

🔧 M1 Mac에서 KonlpyTextSplitter Java 런타임 트러블슈팅

작성일: 2025-09-18 작성자: Jay

1. 문제 설명

M1 Mac (arm64 아키텍처)에서 LangChain 프로젝트에서 **KonlpyTextSplitter**를 초기화하려 할 때, 다음과 같은 오류가 발생

Konlpy의 **Kkma** 태거가 **Java** 런타임을 필요로 하기 때문이지만, 시스템에서 이를 찾지 못하는 상태

오류 메시지

```
CalledProcessError: Command '['/usr/libexec/java_home']' returned non-zero exit status 1.
```

- 의미: 전체 traceback은 Konlpy의 내부 JVM 초기화가 Java 경로를 찾지 못해 실패
- 이 문제는 pyenv로 관리되는 가상 환경에서 M1 Mac에서 흔히 발생
- 원인: **Homebrew**로 설치된 Java를 시스템이 제대로 인식하지 못함

2. 환경 세부 사항

- OS: macOS (M1 arm64)
- Python 버전: 3.13 (**pyenv** 가상 환경 **lc_env**를 통해)
- 관련 패키지: konlpy, langchain-text-splitters
- 초기 설정: Homebrew 설치됨, 하지만 Java가 완전히 등록되지 않음.

3. 근본 원인 분석

- Konlpy의 Kkma 태거는 JPyype를 통해 Java Virtual Machine (JVM)을 시작함
- 초기화 과정에서 **/usr/libexec/java_home**을 호출하여 Java 설치 경로를 찾음
- M1 Mac에서 Homebrew는 OpenJDK를 **/opt/homebrew/opt/openjdk@17**에 설치하지만, 이 경로가 **/Library/Java/JavaVirtualMachines/**에 자동 등록되지 않아 **java_home**이 실패.
- 결과: subprocess로부터 CalledProcessError가 발생하여 ImportError와 유사한 동작을 보임

4. 해결 단계

해결: 호환되는 Java 버전 설치, symlink를 통한 등록, 환경 변수 설정, Konlpy 재설치

아래는 VS Code 터미널에서 lc_env가 활성화된 상태로 실행한 단계별 과정

Step 1: Homebrew를 통해 OpenJDK 설치

arm64 호환 Java 버전 (openjdk@17 추천, Konlpy 호환성 높음)을 설치함.

```
brew install openjdk@17
```

출력 예시:

```
==> Downloading
https://ghcr.io/v2/homebrew/core/openjdk/17/manifests/17.0.16

#####
100.0%
==> Fetching openjdk@17
==> Downloading
https://ghcr.io/v2/homebrew/core/openjdk/17/blobs/sha256:eb27427

#####
100.0%
==> Pouring openjdk@17--17.0.16.arm64_sequoia.bottle.tar.gz
==> Caveats
For the system Java wrappers to find this JDK, symlink it with
sudo ln -sfn /opt/homebrew/opt/openjdk@17/libexec/openjdk.jdk
/Library/Java/JavaVirtualMachines/openjdk-17.jdk

openjdk@17 is keg-only, which means it was not symlinked into
/opt/homebrew,
because this is an alternate version of another formula.

If you need to have openjdk@17 first in your PATH, run:
echo 'export PATH="/opt/homebrew/opt/openjdk@17/bin:$PATH"' >>
~/.zshrc

For compilers to find openjdk@17 you may need to set:
export CPPFLAGS="-I/opt/homebrew/opt/openjdk@17/include"
==> Summary
🍺 /opt/homebrew/Cellar/openjdk@17/17.0.16: 636 files, 305MB
```

Step 2: 기존 symlink 제거 (충돌 방지)

```
sudo rm -rf /Library/Java/JavaVirtualMachines/openjdk-17.jdk
```

Step 3: Java 등록을 위한 symlink 생성

brew caveat를 따라 설치된 Java를 등록함

```
sudo ln -sfn /opt/homebrew/opt/openjdk@17/libexec/openjdk.jdk
/Library/Java/JavaVirtualMachines/openjdk-17.jdk
```

Step 4: JAVA_HOME과 PATH 환경 변수 설정

현재 세션에서 임시 설정:

```
export JAVA_HOME=/Library/Java/JavaVirtualMachines/openjdk-17.jdk/Contents/Home
export PATH="$JAVA_HOME/bin:$PATH"
```

~/zshrc에 추가하여 영구 설정:

```
echo 'export JAVA_HOME=/Library/Java/JavaVirtualMachines/openjdk-17.jdk/Contents/Home' >> ~/.zshrc
echo 'export PATH="$JAVA_HOME/bin:$PATH"' >> ~/.zshrc
source ~/.zshrc
```

확인:

```
echo $JAVA_HOME
which java
java -version
/usr/libexec/java_home -V
```

예상 출력 (java -version):

```
openjdk version "17.0.16" 2025-07-15
OpenJDK Runtime Environment Homebrew (build 17.0.16+0)
OpenJDK 64-Bit Server VM Homebrew (build 17.0.16+0, mixed mode, sharing)
```

Step 5: Konlpy 재설치 및 테스트

Java가 준비되었으니 Konlpy를 재설치하기

```
pip uninstall -y konlpy
pip install --no-cache-dir konlpy
```

Jupyter 노트북 셀에서 테스트 (커널 재시작 후):

```
from langchain_text_splitters import KonlpyTextSplitter

text_splitter = KonlpyTextSplitter()          # 기본 초기화 (Kkma 호출 테스트)
```

```
test_text = "이 문장을 한국어로 분할해 보세요. Konlpy가 제대로 동작하나요?"
chunks = text_splitter.split_text(test_text)
print(chunks)                                     # 형태소/문장 단위 리스트 출력 확
인
```

커스텀 설정 적용:

```
text_splitter = KonlpyTextSplitter(
    chunk_size=200,
    chunk_overlap=0,
)

# 긴 텍스트 분할 테스트
long_text = "여기에 긴 한국어 문서를 넣으세요..."
chunks = text_splitter.split_text(long_text)
print(len(chunks), chunks[:3])
```

5. 추가 팁

- **대체 태거:** Kkma에 여전히 문제가 있으면 Mecab 사용 (Java 불필요, 더 빠름).
 - 설치: `brew install mecab mecab-ko mecab-ko-dic; pip install konlpy`
 - 코드: `text_splitter = KonlpyTextSplitter(tag="mecab")`
- **정리:** 확인 후 불필요한 Java는 `brew uninstall openjdk@17`으로 제거 가능.
- **일반적인 함정:** VS Code 커널이 `lc_env`를 사용 중인지 확인 (Python: Select Interpreter → `lc_env`). 여러 Java 버전 충돌 시 이전 버전 제거 `brew uninstall openjdk@xx`.
- **참조:** Konlpy 문서의 JVM 설정, Homebrew OpenJDK caveats.

이 트러블슈팅은 Java 런타임 위치 오류를 해결하여 `lc_env` 가상 환경에서 KonlpyTextSplitter를 정상적으로 작동하게 함.