BAEK JIWON

웹 개발자라면 알아야 할,

HTTP 진화 과정

TABLE OF CONTENTS

1 HTTP가 무엇인가

정의

- HTTP(HyperText Transfer Protocol)는 World Wide Web 상에서 정보를 주고받을 수 있는 프로토콜이다. 주로 HTML 문서를 주고받는 데에 쓰인다.

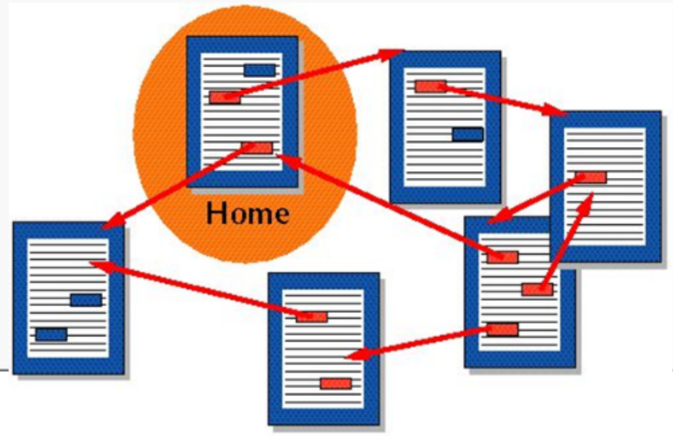
- 위키피디아 -

즉, HTTP는 하이퍼텍스트(HyperText)를 전송하는(Transfer) 프로토콜(Protocol)이고,
전송이 이루어지는 장소는 World Wide Web이다.

1 HTTP가 무엇인가

HyperText

- 다른 문서에 대한 참조를 통해 독자가 한 문서에서 다른 문서로 즉시 접근할 수 있는 텍스트
- HTML 상에서 < a >와 같은 요소를 얘기한다.



Protocol

- 프로토콜은 컴퓨터 내부에서, 또는 컴퓨터 사이에서 데이터의 교환 방식을 정의하는 규칙 체계입니다.

- MDN -

1 HTTP가 무엇인가

World Wide Web

- 월드 와이드 웹은 인터넷에 연결된 컴퓨터를 통해 사람들이 정보를 공유할 수 있는 전세계적인 정보 공간을 말한다. 간단히 웹이라고 부르는 경우가 많다.

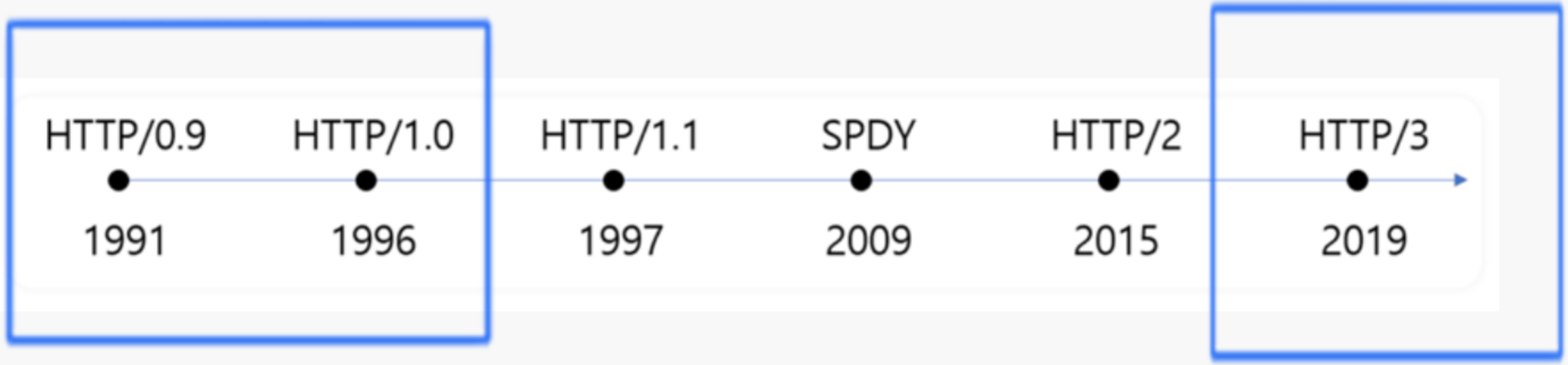
- 위키피디아 -

CERN(유럽 입자 물리 연구소)

팀 버너스리 (컴퓨터 과학자)

1.1 들어가기 앞서...

HTTP의 간략한 역사



표준 이전,
http 초창기 모델

유일한 UDP 기반 통신

1.1 들어가기 앞서...

HTTP 메소드란?

- 클라이언트와 서버 사이에 이뤄지는 요청과 응답 데이터를 전송하는 방식
- HTTP 메서드는 총 9가지가 있으며 주로 GET, POST, PUT, PATCH, DELETE를 사용

메서드 이름	설 명
GET	리스소 조회
POST	등록, 요청 데이터 처리
PUT	리소스 덮어쓰기 (해당 리소스가 없으면 생성)
PATCH	리소스 부분 분경 (PUT은 전체 변경이지만, PATCH는 일부만 변경)
DELETE	리소스 삭제
HEAD	GET과 동일하나 메시지 부분(Body)를 제외하고, 상태 줄과 헤더만 반환
OPTIONS	대상 리소스에 대한 통신 가능 옵션(메서드)을 설명 (주로 CORS에서 사용)
CONNECT	대상 자원으로 식별되는 서버에 대한 터널을 설정
TRACE	대상 리소스에 대한 경로를 따라 메시지 루프백 테스트를 수행

2 HTTP/0.9 - THE FIRST STEP

HTTP/0.9

- 처음부터 붙여진 이름이 아니다.
- 오직 GET 메소드만 존재.
- 응답을 주고 나면 곧바로 연결 해제.

```
GET /mypage.html
```

요청

```
<HTML>
A very simple HTML page
<HTML>
```

응답

3 HTTP/1.0 - EXPANDING CAPABILITIES

HTTP/1.0

- 여전히 표준은 아니지만 0.9 버전의 문제를 해결하고, 기능을 추가하기 위해 개발됨.

HTTP/1.0에 생긴 몇 가지 변화들

1.버저닝
2.상태코드
3.헤더
4.메서드 확장

3 HTTP/1.0 - EXPANDING CAPABILITIES

HTTP/1.0

1. 버저닝

- 요청을 보낼 때마다 버전 정보가 붙는다.

```
GET /mypage.html HTTP/1.0
```

2. 상태코드

- 응답의 시작 부분에 붙어서 전송

```
200 OK
Date : Tue, 15 nov 1994 08:12:31 GMT
...
```

대표적인 예시) 404: 해당 페이지를 찾을 수 없을 때, 요청을 거부하고 싶은 경우

3 HTTP/1.0 - EXPANDING CAPABILITIES

HTTP/1.0

3.헤더

- 요청과 응답에 대한 메타데이터를 전송
- HTML만 지원 가능하던 제한적 상황을 유연하고 확장 가능하게 만들었다.

메타데이터 : 데이터에 관한 구조화된 데이터로, 다른 데이터를 설명해 주는 데이터 [위키피티아]

4.메소드 확장

HTTP / 0.9

GET

\+ HEAD, POST

=>

HTTP / 1.0

GET, HEAD, POST

3 HTTP/1.0 - EXPANDING CAPABILITIES

여전히 남아있는.. HTTP/1.0의 한계

- 요청와 응답 사이에 단 하나의 연결만이 존재한다.
- 응답이 끝나면 연결이 사라지고, 다음 요청을 하기 위해서는 다시 3-way-handshake를 통해서 TCP 커넥션을 생성해주어야 한다.

- 1.0버전은 뭔가 합의를 해서 만들어진 버전이 아니고, 일단 생겨나는 문제들을 뭐든지 해결해보자. 라는 방식으로 접근하며 발전되었기 때문이다.

4 HTTP/1.1 - THE STANDARD PROTOCOL

HTTP/1.1 특징

1. 표준 프로토콜.
2. 이전에 사용된 연결을 다시 열어 시간을 절약.
3. 명령어의 데이터 경로를 세분화하고, 각기 다른 세부 단계를 동시에 수행하게 함으로써, 여러 명령어들을 중첩 수행 가능하게 한다. 이로 인해 첫번째 요청의 응답이 완전히 전송되기 전에 두번째 요청 전송이 가능해지고, 커뮤니케이션 latency를 낮추었다.
4. 뭉텅이진 응답을 나누어 보낼 수 있다.
5. 캐시 제어에 대한 세심한 컨트롤 가능.
6. 클라이언트와 서버로 하여금 교환하기에 가장 적합한 컨텐츠를 보여줄 수 있다.
7. Host 헤더를 지원하여 동일 IP주소에서 다른 도메인을 호스트하는 기능이 가능하게 되었다.
8. GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS로 메서드 확장.

4 HTTP/1.1 - THE STANDARD PROTOCOL

HTTP/1.1의 문제점

1. 헤더의 중복
   a. HTTP/1.1에 다양한 기능이 추가되면서 헤더에도 많은 메타 데이터가 담기게 되었습니다. 하지만 매 요청마다 헤더를 중복해서 전송해야하고, 심지어는 전송하려는 값보다 헤더의 크기가 더 큰 경우도 종종 발생한다는 문제점이 존재했습니다.

2. HOLB(Head-of-Line Blocking) → 낭비 문제..
   a. 무엇보다도 가장 큰 문제. 서버가 항상 요청받은 순서대로 응답해야 하기 때문에 발생하는 문제입니다. HTTP/1.1에서는 하나의 연결 내에서 응답 다중화(multiplexing) 를 할 수 없었기 때문에 요청이 순차적으로 처리되어야 했는데, 서버가 응답을 작성하던 중간에 문제가 생기면 후속 요청들이 전송되지 못하고 지연되는 문제가 있었습니다.

→ TCP 연결을 여러 개 두는 방식으로 병렬 연결을 구현해 해결하려 하였으나 임시적인 해결책일 뿐, 근본적으로 해결하지 못함

5 HTTP/2 (HTTP OVER SPDY) - IMPROVED PERFORMANCE

HTTP/2 핵심 특징과 변화 6가지

1. 이진 프로토콜 사용

   HTTP/2는 이진(binary) 프로토콜을 사용하여 데이터를 바이너리로 인코딩합니다. 이전 버전인 HTTP/1.1이 텍스트 기반 프로토콜이었던 것과 달리, HTTP/2는 효율적인 데이터 전송을 위해 바이너리 프레이밍을 사용합니다.

2. 응답 다중화 지원

   HTTP/2는 하나의 TCP 연결에서 여러 요청을 동시에 처리할 수 있습니다. 이는 스트림(stream), 메시지(message), 프레임(frame)이라는 단위로 더 세분화되어 구현됩니다. 이 기능 덕분에 HTTP/1.1에서의 HOLB(Head-Of-Line Blocking) 문제를 해결했습니다.

3. HPACK을 통한 헤더 필드 압축

   HTTP/2는 HPACK이라는 허프만 코딩 기법을 사용하여 헤더 필드를 압축합니다. 이를 통해 변경되지 않은 헤더 정보는 재전송하지 않고 오버헤드를 최소화합니다.

5 HTTP/2 (HTTP OVER SPDY) - IMPROVED PERFORMANCE

HTTP/2 핵심 특징과 변화 6가지

4. 서버 푸시(Server Push)

클라이언트의 요청 없이도 서버가 미리 리소스를 보낼 수 있습니다.

5. 스트림 우선순위 지정

정수와 트리 구조를 사용하여 스트림별 우선순위를 지정합니다.

6. 프로토콜 자체의 흐름 제어 기능

네트워크 트래픽 관리 및 제어를 위한 기능을 포함합니다.

6 HTTP/3 (HTTP OVER QUIC) - THE FUTURE STEP

HTTP/3 등장 배경

TCP의 한계, 특히 HOLB(Head-Of-Line Blocking) 문제와 느린 연결 속도를 해결하기 위해 개발.
TCP의 복잡한 기능 대신, UDP 기반의 QUIC 프로토콜을 사용함으로써 성능을 개선합니다.

QUIC?

UDP 기반으로, 신뢰성 있는 데이터 전송을 위해 필요한 기능들을 직접 구현한다.
연결의 신속성과 신뢰성을 향상시키기 위해 0-RTT 및 1-RTT 연결 설정을 제공한다.

6 HTTP/3 (HTTP OVER QUIC) - THE FUTURE STEP

HTTP/3 특징

1. 독립적인 스트림 운영

   각 스트림이 독립적으로 운영되어, 하나의 스트림에 문제가 발생해도 다른 스트림에 영향을 주지 않습니다.

2. 연결 식별 및 유지의 개선

   연결 ID를 이용해 IP가 변경되어도 연결을 유지할 수 있어, 네트워크 환경 변화에 강합니다.

3. 향상된 보안 및 기능

   TLS 연결 설정이 내부적으로 포함되어 있어 보안이 강화되었습니다.

   우선순위 제어, 서버 푸시 등의 기능이 지원됩니다.

4. 헤더 압축 기법의 개선

   QPACK을 사용하여 헤더를 압축하고, QUIC의 독립적인 스트림 속성에 맞게 헤더 압축 기법을 최적화합니다.

5. TCP와의 호환성 및 전환

   기존 HTTP 체계와 호환성을 지니며, 서버가 클라이언트에게 HTTP/3의 지원 여부를 알려주어 통신 방식을 전환할 수 있습니다.

THANK YOU!

감사합니다.