USER GUIDE

Table of Contents

AI Services > CLOVA Studio > CLOVA Studio 오픈 소스 연동

LangChain 연동

Classic/VPC 환경에서 이용 가능합니다.

CLOVA Studio의 HyperCLOVA X 모델과 임베딩 도구를 손쉽게 연동하여 사용하기 위해 LangChain을 활용할 수 있습니다.
LangChain은 언어 모델 기반 애플리케이션 개발을 지원하는 오픈소스 프레임워크입니다. HyperCLOVA X를 포함한 여러 언어 모델과 벡터 데이터베이스, 검색 엔진 등의 여러 도구를 사슬(Chain)처럼 엮어 연결할 수 있어 기능 간 연결 및 통합 개발 과정을 간소화할 수 있습니다. 따라서 CLOVA Studio를 서드 파티 모델 및 도구와 함께 이용할 경우, LangChain을 연동하여 좀 더 간편한 애플리케이션 개발이 가능합니다.
LangChain 연동 가이드에서는 LangChain 설치 방법과 CLOVA Studio 연동 설정 방법을 설명합니다. 또한 LangChain을 통해 CLOVA Studio의 HyperCLOVA X 모델과 임베딩 도구를 이용하는 예제 코드를 제공하여 실제 개발에 참고할 수 있도록 안내합니다.

참고

- LangChain은 Python 또는 JavaScript 및 TypeScript 언어로 구현되어 있습니다. CLOVA Studio에서는 Python 기반의 LangChain을 지원하며, 이 가이드 역시 Python을 기준으로 설명합니다.
- LangChain은 LangChain Inc.의 상표입니다. 상표의 권리는 LangChain Inc.에 있습니다. 네이버클라우드는 이 가이드에서 참조 목적으로만 사용하며, 이는 LangChain과 네이버클라우드 간 후원, 보증, 제휴를 의미하지 않습니다.
- LangChain은 오픈 소스 소프트웨어로서 네이버클라우드는 LangChain의 품질, 성능을 보증하거나 책임지지 않습니다. LangChain에 대한 자세한 설명은 공식 문서를 참조해 주십시오.

LangChain 설치 및 확인

CLOVA Studio에 연동하여 LangChain을 사용하려면 버전 3.9 이상의 Python 설치가 필요합니다. Python 설치를 완료한 후에는 다음 명령어를 사용하여 LangChain을 설치한 후 연동에 필요한 `langchain-community` 패키지를 설치해 주십시오.

Bash | Copy

```
pip install langchain # install LangChain
pip install langchain-community # install langchain-community package
```

설치되어 있는 LangChain이 CLOVA Studio와 연동 가능한 버전인지 확인하려면 다음 명령어를 사용해 주십시오.

Bash	Copy

```
pip show langchain-community
```

버전 확인 결과 `langchain-community` 패키지 버전이 0.3.3 이하인 경우에는 다음과 같이 연동 가능한 LangChain 버전을 명시하여 설치하는 것이 필요할 수 있습니다.

Bash	Copy

```
pip install langchain-community~=0.3.4
```

연동 범위 확인

LangChain을 통해 이용 가능한 CLOVA Studio의 기능은 다음과 같습니다.

- CLOVA Studio **플레이그라운드** 메뉴의 **챗 모드** 모델
  - 모델: 기본 모델(<예시> HCX-003), 기본 모델을 튜닝한 모델
    - 연관 API: Chat Completions
  - 연동 예제: HyperCLOVA X 모델 이용
- CLOVA Studio **익스플로러** 메뉴의 임베딩(또는 임베딩 v2) 도구
  - 모델: clir-emb-dolphin, clir-sts-dolphin, bge-m3
    - 연관 API: 임베딩, 임베딩v2
  - 연동 예제: 임베딩 도구 이용

연동 설정

LangChain을 통해 CLOVA Studio의 기능을 안전하게 사용할 수 있도록 API Key와 API Gateway Key를 환경 변수로 등록하여 API 호출 시 필요한 인증 정보로 사용합니다. API Key와 API Gateway Key는 CLOVA Studio의 **플레이그라운드**, **익스플로러** 메뉴에서 테스트 앱 또는 서비스 앱을 생성하여 확인할 수 있습니다. 서비스 앱은 실제 서비스에 적용하는 경우에 신청하여 생성합니다.

> **참고**
>
> CLOVA Studio의 오른쪽 상단에 있는 사용자명을 클릭한 후 **앱 신청 현황** 메뉴를 클릭하여 나타나는 화면에서 테스트 앱(서비스 앱)의 **[자세히]** 버튼을 클릭하면 해당 앱의 API Key와 API Gateway Key를 확인할 수 있습니다. 확인 방법에 대한 자세한 설명과 테스트 앱(서비스 앱) 신청 방법은 플레이그라운드 작업 관리를 참조해 주십시오.

API Key와 API Gateway Key를 환경 변수로 등록하는 방법은 다음과 같습니다.

- 터미널을 통해 등록

```bash
export NCP_CLOVASTUDIO_API_KEY="<NCP-CLOVASTUDIO-API-KEY>"
export NCP_APIGW_API_KEY="<NCP-APIGW-API-KEY>"
```

- Python을 통해 등록

```python
import getpass
import os


os.environ["NCP_CLOVASTUDIO_API_KEY"] = getpass.getpass(
        "NCP CLOVA Studio API Key 입력: "
    )
os.environ["NCP_APIGW_API_KEY"] = getpass.getpass(
        "NCP API Gateway API key 입력: "
    )
```

연동 예제

LangChain과 CLOVA Studio 연동 예제 코드를 소개합니다.

- LangChain을 통해 CLOVA Studio의 HyperCLOVA X 모델을 이용하는 예제
- LangChain을 통해 CLOVA Studio의 임베딩 도구를 이용하는 예제

HyperCLOVA X 모델 이용

LangChain을 통해 CLOVA Studio의 HyperCLOVA X 모델을 이용하는 예제 코드는 다음과 같습니다.

```python
from langchain_community.chat_models import ChatClovaX

chat = ChatClovaX(
    model="HCX-003" # 테스트 앱 또는 서비스 앱 인증 정보에 해당하는 모델명 입력 (기본값: HCX-003)
)
```

`ChatClovaX` 클래스의 인스턴스 정의 시 사용할 수 있는 파라미터에 대한 설명은 다음과 같습니다.

필드	타입	필수 여부	설명

`model`	String	Optional	모델 이름 • **플레이그라운드** 메뉴의 **챗 모드** 기본 모델 | 튜닝 모델 • **플레이그라운드** 메뉴의 **챗 모드** 기본 모델: `HCX-003` (기본값) • 튜닝 모델: 입력 불필요
`task_id`	String	Optional	학습 아이디 • 튜닝 모델을 사용하려는 경우 • 튜닝 모델 테스트 앱(서비스 앱)의 `taskId` 입력
`service_app`	Boolean	Optional	서비스 앱 사용 여부 • `true` | `false` (기본값) • `true` : 서비스 앱 사용 • `false` : 서비스 앱 사용 안 함
`timeout`	Integer	Optional	타임아웃(초) • 1~N (기본값: 90)
`max_retries`	Integer	Optional	재시도 처리 횟수 • 2~N (기본값: 2) • 오류 발생 시 자동으로 재시도 처리
`max_tokens`	Integer	Optional	최대 생성 토큰 수 • Chat Completions의 `maxTokens` 참조
`temperature`	Float	Optional	생성 토큰에 대한 다양성 정도 • Chat Completions의 `temperature` 참조
`top_k`	Integer	Optional	생성 토큰 후보군에서 확률이 높은 K개를 후보로 지정하여 샘플링 • Chat Completions의 `topK` 참조
`top_p`	Float	Optional	생성 토큰 후보군을 누적 확률을 기반으로 샘플링 • Chat Completions의 `topP` 참조
`repeat_penalty`	Float	Optional	같은 토큰을 생성하는 것에 대한 패널티 정도 • Chat Completions의 `repeatPenalty` 참조
`stop_before`	List[String]	Optional	토큰 생성 중단 문자 • Chat Completions의 `stopBefore` 참조
`include_ai_filters`	Boolean	Optional	AI 필터 결과 표시 여부 • Chat Completions의 `includeAiFilters` 참조
`seed`	Integer	Optional	모델 반복 실행 시 결괏값의 일관성 수준 조정 • Chat Completions의 `seed` 참조
`api_key`	String	Optional	API Key • 테스트 앱(서비스 앱)의 `X-NCP-CLOVASTUDIO-API-KEY` 값 • 사전 설정 후 값이 변경된 경우, 입력 필요
`apigw_api_key`	String	Optional	API Gateway Key • 테스트 앱(서비스 앱)의 `X-NCP-APIGW-API-KEY` 값 • 사전 설정 후 값이 변경된 경우, 입력 필요
`base_url`	String	Optional	CLOVA Studio 공통 요청 API URL • 기본값: `https://clovastudio.stream.ntruss.com`

LangChain의 Chat model 클래스 메서드인 `invoke`, `stream` 등을 통해 챗 모델을 실행할 수 있습니다. 예제 코드는 다음과 같습니다.

- `invoke`

Python Copy

```python
messages = [
    (
        "system",
        "CLOVA Studio는 HyperCLOVA X 언어 모델을 활용하여 AI 서비스를 손쉽게 만들 수 있는 개발 도구입니다.",
    ),
    ("human", "CLOVA Studio가 무엇인가요?"),
]
ai_msg = chat.invoke(messages)
ai_msg
```

- `stream`

Python Copy

```python
messages = [
    (
        "system",
        "CLOVA Studio는 HyperCLOVA X 언어 모델을 활용하여 AI 서비스를 손쉽게 만들 수 있는 개발 도구입니다.",
    ),
    ("human", "CLOVA Studio가 무엇인가요?"),
]

for chunk in chat.stream(messages):
    print(chunk.content, end="", flush=True)
```


참고

LangChain을 통해 CLOVA Studio의 챗 모델을 이용하는 방법에 대한 자세한 설명은 공식 문서를 참조해 주십시오.

임베딩 도구 이용

LangChain을 통해 CLOVA Studio의 임베딩(임베딩 v2) 도구를 이용하려면 우선 해당 테스트 앱(서비스 앱)의 App ID를 환경 변수로 등록해야 합니다. 등록하는 방법은 다음과 같습니다.

- 터미널을 통해 등록

```bash
export NCP_CLOVASTUDIO_APP_ID="<테스트/서비스 App ID>"
```

- Python을 통해 등록

```python
import os

os.environ["NCP_CLOVASTUDIO_APP_ID"] = input("테스트/서비스 App ID 입력: ")
```

환경 변수 등록 후 LangChain을 통해 CLOVA Studio의 임베딩(임베딩 v2) 도구를 이용하는 예제 코드는 다음과 같습니다.

```python
from langchain_community.embeddings import ClovaXEmbeddings

embeddings = ClovaXEmbeddings(
    model="clir-emb-dolphin",  # 이용할 테스트 앱 또는 서비스 앱에 해당하는 모델 (기본값: clir-emb-dolphin)
    # app_id="..."  # App ID를 환경 변수로 설정하지 않고 이용하려면 직접 입력
)
```


`ClovaXEmbeddings` 클래스의 인스턴스 정의 시 사용할 수 있는 파라미터는 다음과 같습니다.

필드	타입	필수 여부	설명
`model`	String	Optional	모델 이름 • **익스플로러** 메뉴의 임베딩(임베딩 v2) 모델 • `clir-emb-dolphin` (기본값) | `clir-sts-dolphin` | `bge-m3`
`app_id`	String	Optional	앱 아이디 • 테스트 앱 또는 서비스 앱의 App ID • 환경 변수 미설정 또는 다른 값 이용 시 입력 필요
`service_app`	Boolean	Optional	서비스 앱 사용 여부 • `true` | `false` • `true` : 서비스 앱 사용 • `false` (기본값): 서비스 앱 사용 안 함
`api_key`	String	Optional	API Key • 테스트 앱(서비스 앱)의 `X-NCP-CLOVASTUDIO-API-KEY` 값 • 사전 설정 후 값이 변경된 경우, 입력 필요
`apigw_api_key`	String	Optional	API Gateway Key • 테스트 앱(서비스 앱)의 `X-NCP-APIGW-API-KEY` 값 • 사전 설정 후 값이 변경된 경우, 입력 필요
`timeout`	Integer	Optional	타임아웃(초) • 1~N (기본값: 90)
`base_url`	String	Optional	CLOVA Studio 공통 요청 API URL • 기본값: `https://clovastudio.stream.ntruss.com`

LangChain의 Embedding model 클래스 메서드인 `embed_query` , `embed_documents` 등을 통해 임베딩 도구를 실행할 수 있습니다. 예제 코드는 다음과 같습니다.

- `embed_query`

Python Copy

```python
query = "CLOVA Studio는 HyperCLOVA X 언어 모델을 활용하여 AI 서비스를 손쉽게 만들 수 있는 개발 도구입니다."
single_vector = embeddings.embed_query(query)
```

- `embed_documents`

Python Copy

```python
text1 = "CLOVA Studio는 HyperCLOVA X 언어 모델을 활용하여 AI 서비스를 손쉽게 만들 수 있는 개발 도구입니다."
text2 = "LangChain은 언어 모델 기반 애플리케이션 개발을 지원하는 오픈소스 프레임워크입니다."
document = [text1, text2]
multiple_vector = embeddings.embed_documents(document)
```

참고

LangChain을 통해 CLOVA Studio의 임베딩(임베딩 v2) 도구를 이용하는 방법에 대한 자세한 설명은 공식 문서를 참조해 주십시오.