

BeanDefinitionStoreException: Invalid bean definition का अनुवाद

दिखाए गए त्रुटि एक `BeanDefinitionStoreException: Invalid bean definition` है, जो `org.springframework.beans.factory` से है, जो यह संकेत देता है कि आपके `application.properties` या `application.yml` में एक बीन की परिभाषा में कोई समस्या है। यह आमतौर पर `application.properties` या `application.yml` की शुरुआत के दौरान होता है जब `Spring` बीन परिभाषाओं को लोड और पर्स करने की कोशिश करता है लेकिन किसी समस्या से जूझता है, जैसे कि गलत संरचना, अनुपलब्ध निर्भरताओं या संरचना फ़ाइलों में वाक्य रचना त्रुटियाँ।

यहाँ एक चरण-दर-चरण मार्गदर्शिका है, जो `BeanDefinitionStoreException` के ज्ञान के आधार पर इस समस्या को खोजने और इसे ठीक करने के लिए है:

1. त्रुटि को समझें

- त्रुटि `org.springframework.beans.factory.BeanDefinitionStoreException` में होती है, जो विशेष रूप से “`Invalid bean definition`” कहता है।
- स्टैक ट्रेस दिखाता है कि त्रुटि `PlaceholderConfigurerSupport` या संबंधित क्लासों में शुरू होती है, जो आमतौर पर गुणस्थान रखने के लिए उपयोग किए जाते हैं (जैसे कि `@Value` अनुप्रयोग या `<context:property-placeholder>`).
- यह सुझाव देता है कि किसी गुण फ़ाइल, बीन परिभाषा (जैसे कि `@Configuration`, या अनुप्रयोग) या अनुपलब्ध निर्भरता में समस्या हो सकती है।

2. अपने संरचना को जांचें

- गुण फ़ाइलें:** अगर आप गुणस्थान रखने का उपयोग कर रहे हैं (जैसे कि `${property.name}`), तो सुनिश्चित करें:
 - गुण फ़ाइल (जैसे कि `application.properties` या `application.yml`) सही स्थान पर मौजूद है (जैसे कि `src/main/resources`).
 - बीन परिभाषा में संदर्भित गुण फ़ाइल में मौजूद है।
 - गुण फ़ाइल में कोई टाइपो या वाक्य रचना त्रुटियाँ नहीं हैं।
- बीन परिभाषाएं:**
 - अगर `@Configuration` संरचना का उपयोग कर रहे हैं, तो टाइपो, अनुपलब्ध या गलत बीन परिभाषाओं, या गलत नामस्थान घोषणाओं के लिए जांच करें।
 - अगर `@Configuration` आधारित संरचना का उपयोग कर रहे हैं (`@Configuration`), तो सुनिश्चित करें कि `@Bean` विधियाँ सही तरह से परिभाषित हैं, और कोई चक्रवत निर्भरताएं या अनुपलब्ध निर्भरताएं नहीं हैं।
 - अगर अनुप्रयोग जैसे `@Component`, `@Service`, आदि का उपयोग कर रहे हैं, तो सुनिश्चित करें कि पैकेज सही तरह से `@ComponentScan` के साथ स्कैन किए जाते हैं।
- निर्भरताएं:** सुनिश्चित करें कि सभी आवश्यक निर्भरताएं (जैसे कि आपके `pom.xml` के लिए `dependencies` या `build.gradle` के लिए `dependencies`) मौजूद हैं और आपके `Spring` संस्करण के साथ संगत हैं।

3. सामान्य कारण और ठीक करने

- अनुपलब्ध या गलत संरचित गुण फ़ाइल:**
 - सुनिश्चित करें कि आपके `application.properties` या `application.yml` सही तरह से संरचित और लोड किए गए हैं। उदाहरण के लिए, अगर `application.properties` का उपयोग कर रहे हैं, तो सुनिश्चित करें कि फ़ाइल `src/main/resources` में है।

- अगर `context:property-placeholder` में `<context:property-placeholder>` का उपयोग कर रहे हैं, तो सुनिश्चित करें कि `location` गुण सही फ़ाइल (जैसे कि `classpath:application.properties`) पर इशारा करता है।

❑ गलत बीन परिभाषा:

- बीन नाम, क्लास नाम, या विधि नाम में टाइपो के लिए जांच करें।
- सुनिश्चित करें कि बीन परिभाषाओं में संदर्भित सभी क्लास क्लासपाथ पर मौजूद हैं और सही तरह से अनुप्रयुक्त हैं (जैसे कि `@Component`, `@Service`, आदि).

❑ चक्रवत निर्भरताएं:

- अगर दो या अधिक बीन एक-दूसरे पर निर्भर हैं, तो `ApplicationContext` उन्हें शुरू करने में विफल हो सकता है। किसी निर्भरता पर `@Lazy` का उपयोग करें या अपने कोड को पुनः संरचित करें ताकि चक्रवत संदर्भों से बचा जा सके।

❑ संस्करण असंगतता:

- सुनिश्चित करें कि आपके `Spring` संस्करण और अन्य निर्भरताएं (जैसे कि `JUnit`, `JUnit4` संस्करण) संगत हैं। स्टैक ट्रेस में `1.8.0_432` दिखाया गया है, तो सुनिश्चित करें कि आपके `Spring` संस्करण इस `1.8.0_432` संस्करण का समर्थन करता है।

4. स्टैक ट्रेस को जांचें

- ❑ स्टैक ट्रेस में उल्लिखित क्लासों जैसे कि `PropertySourcesPlaceholderConfigurer` या `ContextLoader` को देखें। ये `ApplicationContext` के संरचना शुरू करने और गुण हल करने का हिस्सा हैं।
- ❑ त्रुटि किसी बीन परिभाषा में अनुपलब्ध या गलत गुण से हो सकती है, तो किसी `@Value("${property}")` अनुप्रयोग या `Properties` गुणों को जांचें।

5. डिबग करने के चरण

- ❑ **डिबग लॉगिंग को सक्षम करें:** अपने `application.properties` या लॉग संरचना में निम्नलिखित जोड़ें: `logging.level.org.springframework=DEBUG` यह अधिक विस्तृत लॉग प्रदान करेगा ताकि ठीक बीन या गुण को पहचानने में मदद मिल सके।
- ❑ **संरचना फ़ाइलों को सत्यापित करें:** किसी `Properties` जैसे कि `application.properties` या `context.xml` का उपयोग करें ताकि अपने `Properties` या `Context` संरचना फ़ाइलों में त्रुटियों को सत्यापित करें।
- ❑ **क्लासपाथ को जांचें:** सुनिश्चित करें कि सभी आवश्यक क्लास, जार, और संसाधन क्लासपाथ पर हैं।

6. उदाहरण ठीक करने

- ❑ **अगर `@Value` अनुप्रयोग का उपयोग कर रहे हैं:** सुनिश्चित करें कि गुण `application.properties` में मौजूद है: `properties`
`admin.email=admin@example.com` और बीन सही तरह से परिभाषित है: `java`

```

@Value("${admin.email}")
private String adminEmail;

```
- ❑ **अगर `context:property-placeholder` का उपयोग कर रहे हैं:** सुनिश्चित करें कि आपका `Context` संरचना वैध है, उदाहरण के लिए: `xml`

```

<context:property-placeholder location="classpath:application.properties"/>
<bean id="myBean" class="com.example.MyClass">
  <property name="email" value="${admin.email}"/>
</bean>

```

गुण फ़ाइल और बीन क्लास मौजूद हैं।

7. परिवर्तन के बाद परीक्षण करें

- परिवर्तन करने के बाद अपने एप्लिकेशन को पुनः शुरू करें और देखें कि त्रुटि अभी भी मौजूद है।
- अगर समस्या अभी भी हल नहीं हुई है, तो लॉग या स्टैक ट्रेस में अधिक विशिष्ट त्रुटि संदेशों को देखें।

8. अगर आपको और मदद चाहिए

अगर आप अधिक विवरण प्रदान कर सकते हैं—जैसे कि विशेष बीन परिभाषा, गुण फ़ाइल सामग्री, या `xxxxxx` संरचना—तो अधिक लक्षित सलाह दे सकता हूँ। अन्यथा, मैं वेब पर अधिक संसाधनों या उदाहरणों के लिए खोज कर सकता हूँ अगर आप चाहते हैं कि मैं आगे मदद करूँ। क्या आप चाहते हैं कि मैं अधिक जानकारी खोजे या आपके संरचना के विशिष्ट हिस्सों के साथ मदद करे?