


# LangSmith




# 1. LangSmith 개요





LangSmith는 LangChain에서 개발한 LLM(대규모 언어 모델) 애플리케이션의 개발, 디버깅, 테스트, 평가 및 모니터링을 위한 통합 플랫폼입니다.

LangChain이 LLM 애플리케이션을 구축하는 프레임워크 역할을 한다면, LangSmith는 구축된 애플리케이션이 프로덕션 환경에서 견고하고 효율적으로 작동하도록 돕는 LLMOps(LLM Operations) 도구에 가깝습니다.



## LangSmith 개요

LangChain 생태계에서 LangChain으로 구축한 애플리케이션(LLM 체인, 에이전트, RAG 등)을 **모니터링·평가·운영**할 수 있게 설계된 SaaS(혹은 엔터프라이즈) 플랫폼이다.

아래에서 목적, 주요 기능, 구조/설정, 장점·주의사항, 실습용 체크리스트 순으로 정리한다.

---

### 1. 목적

- LLM 기반 애플리케이션이 복잡해짐에 따라, 단순 호출 로그만으로는 충분한 인사이트를 얻기 어렵다.

LangSmith는 다음 목적을 갖는다:

- 호출(trace) 데이터 수집 및 시각화 → 체인 나비 흐름, 입력 → 출력 → 오류 흐름 등.
  - 평가(evaluation) 메커니즘 제공 → 정량화된 메트릭(예: 정확도, 응답시간, 품질) 생성.
  - 릴리즈·운영 단계에서 애플리케이션의 품질·안정성 확보 및 팀 협업자 간 피드백 루프 형성.
  - 즉 “개발 → 배포 → 운영” 사이클을 LLM 애플리케이션에도 적용할 수 있게 한다.
-

## 2. 주요 기능

다음은 LangSmith가 제공하는 기능들이다. (공개된 정보 중심으로 요약한다)

| 기능명            | 설명  |
|----------------|---|
| 호출 추적(Tracing) | LangChain 내부 체인(LangChain Chain), 에이전트(Agent) 실행 과정에서 입력(prompt), 중간 결과(intermediate steps), 최종 출력, 오류 등을 수집한다. 예컨대 체인 내 여러 블록이 어떻게 연결되었는지가 시각화 된다. |
| 평가(Evaluation) | 미리 정의된 평가 데이터세트(예: 기대 응답 vs 실제 응답)를 통해 모델/체인의 품질을 측정할 수 있다. 품질 점수, 지연시간, 실패율 등을 집계할 수 있다.   |
| 대시보드 및 시각화     | 팀이 볼 수 있는 웹 UI로 호출 흐름, 지연시간 히스토그램, 오류 빈도수, 사용자별 세션 등 다양한 지표를 한눈에 볼 수 있다.  |
| 프로젝트/팀 관리      | 여러 프로젝트(예: dev, prod), 여러 사용자(팀원), 권한 설정 등이 가능할 수 있다.   |
| 통합 및 API 키     | LangChain SDK나 콜백(handler)을 통해 trace 및 평가 데이터를 자동으로 전송 가능하다. 환경변수 설정이나 SDK 설정이 필요하다.  |
| 릴리즈 모니터링       | 새 버전 릴리즈 시 과거 버전 대비 성능 변화, 오류 변화 등을 분석할 수 있다.   |

## LangSmith API Key 발급

1. <https://smith.langchain.com/> 으로 접속하여 회원가입을 진행합니다.
2. 가입후 이메일 인증하는 절차를 진행해야 합니다.
3. 왼쪽 톱니바퀴(Setting) - 가운데 "Personal" - "Create API Key" 를 눌러 API 키를 발급 받습니다.

The screenshot shows the LangSmith 'Personal' settings page. A sidebar on the left contains a settings gear icon at the bottom, which is highlighted with a red circle and arrow labeled '1'. The main content area is titled 'Personal' and includes a 'Create API Key' button, which is highlighted with a red circle and arrow labeled '3'. Below this button is a message about 'Personal Access Tokens' being converted to 'Service Keys', with a red circle and arrow labeled '2' pointing to the 'Personal' tab in the filter section. At the bottom, a table lists two API keys with their descriptions, creation times, and delete actions.

Personal > Settings

Personal

API Keys

You can only access an API key when you first create it. If you lost one, you will need to create a new one.

Personal Access Tokens are available now! All previous API keys have been converted to Service Keys. You can find more information [here](#)

Personal Service All

2 keys

| Key        | Description | Created At              | Actions |
|------------|-------------|-------------------------|---------|
| ██████████ | ██████████  | 2024. 6. 17. 오전 2:41:03 | 🗑️      |
| ██████████ | ██████████  | 2024. 6. 13. 오전 8:10:54 | 🗑️      |

Description 에 본인이 알 수 있는 설명을 넣고 Create API Key 버튼을 클릭하여 생성합니다.

Access an API key when you first create it. If you lost one, you will need to create a new one.

### Create an API Key

Description

테스트키

Key Type

☒ Personal Access Token

Used for authenticating with the API as an individual user.

☐ Service Key

Tied to a service principal for automation and CI workflows.

Create API Key


### 3. 구조 및 설정

강의 교재용으로, 설정 및 코드 예제를 포함해 설명한다. (사용하는 언어는 Python, 그리고 최신 LangChain Core 모듈 경로 기준)

#### 3.1 사전 준비

- API 키 발급: LangSmith 가입 → 프로젝트 생성 → API KEY 확보
- 환경변수 설정 (예)

bash

 코드 복사

```
export LANGSMITH_API_KEY="YOUR_API_KEY"
export LANGCHAIN_TRACING_V2="true"
export LANGCHAIN_PROJECT="my-project-name"
```



### 3.2 LangChain 코드 내 통합

python

 코드 복사

```
from langchain_core.callbacks import LangSmithCallbackHandler
from langchain_core.chat_models import ChatOpenAI
from langchain_core import LLMChain, PromptTemplate

# 콜백 핸들러 설정
langsmith_handler = LangSmithCallbackHandler(
    api_key = None,          # env var로 자동 취득 가능
    project_name = "my-project-name",
    dataset_name = "dev-eval"
)

# 체인 예시
template = "안녕, {name}님! 오늘 기분은 어떠신가요?"
prompt = PromptTemplate.from_template(template)
llm = ChatOpenAI(model_name="gpt-4o-mini", temperature=0.7, callbacks=[langsmith_handler])
```

```
chain = LLMChain(llm=llm, prompt=prompt)
```

```
# 실행
```

```
result = chain.run(name="김철수")
```

```
print(result)
```

위 코드에서 `LangSmithCallbackHandler`가 주요 트레이스 데이터를 수집하고, LangSmith 웹 대시보드로 전송한다.

### 3.3 평가(Evaluation) 세팅

python

📄 코드 복사

```
from langchain_core.evaluation import LangSmithEvalChain

# 기대 응답을 갖는 입력/출력 데이터 준비
data = [
    {"input": "서울의 날씨 알려줘", "expected": "현재 서울의 날씨는 ..."},
    {"input": "파이썬에서 리스트 뒤집는 법 알려줘", "expected": "list[::-1]을 사용하면..."},
]

eval_chain = LangSmithEvalChain(
    project="my-project-name",
    dataset="list_reverse_eval"
)

results = eval_chain.evaluate(data)
print(results)
```

이렇게 하면 각 케이스별로 평가 결과가 LangSmith에 저장된다. 이후 대시보드에서 지표로 활용 가능하다.

## 4. 장점 및 주의사항

### 장점

- Dev → Prod까지 전체 사이클을 추적할 수 있어 책임성과 가시성이 높아진다.
- 팀 단위로 모니터링·피드백 루프 구축하기 좋다.
- 모델 교체나 체인 구조 변경 후 영향을 시각적으로 분석 가능하다.

### 주의사항

- 비용 및 라이선스: 일부 기능은 엔터프라이즈 라이선스 필요하다는 언급이 있다. Reddit +1
- 데이터 프라이버시 및 국내 법률(한국 GDPR/개인정보보호법 등) 고려 필요: 외부 SaaS로 로그 전송 시 민감정보가 포함되지 않도록 설계해야 한다.
- 설정 누락 시 트레이스가 제대로 작동하지 않는다는 커뮤니티 언급이 있다. Reddit
- 일부 기능(특히 셀프호스팅 버전) 정보가 공개적으로는 제한적이다.



감사합니다