

Troubleshoot Deployment in WebSphere Liberty Profile

When you try to access your Spring Boot application at `http://localhost:9080/myapp/` and encounter a “context root not found” error on WebSphere Liberty Profile (WLP), it indicates that WLP has not successfully deployed your application under the expected context root (`/myapp`). This issue typically arises due to problems with deployment configuration, server setup, or the WAR file itself. Below is a comprehensive guide to troubleshoot and resolve this issue.

Understanding the Problem

The “context root not found” error means that WLP cannot locate an application deployed at `/myapp`. Normally, when you place a WAR file named `myapp.war` in WLP’s `dropins` directory, it should automatically deploy with the context root `/myapp`, making it accessible at `http://localhost:9080/myapp/`. Since this isn’t happening, we need to investigate why the deployment failed.

Troubleshooting Steps

1. Check Server Logs for Deployment Messages The first step is to confirm whether WLP has deployed your application.

- **Locate the Logs:**

- If your server is named `myServer`, check the logs at:

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

or

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

- If you’re using the default server, replace `myServer` with `defaultServer`.

- **Look for Deployment Confirmation:**

- You should see a message like:

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

This indicates the application is deployed and available.

- Additionally, look for:

```
CWWKZ0001I: Application myapp started in X.XXX seconds.
```

This confirms the application has started successfully.

- **What to Do:**

- If these messages are absent, the application hasn't deployed. Look for any `ERROR` or `WARNING` messages in the logs that might indicate why (e.g., missing features, file permissions, or startup failures).
- If you see Spring Boot startup logs (e.g., the Spring Boot banner), the application is loading, and the issue might be with the context root or URL mapping.

2. Verify the WAR File Location and Permissions Ensure the WAR file is correctly placed in the `dropins` directory and is accessible to WLP.

- **Check the Path:**

- For a server named `myServer`, the WAR file should be at:

```
/opt/ibm/wlp/usr/servers/myServer/dropins/myapp.war
```

- If using `defaultServer`, adjust accordingly:

```
/opt/ibm/wlp/usr/servers/defaultServer/dropins/myapp.war
```

- **Verify Permissions:**

- Ensure the WLP process has read permissions for the file. On a Unix-like system, run:

```
ls -l /opt/ibm/wlp/usr/servers/myServer/dropins/myapp.war
```

The file should be readable by the user running WLP (e.g., `rw-r--r--`).

- **What to Do:**

- If the file is missing or misplaced, copy it to the correct `dropins` directory:

```
cp target/myapp.war /opt/ibm/wlp/usr/servers/myServer/dropins/
```

- Fix permissions if needed:

```
chmod 644 /opt/ibm/wlp/usr/servers/myServer/dropins/myapp.war
```

3. Confirm `dropins` Monitoring in `server.xml` WLP's `dropins` directory is enabled by default, but custom configurations might disable it.

- **Check `server.xml`:**

- Open the server configuration file:

```
/opt/ibm/wlp/usr/servers/myServer/server.xml
```

- Look for the `applicationMonitor` element:

```
<applicationMonitor updateTrigger="polled" pollingRate="5s" dropins="dropins" />
```

This confirms that WLP monitors the `dropins` directory every 5 seconds for new applications.

- **What to Do:**

- If absent, add the above line within the `<server>` tags or ensure no overriding configuration disables `dropins`.
- Restart the server after changes:

```
/opt/ibm/wlp/bin/server stop myServer  
/opt/ibm/wlp/bin/server start myServer
```

4. Ensure Required Features Are Enabled

WLP requires specific features to deploy a Spring Boot WAR file, such as Servlet support.

- **Check** `server.xml`:

- Verify the `featureManager` section includes:

```
<featureManager>  
    <feature>javaee-8.0</feature>  
</featureManager>
```

The `javaee-8.0` feature includes Servlet 4.0, which is compatible with Spring Boot. Alternatively, at least `servlet-4.0` should be present.

- **What to Do:**

- If missing, add the feature and restart the server.

5. Validate the WAR File Structure

A corrupted or improperly structured WAR file might prevent deployment.

- **Inspect the WAR:**

- Unzip the WAR file to verify its contents:

```
unzip -l myapp.war
```

- Look for:

- * `WEB-INF/classes/com/example/demo/HelloController.class` (your controller class).
- * `WEB-INF/lib/` containing Spring Boot dependencies (e.g., `spring-web-x.x.x.jar`).

- **What to Do:**

- If the structure is incorrect, rebuild the WAR:

```
mvn clean package
```

Ensure your `pom.xml` sets `<packaging>war</packaging>` and marks `spring-boot-starter-tomcat` as `<scope>provided</scope>`.

6. Alternative Deployment via apps Directory If dropins fails, try deploying the application explicitly via the apps directory.

- **Steps:**

- Move the WAR file:

```
mv /opt/ibm/wlp/usr/servers/myServer/dropins/myapp.war /opt/ibm/wlp/usr/servers/myServer/apps/
```

- Edit server.xml to add:

```
<application id="myapp" name="myapp" type="war" location="${server.config.dir}/apps/myapp.war">
    <context-root>/myapp</context-root>
</application>
```

- Restart the server:

```
/opt/ibm/wlp/bin/server restart myServer
```

- **Test Again:**

- Access <http://localhost:9080/myapp/>. If it works, the issue was with dropins.

7. Verify Server Status Ensure the server is running correctly.

- **Check Status:**

```
/opt/ibm/wlp/bin/server status myServer
```

- If stopped, start it:

```
/opt/ibm/wlp/bin/server start myServer
```

- **Test WLP Itself:**

- Visit <http://localhost:9080/>. If WLP is running, you'll see a welcome page (unless overridden). If this fails, there's a broader server issue.

Resolution

To fix the "context root not found" error:

1. **Start Here:**

- Check the server logs (`messages.log` or `console.log`) for deployment messages or errors.
- Ensure `myapp.war` is in the `dropins` directory with proper permissions.

2. **Next Steps:**

- Verify that `server.xml` enables the `javaee-8.0` feature and monitors dropins.
- If unresolved, deploy via the `apps` directory with an explicit `<application>` configuration in `server.xml`.

3. **Final Verification:**

- After making changes, restart WLP and access `http://localhost:9080/myapp/`. You should see “Hello World!” if the deployment succeeds.

By following these steps, you’ll identify whether the issue lies with the server configuration, WAR placement, or the WAR file itself, and resolve it accordingly.