HTTP와 HTTPS

# 1. 프로토콜과 HTTP

# 1.1. 프로토콜

프로토콜이란 개념이 왜 등장했는지 먼저 살펴보자. 두 대의 컴퓨터가 있다. 이 두 대 의 컴퓨터는 필요한 자료들을 공유하고 싶어한다. 그래서 서로 통신을 통해서 자료들을 제공하기로 하였다. 이 때, 정의된 약속같은것이 없다고 생각해보자. 두 컴퓨터는 서로 정보를 주고 받는 과정에서, 어느 데이터가 요청에 대해서 보내온 데이터이고, 어느 데이터가 연결이 제대로 되어 있는지 체크하는 내용인지, 보낸 정보인지, 이런 것들을 판단하는 것은 어려운 일이다. 따라서, 컴퓨터 사이의 통신에 있어서 약속을 정하는데, 이 약속이 곧 프로토콜(Protocol)이다. MDN(Mozilla Developer Network)의 용어 사전에서 프로토콜의 정의를 찾으면 아래와 가다.

*프로토콜은 컴퓨터 내부에서, 또는 컴퓨터 사이에서 데이터의 교환 방식을 정의하는 규칙 체계입니다. 기기간 통신은 교환되는 데이터의 형식에 대해 상호 합의를 요구합니다. 이런 형식을 정의하는 규칙의 집합을 프로토콜이라고 합니다.*

*-MDN 용어사전 프로토콜 항목-*

그렇다면 프로토콜과 HTTP는 무슨 상관일까. HTTP(HyperText Transfer Protocol)는 프로토콜 중의 하나로, 대부분의 경우 HTML문서나 이미지 등을 받아오는데 사용되는 프로토콜이다 [1]. 즉, 우리가 보는 대부분의 웹사이트들은 HTTP로 HTML 문서를 서버에서 가져온 뒤, 이를 웹 브라우저가 렌더링하여 보여주는 형식이다.

그렇다면 HTTP의 정확한 통신 방법은 어떻게 될까? HTTP에서 클라이언트는 다음과 같은 메시지를 서버로 전송한다.

# 1.2 HTTP 프로토콜

GET / HTTP/1.1  
Host: ryanclaire.blogspot.com  
Connection: keep-alive  
DNT: 1  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.75 Safari/537.36   
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9   
Accept-Encoding: gzip, deflate  
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7

클라이언트가 요청하는 HTTP Request Message는 요청(Request), 요청 헤더(Request Header), 빈 줄(Blank Line), 그리고 요청 몸체(Request Body)로 구성된다. 이 중에서 요청 몸체는 존재하지 않아도 된다[2].

이제 한 라인씩 살펴보도록 하자. 노란색 배경으로 표시된 첫째 줄, 요청에는 요청 메소드와 HTTP 버전이 적혀 있다. HTTP 버전은 HTTP 1.0과 HTTP 1.1을 사용할 수 있다. 사용가능한 요청 메소드와 그 역할은 아래 표와 같다[2].

|  |  |
| --- | --- |
| 요청 메소드 | 역할 |
| GET | 클라이언트가 서버에서 웹 리소스를 받아올 때 사용(http 1.0/1.1) |
| HEAD | GET을 통해 받아오는 응답(Response)의 헤더를 받아올 수 있음(http 1.0/1.1) |
| POST | 웹 서버에 데이터를 보낼 때 사용(http 1.0/1.1) |
| PUT | 서버에 데이터 저자을 요청할 때 사용(http 1.1) |
| DELETE | 서버에 데이터 삭제를 요청할 때 사용(http 1.1) |
| TRACE | 수행 작업의 diagnostic trace를 요청할 때 사용(http 1.1) |
| OPTIONS | 지원하는 요청 목록을 받아올 때 사용(http 1.1) |
| CONNECT | 프록시에게 다른 호스트에 연결하고 콘텐츠를 분서갛거나 캐시 없이 응답하도록 지시하는데 사용(http 1.1) |

다음으로 하늘색 배경으로 표시된 줄들은 요청 헤더에 해당한다. 요청 헤더에 들어가느 내용들은 아래 표와 같다[2].

|  |  |
| --- | --- |
| 내용명 | 역할 |
| Host | 도메인 이름 |
| Accept | 클라이언트가 다룰 수 있는 MINE 타입이 열거됨 |
| Accept-Language | 클라이언트가 다룰 수 있고 선호하는 언어 타입이 열거됨 |
| Accept-Charset | 클라이언트가 다룰 수 있고 선호하는 문자 집합이 열거됨(UTF-8 등) |
| Accept-Encoding | 클라이언트가 지원하는 인코딩 타입이 열거됨(gzip, deflate 등) |
| Connection | 요청후 연결을 닫을 것인지 유지할 것인지를 나타냄 |
| User-Agent | 요청이 만들어진 브라우저 타입을 알려줌 |
| Content-Length | POST 요청을 사용할 때, 요청 몸체의 길이를 나타냄 |
| Content-Type | POST 요청을 사용할 때, 요청 몸체 데이터의 형식을 MIME 타입으로 나타냄 |

초록색으로 표시된 부분은 빈 줄로, 요청 헤더와 요청 몸체를 분리한다.

파란색으로 표시된 부분은 요청 몸체 부분으로, 이 예제에서는 생략되어 있다.

이로서 HTTP 프로토콜 상에서 클라이언트가 서버에 요청을 보내는 경우의 메시지 구조에 대해서 알아보았다. HTTP 프로토콜 상에서 서버가 응답으로 보내는 메시지 또한 매우 유사한 구조를 가지고 있다. 서버가 클라이언트에게 보내는 답장 메시지는 상태(Status), 응답 헤더(Response Header), 빈 줄(Blank line), 그리고 응답 몸체(Response Body)로 구성되어 있다. 요청 메시지와 다른점은 응답 몸체는 요청 몸체와는 달리 필수적으로 존재해야 한다는 것이다[2].

아래는 HTTP 프로토콜 상에서 서버가 클라이언트에게 보내는 요청 응답문의 예시이다.

HTTP/1.1 200 OK  
Content-Type: text/html; charset=UTF-8  
Expires: Fri, 16 Oct 2020 03:24:32 GMT  
Date: Fri, 16 Oct 2020 03:24:32 GMT  
Cache-Control: private, max-age=0  
Last-Modified: Thu, 15 Oct 2020 14:47:59 GMT  
ETag: W/"5ff21f5b961238a44cb1cd27f6e753ccd873730fc8c28050213852c3fa960c18"   
Content-Encoding: gzip  
X-Content-Type-Options: nosniff  
X-XSS-Protection: 1; mode=block  
Server: GSE  
Alt-Svc: h3-Q050=":443"; ma=2592000,h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-T051=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"   
Transfer-Encoding: chunked  
<빈 라인>  
--중략--

노란색으로 표시된 부분은 상태(Status)이다. HTTP 버전이 가장 앞에 오고, 그 다음으로 상태 코드, 그 다음으로 해당 상태 코드가 발생한 이유가 따라온다. 각 상태 코드와 해당 코드가 발생한 이유에 대해서는 아래 표에 정리되어 있다[2].

|  |  |
| --- | --- |
| 상태 코드 | 의미 |
| 1XX | Informational. 서버의 상태 정보 |
| 100 Continue | 서버에서 요청을 받았으며, 응답을 보내기 위해 처리하는 중임 |
| 2XX | Success. 요청을 수신하였고, 의미를 이해하였으며, 이를 처리했다는 것을 의미함. |
| 200 OK | 요청을 이행했다. |
| 3XX | Redirectional. 요청을 처리하기 위해 다른 액션이 필요로 함을 의미함 |
| 301 Move Permanently | 요청된 리로스스이 위치가 영구적으로 변경되었음을 의미함 |
| 302 Found & Redirect | 요청된 리소스의 위치가 임시로 변경되었음을 의미함 |
| 4XX | Client Error. 요청에 문법 오류가 있거나 의미를 이해하지 못함을 의미함 |
| 400 Bad Request | 서버에서 요청을 이해하지 못함. 구문 오류의 가능성이 있음을 의미함 |
| 403 Forbidden | 클라이언트의 신원에 상관없이 서버에서 리소스 제공을 거부함을 의미함 |
| 404 Not Found | 요청된 리소스를 서버에서 찾지 못함을 의미함 |
| 405 Method Not Allowed | 클라이언트의 요청을 서버에서 허용하지 않음을 의미함 |
| 5XX | Server Error. 서버에서 요청을 처리하지 못했음을 의미함 |
| 500 Internal Server Error | 서버 내부 프로그램의 오류 |
| 501 Method Not Implemented | 요청 메소드가 잘못되었음을 의미함 |
| 502 Bad Gateway | 프록시 또는 게이트웨이는 업스트림 서버에서 잘못된 응답을 수신했음을 의미함 |
| 503 Service Unavailable | 과부하나 유지 보수로 인해 서버가 응답할 수 없음을 의미함 |

하늘색으로 나타낸 줄은 응답 헤더를 의미하며, 응답 헤더에는 아래의 표에 나타난 내용들이 들어간다.

|  |  |
| --- | --- |
| 내용명 | 의미 |
| Location | 요청 처리 완료 또는 새 리소스 식별을 위해 요청된 URI가 아닌 위치로 Redirection하는데 사용됨 |
| Server | 요청 처리릉 ㅟ한 서버의 소프트웨어 정보가 들어감 |
| WWW-Authenticate | 요청-URI에 적용될 수 있는 인증체계와 매개변수 정보가 들어감 |
| Allow | 요청-URI에서 지원하는 메소드가 열거됨 |
| Content-Encoding | 몸체에 포함된 컨텐츠의 인코딩 형태를 알려주는데 사용됨 |
| Content-Type | 컨텐츠의 미디어 타입을 알려주는데 사용됨 |
| Expires | 컨텐츠가 더 이상 유효하지 않을 수 있는 날짜/시간을 제공하는데 사용함 |
| Last=Modified | 리소스가 마지막으로 업데이트되었다고 여겨지는 날짜/시간을 제공하는데 사용됨 |
| Accept-Rangers | 서버가 리소스 요청에 대한 수락 범위를 나타내는데 사용함 |
| ETag | 요청된 변형에 대한 엔티티 태그의 현재 값을 나타내는데 사용됨 |
| Content-Language | 컨텐츠에서 사용된 언어를 나타내는데 사용됨 |
| Content-Location | 메시지에 포함된 리소스의 위치를 공유하는데 사용됨 |
| Content-MD5 | 몸체 컨텐츠의 MD5 코드를 제공하는데 사용됨 |
| Transfer-Encoding | 송수신자 간에 안전하게 전송하기 위해 메시지 본문에 적용된 변환 유형을 나타냄 |

녹색으로 나타낸 줄은 빈 줄을 의미하며, 응답 헤더와 응답 몸체를 분리시켜준다.

파란색으로 나타낸 줄은 응답 몸체를 의미한다.

# 참고문헌(Reference)

[1] "HTTP", 한국어 위키백과, Accessed on: 2022-09-11 0:30:40

[2] RYAN, “HTTP 프로토콜 구조 개요와 예제”, ry.cl. blog, 2020-10-17, Accessed on: 2022-09-11 19:07:48