먼저 REST란 REresentational State Transfer의 약자로 자원을 URI로 식별하고 해당 자원의 상태를 HTTP로 주고받는 리소스 중심의 설계 원칙이다. REST에서는 리소스를 URI(Uniform Resource Identifier)로 표현하고 HTTP 메서드(POST, GET, PUT, DELETE, PATCH 등)로 조작한다. REST는 다음의 3가지로 구성이 되어있다.

1. 자원(HTTP URI)
2. 자원에 대한 행위(HTTP Method)
3. 자원에 대한 행위의 내용(HTTP Message Pay Load).

REST API란 HTTP를 기반으로 REST 원칙을 따르는 API이다. REST 설계 원칙으로는 다음과 같은 것들이 있다.

1. URI는 동사보다는 명사를, 대문자보다는 소문자를 사용해야 한다
2. 마지막에 슬래시 /를 포함하지 않는다
3. 언더바 \_ 대신 하이픈 – 사용
4. 파일 확장자는 URI에 포함하지 않는다
5. URI에 행위(동작)를 포함하지 않는다

다음으로 HTTP란 웹에서 클라이언트와 서버가 데이터를 주고받을 때 사용하는 프로토콜이다. 요청(Request)과 응답(Response)이라는 방식으로 동작한다. HTTP 프로토콜은 무상태성과 비연결성이라는 특징이 있어 연결을 유지하지 않기 때문에 일반적인 경우에서 최소한의 자원으로 빠른 응답을 가능하게 한다.

HTTP 메시지는 시작 라인(start-line), 헤더(header), 공백 라인(empty line, CRLF), 바디(message body)의 4단계로 구성되어 있다.

요청 메시지의 시작 라인은 method SP(공백) request-target SP HTTP-version CRLF(개행)로 구성돼 있으며 method는 HTTP 메서드 종류를, request-target은 리소스의 경로(URL)을 의미한다.

응답 메시지의 시작 라인은 HTTP-version SP status-code SP reason-phrase CRLF 으로 구성돼 있으며 status-code는 HTTP 상태 코드를, reason-phrase는 짧은 코드 설명을 의미한다.

헤더는 field-name ":" OWS(띄어쓰기 허용) field-value OWS 의 구조를 가지고 있다.

CRLF는 시작 라인의 끝, 각 헤더의 줄 끝, 헤더 블록을 종료하면서 끝을 표시하는데 사용된다.

바디는 실제 전송할 데이터들로 이루어져 있으며 HTML 문서, 이미지, 영상, JSON 등 byte로 표현할 수 있는 모든 데이터들이 담길 수 있다.

브라우저에 URL을 입력 후 요청하여 서버에서 응답하는 과정

1. 사용자가 주소창에 URL 입력

URL의 구성 요소

1. 프로토콜: https://
2. 도메인: naever.com
3. 경로: /blog
4. 쿼리/해시: 요청 파라미터, 문서 내부 위치
5. 브라우저가 도메인의 IP 주소 조회

도메인 이름을 실제 서버의 IP 주소로 변환하기 위해 DNS 질의를 함

이 때 다음 순서로 계층 캐시를 조회함

1. 브라우저 캐시
2. OS 캐시
3. /etc/hosts (수동 등록)
4. 라우터 / ISP DNS 서버
5. DNS 재귀 질의 (Root → TLD → 권한 DNS)
6. 서버와의 TCP 연결
7. TCP 연결이 완료되면 브라우저는 HTTP 요청을 전송
8. 웹 서버가 요청 처리 = HTTP 응답 전송
9. 브라우저가 응답을 수신하고 콘텐츠를 렌더링