

LLM 기반 HTTP 미들웨어 생성 시스템

1. 시스템 개요

- 사용자의 자연어 요청을 바탕으로
- HTTP 요청을 처리하는 미들웨어 함수를 자동 생성하고,
- 이를 테스트 및 개선하는 시스템입니다.

이 시스템은 **개발 효율성**과 **코드 품질 향상**을 목표로 하며, 자연어 처리와 LLM(Antropic Claude)을 활용하여 요구사항 분석, 코드 생성, 검증, 개선 프로세스를 자동화합니다.

2. 핵심 컴포넌트

2.1 자연어 처리기 (Natural Language Parser)

- 기능:** 사용자의 자연어 입력을 분석하여 의도(intent), 주요 개체(entities), 요구사항(requirements), 제약조건(constraints)을 추출합니다.
- 예시 입력:**
"러시아에서 오는 요청을 차단해줘"
- 출력:**

```
{  
  "intent": "block_request",  
  "entities": ["Russia"],  
  "requirements": ["Block requests from Russia"],  
  "constraints": [],  
  "parameters": {"country": "Russia"}  
}
```

2.2 HTTP 요청 분석기 (Request Analyzer)

- 기능:** 샘플 HTTP 요청 데이터를 분석하여 주요 특성을 추출합니다.
- 분석 대상:**
 - HTTP Method (GET, POST 등)
 - Headers (Content-Type, Authorization 등)
 - Body 및 Query Parameters
 - 요청의 Path 및 IP 주소
- 활용 사례:**
 - CORS 정책 검증
 - JWT 토큰 유효성 검사
 - URL 파라미터 정제

2.3 함수 생성기 (Function Generator)

- 기능: 분석된 요구사항을 기반으로 Python으로 작성된 HTTP 미들웨어 코드를 자동 생성합니다.
- 지원 기능:
 - IP 기반 필터링
 - 콘텐츠 필터링
 - 요청 변환 및 헤더 수정
 - 로깅 및 캐싱
- 예시 코드:

```
**def** middleware(request):  
    **if** request.headers.get("X-Country") == "Russia":  
        **return** Response("Blocked", status=403)  
    **return** next(request)
```

2.4 테스트 엔진 (Test Engine)

- 기능: 생성된 함수의 동작을 검증하며 성능 및 정확도를 측정합니다.
- 테스트 방식:
 - 더미 HTTP 요청 시나리오 생성
 - 코드 실행 결과와 요구사항의 일치 여부 확인
- 검증 프로세스:
 - ValidationAgent가 피드백 제공(예: 보안 문제, 성능 개선 필요 사항 등)

2.5 문서화 에이전트 (Documentation Agent)

- 기능: 생성된 코드에 대한 자동 문서화를 수행하여 가독성과 유지보수성을 높입니다.
- 출력 예시:

```
# 이 미들웨어는 러시아에서 오는 모든 요청을 차단합니다.# 사용 예시:#  
request.headers["X-Country"] = "Russia"
```

3. 작동 프로세스

1. 사용자가 자연어로 요청 입력 (예: "러시아에서 오는 요청을 차단해줘").
2. ParsingAgent가 입력 내용을 분석하여 JSON 형태의 요구사항 도출.
3. MiddlewareAgent가 요구사항에 맞는 Python 코드를 자동 생성.
4. SampleRequestAgent가 샘플 HTTP 요청 시나리오를 생성.
5. ValidationAgent가 생성된 코드를 검증하고 피드백 제공.
6. DocumentationAgent가 코드에 대한 문서를 작성.
7. 필요 시 개선된 코드와 문서를 추가적으로 생성.

4. 주요 기능 요약

요구사항 분석

- 사용자의 자연어 입력을 구조화된 JSON 객체로 변환.

코드 생성

- Python 기반의 미들웨어 함수 자동 생성.

코드 검증

- ValidationAgent를 통해 보안, 성능, 요구사항 충족 여부를 검증.

개선 프로세스

- 검증 피드백을 바탕으로 개선된 코드와 문서를 생성.

문서화

- DocumentationAgent를 활용한 자동화된 코드 주석 및 설명 작성.

5. 데이터 저장 및 관리

SQLite 데이터베이스 구조 (`middleware_history`)

Column Name	Type	Description
<code>id</code>	INTEGER	고유 식별자
<code>timestamp</code>	TEXT	작업 시간
<code>input_text</code>	TEXT	사용자가 입력한 자연어 요청
<code>requirements</code>	TEXT	분석된 요구사항(JSON)
<code>initial_code</code>	TEXT	최초로 생성된 미들웨어 코드
<code>initial_documentation</code>	TEXT	최초로 작성된 문서
<code>validation</code>	TEXT	검증 결과
<code>improved_code</code>	TEXT	개선된 코드
<code>improved_documentation</code>	TEXT	개선된 문서