

## समस्या को समझना

यह लगता है कि आपने mvn package का उपयोग करके mvn package का उपयोग करके एक `war` (या `jar`) फ़ाइल सफलतापूर्वक बनाया है, और निर्माण प्रक्रिया में कोई त्रुटि नहीं आई। फिर भी, जब आप इस `war` को `application.xml` (या `application.xml`) पर डिप्लॉय करते हैं और सर्वर को शुरू करते हैं, तो आप “`java.lang.ClassNotFoundException: com.example.MyApp`” त्रुटि से सामना करते हैं। यह हैरान करने वाला है क्योंकि एक सुलभ निर्माण आमतौर पर सब कुछ ठीक होने का संकेत देता है, फिर भी सर्वर आपकी एप्लिकेशन को अपेक्षित तरीके से पहचान नहीं कर रहा है। चलिए इसे तोड़ते हैं और समझते हैं कि क्या हो रहा हो सकता है, साथ ही समस्या को सुलझाने के कदम भी देखते हैं।

### समस्या को समझना

एक `war` फ़ाइल एक मानक `war` आर्काइव है जो आपकी एप्लिकेशन को पैकेज करता है, आमतौर पर एक या अधिक मॉड्यूल जैसे `web` फ़ाइलें (वेब एप्लिकेशन के लिए), `ejb` फ़ाइलें (या लाइब्रेरी के लिए) और एक डिप्लॉयमेंट डिस्क्रिप्टर (`application.xml`) शामिल होती हैं। जब आप एक `war` को `web` पर डिप्लॉय करते हैं, तो सर्वर इन मॉड्यूलों को पहचानना चाहिए और एप्लिकेशन को शुरू करना चाहिए। “`java.lang.ClassNotFoundException: com.example.MyApp`” संदेश यह सुझाता है कि `web` या तो आपकी `web` में कोई मॉड्यूल नहीं मिल पा रहा है या उन्हें पहचान नहीं कर रहा है, जिससे एप्लिकेशन शुरू नहीं हो पा रहा है।

जिस समय आपका `war` निर्माण सफल रहा (“सब कुछ सुलभ”), समस्या इन तीन क्षेत्रों में से किसी एक में हो सकती है: 1. **फ़ाइल के सामग्री:** `web` में अपेक्षित मॉड्यूल हो सकते हैं या डिप्लॉयमेंट डिस्क्रिप्टर (`application.xml`) गायब हो सकता है या गलत हो सकता है। 2. **डिप्लॉयमेंट प्रक्रिया:** आप `web` को `web` पर डिप्लॉय करने का तरीका, सर्वर द्वारा इसे खोजने और प्रोसेस करने की अपेक्षा से मेल नहीं खा सकता है। 3. **सर्वर कॉन्फ़िगरेशन:** `web` आपके `web` में मॉड्यूल को पहचानने के लिए सेट अप नहीं हो सकता है, क्योंकि आवश्यक विशेषताएं गायब हैं या गलत कॉन्फ़िगरेशन है।

चलिए इन सम्भावनाओं को खोजते हैं और कार्यात्मक कदम प्रदान करते हैं ताकि समस्या को निदान और ठीक कर सकें।

### सम्भावित कारण और समाधान

1. **फ़ाइल खाली हो सकती है या गलत तरीके से पैकेज की गई हो सकती है** हालांकि निर्माण सफल रहा, लेकिन संभव है कि आपके `war` में कोई मॉड्यूल (जैसे `web` या `ejb` फ़ाइलें) नहीं हों या `application.xml` फ़ाइल, जो सर्वर को बताती है कि कौन से मॉड्यूल शामिल हैं, गायब हो या गलत हो।

□ **यह क्यों होता है:** एक `war` प्रोजेक्ट में, `maven-ear-plugin` को सेट करने के लिए जिम्मेदार है। यह `pom.xml` कॉन्फ़िगरेशन या निर्भरताओं के आधार पर मॉड्यूल शामिल करता है। यदि कोई मॉड्यूल निर्धारित नहीं हैं, या निर्भरताएं (जैसे एक `web`) सही तरीके से परिभाषित या सुलझाई नहीं हैं, तो `war` खाली हो सकता है या एक सही `application.xml` के बिना हो सकता है।

□ **इसका जांच करने का तरीका:**

- अपने `war` फ़ाइल (यह एक `war` आर्काइव है) को एक टूल जैसे `unzip` का उपयोग करके खोलें या टर्मिनल में `jar tf myApp.war` चलाएं ताकि इसके सामग्री को सूचीबद्ध कर सकें।
- इनकी तलाश करें:
  - ★ मॉड्यूल फ़ाइलें (जैसे `my-web.war`, `my-ejb.jar`) `WEB-INF` के रूट में।
  - ★ एक `META-INF/application.xml` फ़ाइल।
- `application.xml` के अंदर, आप अपने मॉड्यूलों को परिभाषित करने वाले प्रविष्टियों को देखेंगे, जैसे:

```
<?xml version="1.0" encoding="UTF-8"?>
<application>
  <module>
    <web>
      <web-uri>my-web.war</web-uri>
      <context-root>/myapp</context-root>
    </web>
  </module>
</application>
```

#### □ इसका ठीक करने का तरीका:

- □□□ मॉड्यूल के लिए pom.xml की जांच करें। सुनिश्चित करें कि आपने शामिल करने के लिए मॉड्यूलों के लिए निर्भरताओं को परिभाषित किया है, जैसे:

```
<dependencies>
  <dependency>
    <groupId>com.example</groupId>
    <artifactId>my-web</artifactId>
    <type>war</type>
  </dependency>
</dependencies>
```

- आवश्यकता के अनुसार maven-ear-plugin को कॉन्फ़िगर करें:

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-ear-plugin</artifactId>
  <version>3.3.0</version>
  <configuration>
    <modules>
      <webModule>
        <groupId>com.example</groupId>
        <artifactId>my-web</artifactId>
        <contextRoot>/myapp</contextRoot>
      </webModule>
    </modules>
  </configuration>
</plugin>
```

- mvn clean package के साथ पुनः निर्माण करें और □□□ सामग्री को पुनः जांच करें।

यदि □□□ खाली है या application.xml गायब/गलत है, तो यह संभवतः मूल कारण है। □□□□ कॉन्फ़िगरेशन को ठीक करने से इसे हल किया जा सकता है।

**2. डिप्लॉयमेंट विधि समस्या** आप `dropins` को `server.xml` पर डिप्लॉय करने का तरीका भी समस्या हो सकता है। `dropins` दो प्राथमिक डिप्लॉयमेंट विधियों का समर्थन करता है: `dropins` डिरेक्टरी और `server.xml` में स्पष्ट कॉन्फिगरेशन।

□ **dropins डिरेक्टरी का उपयोग:**

- यदि आपने `dropins` को `wlp/usr/servers/<serverName>/dropins/` डिरेक्टरी में रखा है, तो `dropins` इसे स्वचालित रूप से पहचानना और डिप्लॉय करना चाहिए।
- फिर भी, `dropins` फ़ाइलों के लिए स्वचालित डिप्लॉयमेंट कभी-कभी अपेक्षित तरीके से काम नहीं कर सकता, खासकर यदि अतिरिक्त कॉन्फिगरेशन (जैसे कंटेक्स्ट रूट) की आवश्यकता होती है।

□ **server.xml का उपयोग:**

- `dropins` फ़ाइलों के लिए, यह अक्सर बेहतर है कि आप `wlp/usr/servers/<serverName>/server.xml` में एप्लिकेशन को स्पष्ट रूप से कॉन्फिगर करें:

```
<server>
  <featureManager>
    <feature>servlet-3.1</feature> <!--
  </featureManager>
  <application id="myApp" name="myApp" type="ear" location="${server.config.dir}/apps/myApp.ear"/>
</server>
```

- `dropins` को `wlp/usr/servers/<serverName>/apps/` में रखें (या `location` पथ को अनुकूलित करें)।

□ **इसका जांच करने का तरीका:**

- सुनिश्चित करें कि आपने `dropins` को कहाँ रखा है और सर्वर को कैसे शुरू किया है (जैसे, `./bin/server run <serverName>`).
- सर्वर लॉग (जैसे, `wlp/usr/servers/<serverName>/logs/console.log` या `messages.log`) में डिप्लॉयमेंट संदेशों की जांच करें।

□ **इसका ठीक करने का तरीका:**

- ऊपर दिखाए गए तरीके से `server.xml` में `dropins` को कॉन्फिगर करने की कोशिश करें, `dropins` पर निर्भर करने के बजाय।
- बदलावों के बाद सर्वर को पुनः शुरू करें: `./bin/server stop <serverName>` के बाद `./bin/server start <serverName>`.

यदि `dropins` को सर्वर के साथ सही तरीके से पंजीकृत नहीं किया गया था, तो यह त्रुटि का कारण हो सकता है।

---

**3. गायब सर्वर विशेषताएं** `dropins` एक हल्का सर्वर है जो केवल `server.xml` में आपने सक्रिय किए गए विशेषताओं को लोड करता है। यदि आपके `dropins` में मॉड्यूल शामिल हैं जो विशेष विशेषताओं (जैसे, सर्वलेट्स, `dropins`) की आवश्यकता होती हैं, और वे विशेषताएं सक्रिय नहीं हैं, तो `dropins` मॉड्यूलों को पहचानने या लोड करने में विफल हो सकता है।

- **यह क्यों होता है:** उदाहरण के लिए, एक `dropins` फ़ाइल को `servlet-3.1` विशेषता (या उच्चतर) की आवश्यकता होती है, जबकि एक `dropins` मॉड्यूल को `ejbLite-3.2` की आवश्यकता होती है। इनके बिना, सर्वर मॉड्यूलों को प्रोसेस करने में विफल हो सकता है।

□ **इसका जांच करने का तरीका:**

- `server.xml` को देखें और `<featureManager>` खंड की जांच करें।

- आम विशेषताएं शामिल हैं:

- ★ `<feature>servlet-3.1</feature>` वेब मॉड्यूलों के लिए।
- ★ `<feature>ejbLite-3.2</feature>` EJB मॉड्यूलों के लिए।

- सर्वर लॉग में गायब विशेषताओं के संदेशों की जांच करें (जैसे, “अनिवार्य विशेषता इंस्टॉल नहीं है”).

#### □ इसका ठीक करने का तरीका:

- `server.xml` में अपने एप्लिकेशन की आवश्यकताओं के आधार पर आवश्यक विशेषताएं जोड़ें:

```
<featureManager>
    <feature>servlet-3.1</feature>
    <!--
    -->
</featureManager>
```

- बदलाव लागू करने के लिए सर्वर को पुनः शुरू करें।

यदि विशेषताएं गायब हैं, तो उन्हें सक्रिय करने से `server.xml` को मॉड्यूल पहचानने में मदद मिलेगी।

## निदान कदम

समस्या को निदान करने के लिए इन कदमों का पालन करें:

#### 1. `server.xml` फ़ाइल की जांच करें:

- `jar tf myApp.ear` चलाएं या इसे अनज़िप करें।
- सुनिश्चित करें कि इसमें आपके मॉड्यूल (जैसे `my-web.war`) और एक वैध `META-INF/application.xml` शामिल हैं।

#### 2. `server.xml` निर्माण की जांच करें:

- `server.xml` मॉड्यूल के `pom.xml` को समीक्षित करें ताकि निर्भरताओं और `maven-ear-plugin` कॉन्फ़िगरेशन की पुष्टि हो सके।
- `mvn clean package` फिर से चलाएं और निर्माण आउटपुट में मॉड्यूलों को शामिल करने के संदेशों की जांच करें (जैसे, “मॉड्यूल `my-web.war` को `myApp.ear` जोड़ना”).

#### 3. डिप्लॉयमेंट की पुष्टि करें:

- `server.xml` का स्थान (जैसे, `dropins` या `apps`) की पुष्टि करें।
- यदि `dropins` का उपयोग किया जा रहा है, तो इसे `apps` में स्थानांतरित करने की कोशिश करें और इसे `server.xml` में जोड़ें।

#### 4. सर्वर लॉग की जांच करें:

- सर्वर को शुरू करें और “`server.xml` में `my-web.war` को जोड़ना” से अधिक विस्तृत त्रुटि संदेशों के लिए `console.log` या `messages.log` की जांच करें।
- “एप्लिकेशन में कोई मॉड्यूल नहीं है” या विशेषता से संबंधित त्रुटियों के लिए संकेतों की तलाश करें।

#### 5. एक सरल `server.xml` के साथ परीक्षण करें:

- एक ज्ञात काम करने वाले `pom.xml` का उपयोग करके एक एकल `server.xml` फ़ाइल के साथ एक बुनियादी `server.xml` बनाएं और इसे `server.xml` पर डिप्लॉय करें। यदि यह काम करता है, तो समस्या आपके मूल `server.xml` पैकेजिंग से विशेष है।

## सबसे संभव कारण

जिस समय आपका निर्माण सुलभ रहा, लेकिन IntelliJ “pom.xml में कोई मॉड्यूल नहीं है” या application.xml गायब/गलत है. यह तब हो सकता है जब आपका pom.xml मॉड्यूलों को शामिल करने के लिए सही तरीके से परिभाषित नहीं करता, हालांकि IntelliJ ने कोई त्रुटि नहीं दर्ज की। अन्यथा, यदि IntelliJ सही है, तो समस्या डिप्लॉयमेंट (जैसे, server.xml में कॉन्फिगर नहीं) या गायब सर्वर विशेषताओं से हो सकती है।

## अंतिम सिफारिश

- IntelliJ सामग्री (jar tf myApp.ear) और pom.xml कॉन्फिगरेशन की जांच से शुरू करें।
- यदि IntelliJ खाली है या application.xml से वंचित है, तो IntelliJ सेटअप को ठीक करें और पुनः निर्माण करें।
- यदि IntelliJ सही लगता है, तो इसे server.xml में कॉन्फिगर करें, सुनिश्चित करें कि आवश्यक विशेषताएं सक्रिय हैं, और पुनः डिप्लॉय करें।
- अतिरिक्त संदर्भ के लिए सर्वर लॉग की जांच करें।

इन कदमों का पालन करके, आप यह पहचान सकते हैं कि समस्या निर्माण, डिप्लॉयमेंट या सर्वर कॉन्फिगरेशन में है और IntelliJ पर आपकी एप्लिकेशन को सफलतापूर्वक चलाने में मदद मिलेगी। मुझे किसी विशेष कॉन्फिगरेशन में मदद की आवश्यकता हो तो मुझे बताएं!