes acceder a la aplicación se determina por su **raíz del contexto**, que es simplemente el nombre del archivo WAR o directorio. En este caso, ya que el directorio se llama `myApp.war`, la raíz del contexto se convierte en `/myApp`. Por defecto, Liberty se ejecuta en el puerto 9080, por lo que la URL completa para acceder a la aplicación es `http://localhost:9080/myApp/`.

### **7. Servir el Archivo de Bienvenida**

Cuando visitas `http://localhost:9080/myApp/` en tu navegador, Liberty necesita determinar qué mostrar. Busca un **archivo de bienvenida**, una página predeterminada para servir cuando no se solicita un archivo específico. En este caso, `index.jsp` está ubicado en la raíz del directorio `myApp.war` y se reconoce como un archivo de bienvenida por defecto. Liberty compila y sirve `index.jsp`, que luego muestra “Hello World!” en tu pantalla.

## **Resumen**

Esta aplicación “Hello World” muestra cómo WebSphere Liberty simplifica el despliegue de aplicaciones web. Al aprovechar el directorio `dropins` para el despliegue automático, soportar WARs descomprimidos para una edición fácil y usar la característica `javaee-8.0` para la compatibilidad con Java EE, Liberty te permite desplegar y probar rápidamente una aplicación web simple sin una configuración compleja. El resultado es una forma ligera y amigable para los desarrolladores de ver “Hello World!” en <http://localhost:9080/myApp/>, perfecta para aprender o prototipar.