

What is HTTP

HTTP(Hypertext Transfer Protocol)

HTTP는 클라이언트와 서버가 서로 통신하는 방식을 표준화하는 TCP/IP 기반 애플리케이션 계층 통신 프로토콜

▼ HyperText란

웹 페이지를 다른 페이지로 연결하는 링크 이를 위한 마크업 언어가 바로 HTML이겠다.

| HTML에 a태그라 생각하면된다.

▼ 마크업언어란?

문서나 데이터의 구조를 나타내는 언어

▼ TCP IP란?

TCP/IP는 네트워크 프로토콜 스위트(Suite: 모음곡)로, 온라인상의 안전하고 효율적인 데이터 전송의 필수 요건, 인터넷에서 사용되는 기본 통신 프로토콜

TCP: Transmission Control Protocol(전송 제어 프로토콜)

IP: Internet Protocol (인터넷 프로토콜)

주요 프로토콜	TCP/IP 프로토콜 계층모델	OSI 계층모델
TELNET, FTP, SNMP, DHCP 등	Application	Application
		Presentation
		Session
SCTP, TCP, UDP	Transport	Transport
IGMP ICMP IP ARP	Internet (IP 계층)	Network
	Link (Network Access)	Data Link
		Physical

인터넷을 통해 콘텐츠가 요청되고 전송되는 방식이다.

HTTP/0.9 - 원 라이너(1991)

- GET이라는 단일 메서드만 사용하는 가장 단순한 프로토콜
- HTTP 헤더가 없다

▼ HTTP 헤더란?

HTTP 헤더는 클라이언트와 서버가 요청 또는 응답으로 부가적인 정보를 전송할 수 있게 해줍니다.

▼ 헤더란?

정보기술에서 **헤더(header)**는 저장되거나 전송되는 데이터 블록의 맨앞에 위치한 보충 데이터를 가리킨다. 데이터에서 헤더를 따르는 데이터는 페이로드(payload), 바디(body)로 불리기도 한다.

헤더 활용예시(페이로드)

```

let cart = createSlice({
  name: "cart",
  initialState: [
    { id: 0, name: "White and Black", count: 2 },
    { id: 2, name: "Grey Yordan", count: 1 },
  ],
  reducers: {
    addCount(state, action) {
      let idx = state.findIndex((a) => {
        return a.id == action.payload;
      });
      state[idx].count++;
    },
    addItem(state, action) {
      state.push(action.payload); // payload를 상태에 추가
      console.log(action.payload);
    },
  },
});

```

```

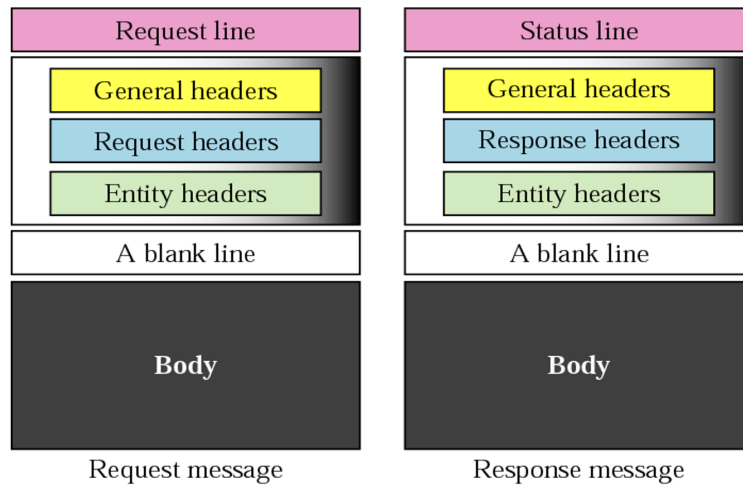
<button
  className="btn btn-danger"
  onClick={() => {
    dispatch(addItem({ id: shoe.id, name: shoe.title, count: 1 }));
  }}
>
// 저오브젝트가 payload라 생각하면 된다.

```

HTTP Header 종류(4가지)

- General header: 요청과 응답 모두 적용되지만 바디에서 최종적으로 전송되는 데이터와는 관련이 없는 헤더
- Request header: 패치 될 리소스나 클라이언트 자체에 대한 자세한 정보를 포함하는 헤더
- Response header: 위치 또는 서버 자체에 대한 정보(이름, 버전 등)와 같이 응답에 대한 부가적인 정보를 갖는 헤더
- Entity header: 콘텐츠 길이나 MIME 타입과 같이 엔티티 바디에 대한 자세한 정보를 포함하는 헤더

Figure 27.8 Headers



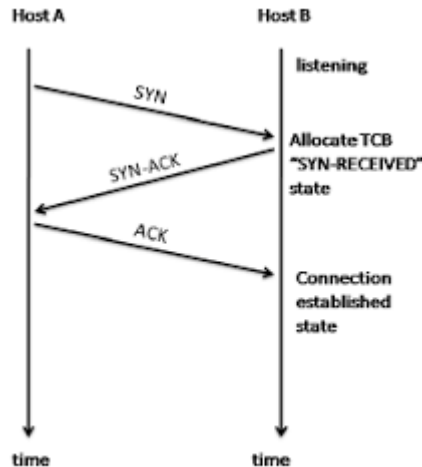
McGraw-Hill

©The McGraw-Hill Companies, Inc., 2004

HTTP/1.0 - (1996)

- 이미지, 비디오, 파일, 일반 텍스트 또는 기타 콘텐츠 유형과 같은 다른 응답 형식도 처리할 수 있게 됨
- 더 많은 메서드 추가 (POST, HEAD)
- HTTP 헤더 추가
- 단점
 - 연결당 여러 요청을 받을 수 없다.
 - Why Bad? → 이미지 10개, 스타일시트 5개, 자바스크립트 파일 5개, 총 20개의 항목이 있는 웹페이지를 방문했을 때 해당 웹페이지에 대한 요청이 이루어질 때 가져와야 하는 항목이 있다고 가정해 봅시다. 서버는 요청이 완료되는 즉시 연결을 닫기 때문에 각 항목이 개별 연결에서 하나씩 제공되는 20개의 개별 연결이 연속적으로 발생합니다. 이렇게 많은 수의 연결이 이루어지면 3자 핸드셰이크와 느린 시작으로 인해 새로운 TCP 연결이 필요하기 때문에 성능에 심각한 타격을 입게 됩니다. (이해못함)

▼ Three handshake란?



TCP 연결시 데이터 전송을 하기 전에 정확한 전송을 보장하기 위해 상대방 컴퓨터와 사전에 세션을 수립하는 과정

쉽게말해, 신뢰확보를 하기위한 초기인증

SYN(Synchronize Sequence Number): Sequence Number를 랜덤으로 설정하여 세션을 연결하는 데 사용, Connection을 생성할 때 사용되는 flag

ACK(Acknowledgement): 응답 확인, 패킷을 받았다는 것을 의미하는 flag

동작방식

1. Client가 Server에게 접속을 요청하는 SYN플래그를 보냄
2. Server는 SYN을 확인하고 SYN+ACK플래그를 Client에게 전송 그후 Server는 ACK를 받기위해 대기상태로 변경
3. SYN + ACK를 받은 Client는 서버에게 ACK보내고 연결성립

HTTP/1.1 - (1997)

- 새로운 메서드 추가(PUT, PATCH, OPTIONS, DELETE)
- 1.0에서는 호스트 헤더가 필요하지 않았지만 1.1에서는 호스트 헤더가 필요로 해짐

▼ Why?

~~HTTP/1.0에서는 연결당 하나의 요청, 요청 완료되면 즉시 닫힘~~

~~HTTP/1.1에서는 영구 연결, 즉 요청 완료되도 열려 있어서 순차적 요청을 허용하는 연결 도입 이때 연결을 닫으려면 닫기 헤더를 사용해야함.~~

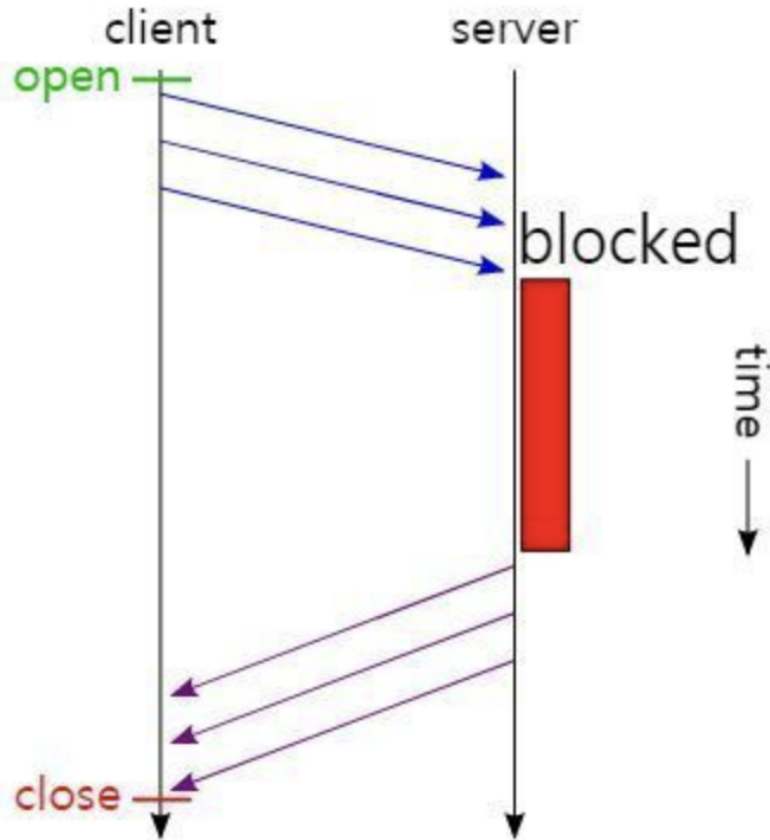
- 클라이언트가 서버 응답을 기다리지 않고 서버에 여러 요청을 보낼 수 있는 파이프라이닝도 지원
 - 하지만 결국 완전한 멀티플렉싱이 아닌 응답처리를 미루는 방식이므로 각 응답의 처리는 순차적으로 처리되며, 결국 후순위의 응답은 지연될 수 밖에 없다.

▼ 파이프라이닝이란?

명령어를 순차적으로 실행하는 프로세스에서 한번에 하나의 명령어만 실행하는 것이 아닌 다른 명령어 또한 실행하면서 동시에 여러 개의 명령어를 실행하는 기법

| 명령어 여러개 실행시키기

- ~~연결 하나당 하나의 요청을 처리 하도록 설계됨~~
- HTTP/1.0 vs HTTP/1.1
<http://www.ra.ethz.ch/cdstore/www8/data/2136/pdf/pd1.pdf>
- 단점
 - HOL(Head of line blocking)
 - 첫번째 요청이 block되면 후 순위 요청들도 block되는 현상



- 무거운 헤더 구조
 - ~~한 번에 여러 요청을 보낼러하나~~ 헤더의 많은 부분이 중복됨

HTTP/2.0 - (2015)

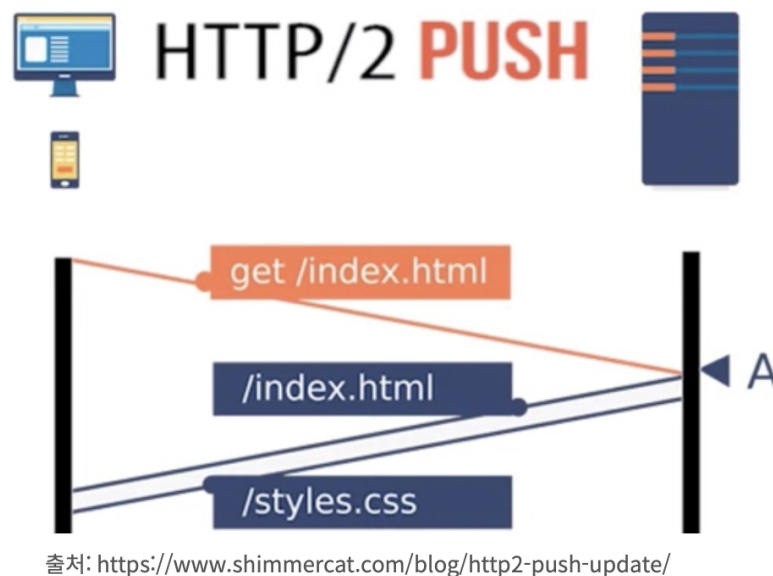
- 텍스트 → 바이너리
 - 더 이상 육안으로 볼 수 없다.
 - 주요구성요소는 프레임과 스트림
 - HTTP는 이제 하나이상의 프레임(메타 데이터를 위한 HEADERS 프레임, 페이로드를 위한 DATA프레임 및 다양한 유형의 프레임 존재) 프레임의 집합을 스트림
- ▼ 메타 데이터란?
 - 데이터에 대한 데이터

| 정보를 효율적으로 찾기 위한 데이터

메타데이터(metadata or **metainformation**)는 데이터(data)에 대한 데이터이다. 이렇게 흔히들 간단히 정의하지만, 캐런 코일(Karen Coyle)에 의하면 '어떤 목적을 가지고 만들어진 데이터(constructed data with a purpose)'라고도 정의한다. 즉, 다른 데이터를 정의하고 기술하는 데이터(data that defines and describes other data)이다. 가령 도서관에서 사용하는 서지기술용으로 만든 것이 그 대표적인 예이다. 지금은 온톨로지의 등장과 함께 기계가 읽고 이해할 수 있는(Machine Actionable) 형태의 메타데이터가 많이 사용되고 있다. 설명 메타데이터, 구조화 메타 데이터로 구분된다.

- 멀티플렉싱
 - TCP 추가 연결 없이 비동기적으로 전송, 서버는 비동기적으로 응답
- ▼ 멀티플렉싱이란?

하나의 통신 채널을 통해서 둘 이상 데이터를 전송하는데 사용되는 기술
- HAPCK을 이용한 헤더압축
- 서버푸시 - 단일 요청에 대한 다중 응답
 - 리소스를 요청할 것을 알고있는 서버가 요청없이 리소스를 푸시하는 것

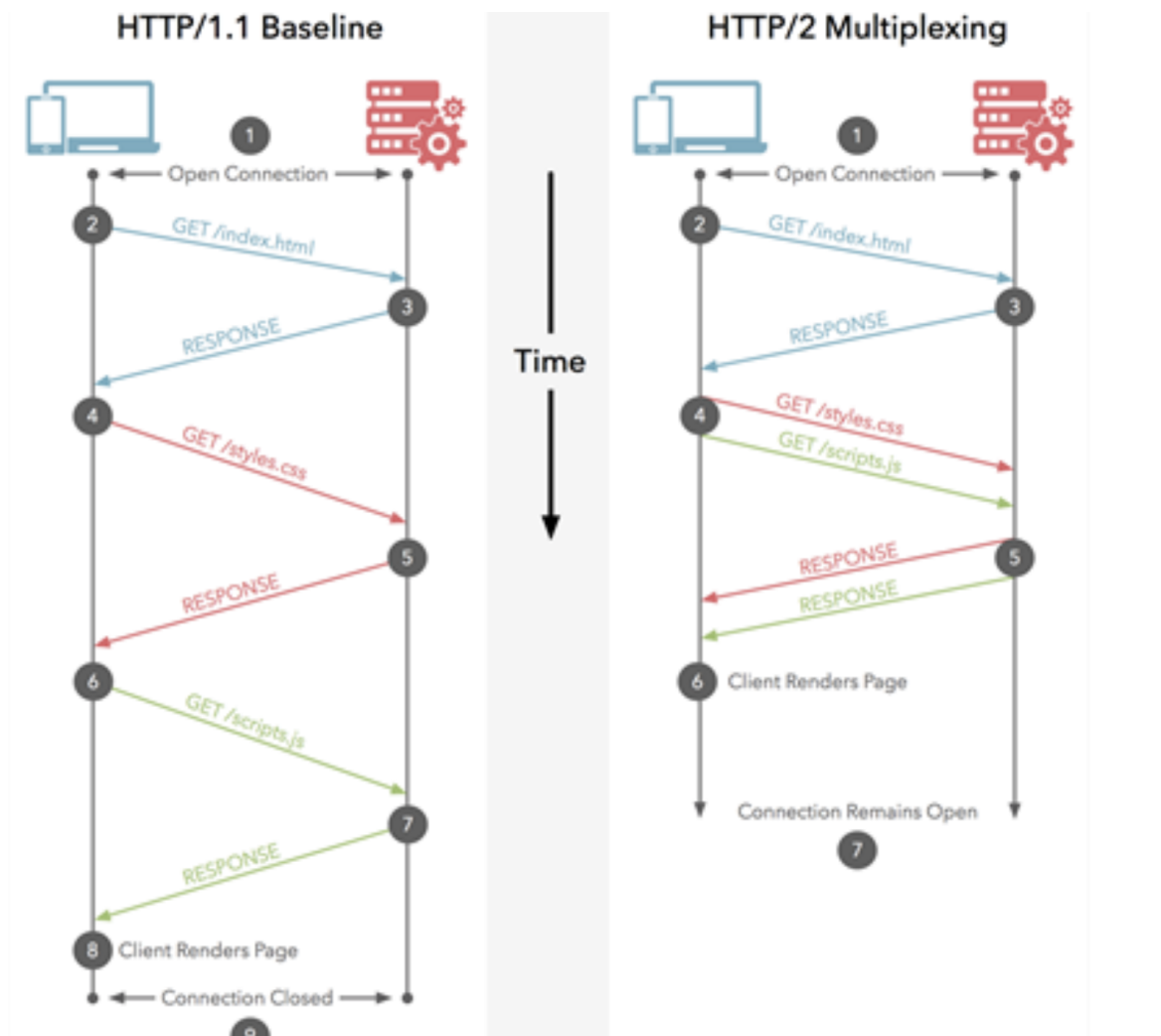


- 요청 우선순위 지정
- 보안
 - TLS의 의무화는 아니지만 벤더(공급자)들은 TLS가 이용될 때만 지원한다고 밝히면서 사실상 의무화

▼ TLS란

Transport Layer Security는 인터넷 상의 커뮤니케이션을 위한 개인 정보와 데이터 보안을 용이하기 위해 사용되는 프로토콜

HTTP/1.1 vs HTTP/2



HTTP1.1 은 먼저 수신된 요청에대한 순서로 응답을 보내야함

오늘나온 질문

서버는 리퀘스트를 보내는가?