Google Gemini Embedding API - Timeout 및 Illegal Metadata 오류 트러블슈팅

> 작성일: 2025-09-19
>
> 작성자: Jay

---

1. 문제 상황

1.1. `503 Illegal metadata`, `60초 타임아웃 초과`

`LangChain`의 `SemanticChunker`와 `GoogleGenerativeAIEmbeddings`를 사용하여 `gemini-embedding` 모델을 호출할 때, 다음 오류가 반복 발생:

```
503 Illegal metadata
Timeout of 60.0s exceeded
```

1.2. `NameError`

`chunks` 변수가 생성되지 않아 아래와 같이 `NameError`가 발생:

```
NameError: name 'chunks' is not defined
```

1.3. `gRPC에서의 충돌`

오류 로그에는 `gRPC 레이어`에서 `Illegal metadata` 에러와 함께 재시도 실패까지 기록.

---

2. 문제 원인 분석

- `GoogleGenerativeAIEmbeddings` 내부에서 API 호출 시, 메타데이터 헤더가 올바르지 않아 서버가 요청을 거부함.
- 긴 입력 텍스트(`file`)를 한 번에 임베딩하려 해 타임아웃 발생.
- API 키가 잘못되었거나 누락되었을 수도 있음.
- 네트워크(VPN, 프록시) 문제 또는 라이브러리 버전 불일치 가능성.

---

3. 해결 과정

3.1. `API 키` 확인 및 재설정

- 노트북 환경에서 환경변수(.env) 잘 로드됐는지 출력해 확인

```
    import os
    from dotenv import load_dotenv

    load_dotenv()
    print("GOOGLE_API_KEY:", os.getenv("GOOGLE_API_KEY")[:10] + "...")
# API가 유효합니다
```

3.2. 라이브러리 버전 재설치 및 업그레이드

```
    pip uninstall -y google-generativeai langchain-google-genai google-ai-
generativelanguage
    pip install -U google-generativeai langchain-google-genai
```

3.3. Google SDK 직접 호출 테스트

```
    import google.generativeai as genai
    import os

    genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))

    result = genai.embed_content(
        model="models/gemini-embedding-001",
        content="What is the meaning of life?",
        task_type="retrieval_document"
    )

    print(result['embedding'][:10])
```

- 위 테스트가 성공하면 LangChain 래퍼 문제, 실패하면 키 혹은 네트워크 문제 있는 것
- 성공!

3.4. LangChain 코드용으로 수정 및 배치 분할 적용

긴 문서를 한꺼번에 호출하지 말고 미리 큰 청크로 분할하여 임베딩 수행

```
    from langchain_experimental.text_splitter import SemanticChunker
    from langchain_google_genai import GoogleGenerativeAIEmbeddings
    from langchain_text_splitters import RecursiveCharacterTextSplitter
    from dotenv import load_dotenv
```

```
    import os

    load_dotenv()

    if not os.getenv("GOOGLE_API_KEY"):
        os.environ["GOOGLE_API_KEY"] = input("Enter your Google API key:
")

    embeddings = GoogleGenerativeAIEmbeddings(
        model="models/gemini-embedding-001",
        task_type="retrieval_document",
        google_api_key=os.getenv("GOOGLE_API_KEY")
    )

    text_splitter = SemanticChunker(embeddings)

    # 앞의 오류가 타임아웃이었으므로 배치, 청크 사이즈 제한 (안전하게 설정)
    batch_splitter = RecursiveCharacterTextSplitter(
        chunk_size=100000,
        chunk_overlap=0
        )

    batches = batch_splitter.split_text(file)                              #
file: 긴 텍스트

    all_chunks = []

    for batch in batches:
        chunks = text_splitter.split_text(batch)
        all_chunks.extend(chunks)

    print(all_chunks)
```

3.5. 네트워크 환경 점검과 타임아웃 대비

- VPN, 프록시 없이 재시도
- 필요 시 임베딩 클라이언트 타임아웃 조정 (LangChain 소스 수정 필요)

3.6. 추가 권장

- 필요 시 라이브러리 버전 하향(예: `google-generativeai==0.5.4`, `langchain-google-genai==0.0.5`)도 시도
- 장기적으론 `Google 공식 SDK 최신 버전`과 `LangChain 통합 버전` 확인 필수

---

4. 요약

> `Google Gemini Embedding`의 `gRPC 호출` 시 `메타데이터 불일치` + `긴 텍스트 임베딩`으로 인한 `타임아웃` 문제에서 비롯

> API 키 재확인, 라이브러리 재설치, 긴 텍스트 배치 분할 호출 적용으로 문제를 성공적으로 해결

---

5. 참고

- Google Generative AI Embeddings 공식 문서
- LangChain GoogleGenAI 통합 문서
- Google Cloud API 인증 관련 기본 가이드