

7주차(1)_Nginx, Flask

Flask란?

Framework는 소프트웨어의 목적에 따라 효율적으로 코드의 구조를 짜는 개발 방식으로,

- 코드 관리가 체계적으로 가능하므로 유지보수가 용이
- 기본적인 설계 및 기능 라이브러리를 제공하므로 개발 생산성이 높음
- 코드의 재사용성이 높음
- 추상화된 코드를 제공하여 확장성이 좋음

위의 이유 때문에 사용하고 있다.

Flask는 웹 어플리케이션 개발을 위해 사용하는 **Web Framework** 중 하나로, Python을 사용한다. Web Framework는 웹 페이지, 어플리케이션 등의 웹 서비스의 개발을 보조하기 위한 어플리케이션 프레임워크를 말한다. 대표적으로 Flask와 Django가 있다.

Flask는 가볍게 배울 수 있고, 가볍게 사용할 수 있고, 가볍게 배포할 수 있는 장점이 있지만, Django에 비해서 기능이 적기 때문에 복잡한 어플리케이션을 만들 때, 설정해줄 것이 많다. 보통, 초보자&가벼운 어플리케이션은 Flask를, 복잡한 어플리케이션에는 Django를 사용한다.

Rest API란?

통상적으로 백엔드 개발은 데이터베이스 설계나 REST API 개발을 의미한다고 생각하면 된다.

특히, Rest(ful) API 서버는 **HTTP 메소드**를 이용해서 자원(database)을 조작하고 상태를 주고 받는다.

GET(조회), POST(추가), PUT/PATCH(수정), DELETE(삭제)



ex) 마켓컬리의 상품 데이터베이스

상품 목록 조회 – GET (list)

특정 상품에 대한 정보 회의 – GET (retrieve)

상품 추가 – POST

상품 정보 수정 – PUT(update) /PATCH(partial update)

이때, 서버(백엔드)와 브라우저(프론트엔드)는 **JSON 형식**으로 서로 통신한다. 예를 들어 프론트에서 버튼을 눌러서 데이터에 대한 작업 요청이 발생하면, 백엔드에서는 해당 응답을 JSON(.json) 형식으로 해준다. JSON은 파이썬의 dictionary처럼 key-value형식으로 구성된 text를 말한다.

Nginx란?

Web Server 소프트웨어로, 해당 실습에서는 클라이언트(웹 브라우저)에서 요청을 받으면, 요청에 맞는 응답을 보내는 HTTP Web Server로 활용한다.

Flask를 사용해 웹사이트(어플리케이션)을 만들었다면, 해당 웹사이트를 웹서버에 올려서 사용자가 웹사이트를 사용할 수 있도록 설정해야한다. 이러한 역할이 웹 서버 소프트웨어인 Nginx가 하는 일이다! 따라서 둘을 연동해야만 우리가 흔히 생각하는 웹사이트가 완성된다.

여기서는 프론트엔드로 사용되는 Flask와 백엔드로 사용되는 Nginx 둘을 연결해줄 미들웨어로 uWSGI를 사용하는데, 미들웨어는 프론트엔드와 백엔드 사이의 통신 프로토콜을 담당해준다고 생각하면 된다.