
- 출처: LangChain 공식 문서 또는 해당 교재명
- 원본 URL: https://smith.langchain.com/hub/teddynote/summary-stuff-documents

---

`WebBaseLoader`

- **`WebBaseLoader`** = 웹 기반 문서를 로드하는 로더
- **`bs4` 라이브러리 사용 → 웹 페이지 파싱**
  - `bs4.SoupStrainer` 사용 → 파싱할 요소 지정
  - `bs_kwargs` 매개변수 사용 ← `bs4.SoupStrainer` 의 추가적인 인수 지정

- `API 공식 도큐먼트` 참고

```
# API KEY를 환경변수로 관리하기 위한 설정 파일
import os
from dotenv import load_dotenv

# API KEY 정보로드
load_dotenv()               # true
```

- 사전에 `VS Code` 터미널에 설치할 것

```
        pip install beautifulsoup4
```

```
import bs4                                                                  # BeautifulSoup 임포트
from langchain_community.document_loaders import WebBaseLoader          # WebBaseLoader 임포트

# 뉴스기사 내용 로드하기
loader = WebBaseLoader(
    web_paths=("https://n.news.naver.com/article/437/0000378416",),  # 웹 페이지 경로
    bs_kwargs=dict(
        parse_only=bs4.SoupStrainer(
            "div",                                                  # 파싱할 HTML 요소
            attrs={"class": ["newsct_article _article_body", "media_end_head_title"]},  # 파싱할 요소의 클래스
        )
    ),
    header_template={
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0
    },
)

# 문서 로드
docs = loader.load()

# 문서의 수 출력
print(f"문서의 수: {len(docs)}")

# 문서 출력
docs
```

- 셀 출력 (1.9s)

```
USER_AGENT environment variable not set, consider setting it to identify your requests.
문서의 수: 1
```

```
[Document(metadata={'source': 'https://n.news.naver.com/article/437/0000378416'}, page_content="\n출산 직원에게 '1억원
```

- 출력된 문서(`Docs`) `Data Wrangler`에서 열어본 화면

- `SSL` 인증 오류를 우회하기 위해 **`verify`** 옵션을 설정할 수 있음

```
# ssl 인증 우회
loader.requests_kwargs = {"verify": False}              # SSL 인증서 검증하지 않도록 설정

# 데이터 로드
docs = loader.load()
```

- 셀 출력

```
/Users/jay/.pyenv/versions/lc_env/lib/python3.13/site-packages/urllib3/connectionpool.py:1097: InsecureRequestWar
warnings.warn(
```

- 경고 메시지 해석 = **`InsecureRequestWarning`**
  - **`HTTPS`** 요청이 검증되지 않은 상태로 진행되고 있음을 경고
  - **`SSL`** **인증서를 우회했기 때문에 발생하는 경고** = 이 경고는 보안상의 이유로 발생하며, SSL 인증서를 검증하는 것을 권고

- 여러 웹피이지를 한 번에 로드 가능
  - **`urls`** 리스트 = 로더에 전달 → 전달된 `urls`의 순서대로 문서 리스트 반환

```
from langchain_community.document_loaders import WebBaseLoader
import bs4

# WebBaseLoader 인스턴스 생성
loader = WebBaseLoader(
    web_paths=[
        "https://n.news.naver.com/article/437/0000378416",                                          # 첫 번째 웹 페이지 경로
        "https://n.news.naver.com/mnews/hotissue/article/092/0002340014?type=series&cid=2000063",   # 두 번째 웹 페이지 경로
    ],
    bs_kwargs=dict(
        parse_only=bs4.SoupStrainer(
            "div",                                                                                  # 파싱할 HTML 요소
            attrs={"class": ["newsct_article _article_body", "media_end_head_title"]},              # 파싱할 요소의 클래스
        )
    ),
    header_template={
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.
    },
)

# 데이터 로드
docs = loader.load()

# 문서 수 확인
print(len(docs))          # 2 (셀 출력: 0.4s)
```

- 웹에서 가져온 결과 출력하기

```
print(docs[0].page_content[:500])
print("\n", "===" * 10, "\n")
print(docs[1].page_content[:500])
```

- 셀 출력

```
출산 직원에게 '1억원' 쏜다…회사의 파격적 저출생 정책


[앵커]올해 아이 낳을 계획이 있는 가족이라면 솔깃할 소식입니다. 정부가 저출생 대책으로 매달 주는 부모 급여, 0세 아이는 100만원으로 올렸습니다. 여기
```

```
==============================


고속 성장하는 스타트업엔 레드팀이 필요하다


[이균성의 溫技] 초심, 본질을 잃을 때한 스타트업 창업자와 최근 점심을 같이 했다. 조언을 구할 게 있다고 했다. 당장 급한 현안이 있는 건 아니었다. 여
```

- **여러 `URL` 동시에 스크래핑** → 스크래핑 과정 가속화
- 동시 요청 시 제한: **기본값 = 초당 2회**
- 스크래핑 서버 제어가 필요한 경우
  - `requests_per_second` 매개변수 변경 → 최대 동시 요청 수 늘릴 수 있음
  - 단, 이 방법은 스크래핑 속도를 높일 수는 있지만 서버로부터 차단될 수 있으므로 주의가 필요

```
# nest_asyncio: 비동기 이벤트 루프를 중첩하여 사용할 수 있는 라이브러리
# Jupyter 노트북 환경에서 비동기 코드 실행 시 유용
import nest_asyncio             # nest_asyncio 임포트

nest_asyncio.apply()            # nest_asyncio 적용하기
```

```
# 초당 요청 수 설정
loader.requests_per_second = 1

# 비동기 로드
docs = loader.aload()
```

- 셀 출력 (0.4s)

```
/var/folders/h3/l7wnkv352kqftv0t8ctl2ld40000gn/T/ipykernel_28587/2225555786.py:5: LangChainDeprecationWarning: Se

docs = loader.aload()
Fetching pages: 100%|##########| 2/2 [00:00<00:00,  5.19it/s]
```

```
# 결과 출력
docs
```

- 셀 출력

```
[Document(metadata={'source': 'https://n.news.naver.com/article/437/0000378416'}, page_content="\n출산 직원에게 '1억원
Document(metadata={'source': 'https://n.news.naver.com/mnews/hotissue/article/092/0002340014?type=series&cid=2000
```

프록시 사용

- `IP` **차단 우회를 위해** 때때로 사용 필요한 경우가 있음
- 프록시 사용 시: **로더(및 그 아래의 `requests`)에 프록시 딕셔너리 전달 가능**

```
from langchain_community.document_loaders import WebBaseLoader             # WebBaseLoader 임포트

# WebBaseLoader 인스턴스 생성
# 웹 기반 로더 초기화
loader = WebBaseLoader(
    "https://www.google.com/search?q=parrots",                              # 웹 페이지 경로

    # 프록시 설정
    proxies={
        "http": "http://{username}:{password}:@proxy.service.com:6666/",    # HTTP 프록시 설정
```

```
        "https": "https://{username}:{password}:@proxy.service.com:6666/",  # HTTPS 프록시 설정
    },
)

# 문서 로드
docs = loader.load()
```

- 프록시 설정 예시

```
from langchain_community.document_loaders import WebBaseLoader

# WebBaseLoader 인스턴스 생성
loader = WebBaseLoader(
    "https://www.google.com/search?q=parrots",

    # 프록시 설정
    proxies={
        # 실제 프록시 정보를 여기에 입력해야 함
        # 본인의 실제 정보로 교체할 부분: {YOUR_USERNAME}, {YOUR_PASSWORD}, {YOUR_PROXY_HOST}, {YOUR_PROXY_PORT}
        # {YOUR_USERNAME} = 실제 사용자 이름
        # {YOUR_PASSWORD} = 실제 비밀번호
        # {YOUR_PROXY_HOST} = 실제 프록시 주소
        # {YOUR_PROXY_PORT}를 실제 포트 번호
        # 프록시 서버에 사용자 이름과 비밀번호가 필요하지 않은 경우에는 {YOUR_USERNAME}:{YOUR_PASSWORD}@ 부분을 통째로 삭제할 것
        "http": "http://{YOUR_USERNAME}:{YOUR_PASSWORD}@{YOUR_PROXY_HOST}:{YOUR_PROXY_PORT}/",
        "https": "https://{YOUR_USERNAME}:{YOUR_PASSWORD}@{YOUR_PROXY_HOST}:{YOUR_PROXY_PORT}/",
    },
)

# 문서 로드
# 프록시 서버를 통해 웹 페이지에 접속을 시도합니다.
try:
    docs = loader.load()
    print("문서가 성공적으로 로드되었습니다.")
    # 로드된 문서 내용 일부를 출력해봅니다.
    print(docs[0].page_content[:500])
except Exception as e:
    print(f"오류가 발생했습니다: {e}")
    print("\n프록시 설정이 올바른지 다시 확인해 주세요.")
```

- 프록시 정보 찾는 방법
  - `브라우저` 설정: 크롬, 파이어폭스 등 사용 중인 브라우저의 네트워크 또는 프록시 설정 메뉴를 직접 확인하기
  - `운영 체제` 설정: 윈도우, macOS, 리눅스 등 운영 체제의 네트워크 설정에서 프록시 설정 확인하기
  - `네트워크 환경`: 회사나 학교 등 특정 네트워크 환경에 접속되어 있다면, 해당 네트워크 관리자에게 문의하여 프록시 서버 정보 요청하기

- *next: 텍스트(`TextLoader`)*
