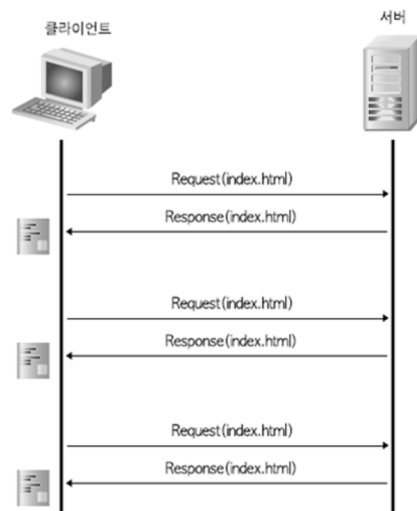


HTTP Protocol

- 응용 계층 프로토콜
- TCP 계열로 80번 포트 사용
- WWW 상에서 HTML 문서를 주고 받는 프로토콜
- 클라이언트와 서버 사이에 이뤄지는 요청(request) / 응답 (response) 프로토콜
- 현재 인터넷에서 사용되는 HTTP 버전은 1.0과 1.1
- 무상태(stateless) 프로토콜

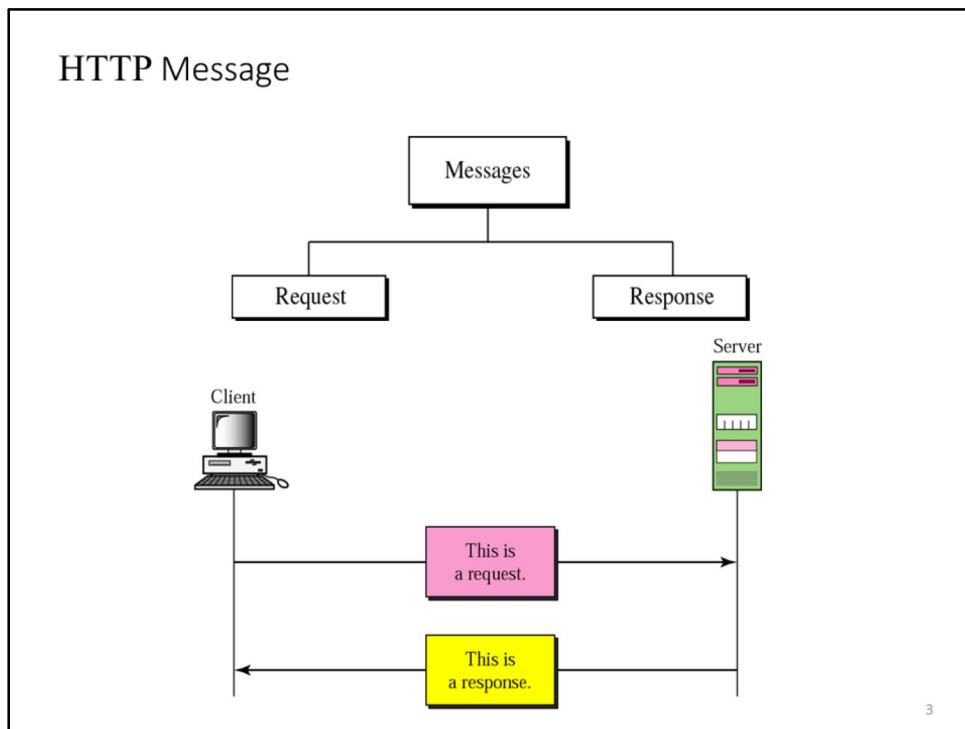
Stateless Protocol



2

Stateless protocol

- 과거에 수행한 통신 내용을 기억하지 않는 프로토콜
- 페이지를 요청하고 응답 받는 것으로 통신을 종결
- 이전에 전송 받은 페이지와 새로 전송 받는 페이지간의 연결성이 전혀 없음
- 프로그램을 통한 세션 관리를 통해 사용자의 요청을 기억하고 처리하는 기능필요



클라이언트와 서버 사이에 이뤄지는 요청(request)/응답(response) 프로토콜

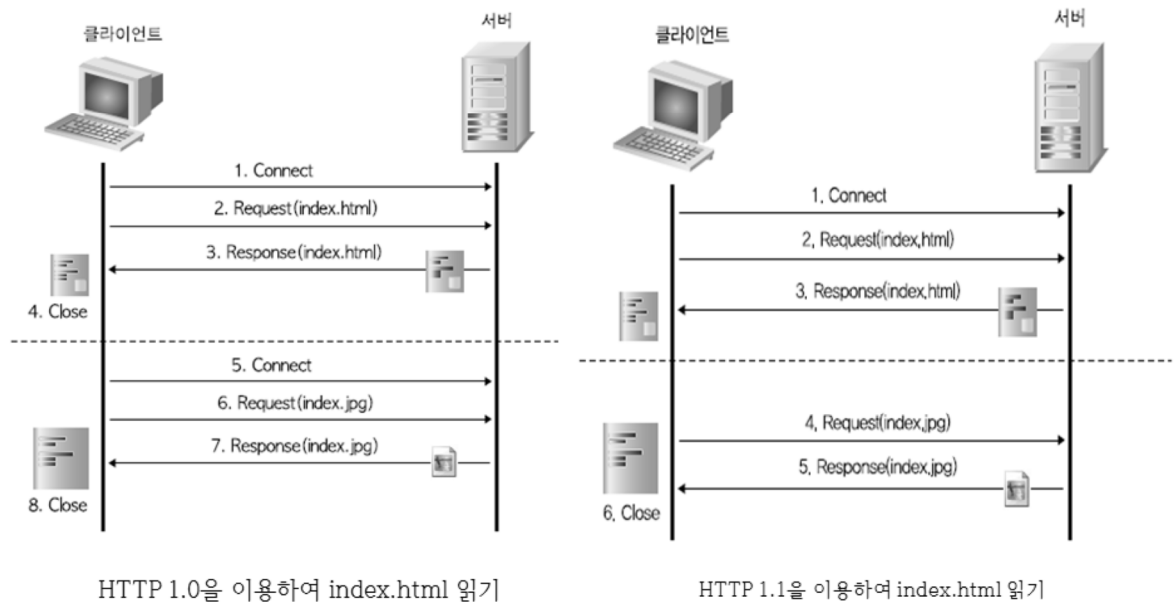
-클라이언트의 웹 브라우저가 서버의 웹 페이지나 그림 정보를 요청하면 서버는 이 요청에 응답하여 필요한 정보를 클라이언트에게 전달

HTTP 메시지

-요청 메시지(request) : 클라이언트 → 서버

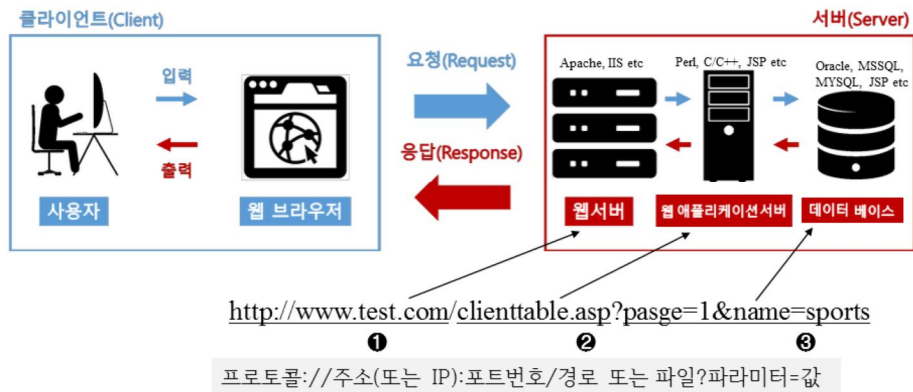
-응답 메시지(response) : 서버 → 클라이언트

HTTP Version



- HTTP 1.0
 - 텍스트가 저장된 HTML 문서를 먼저 전송 받은 후 연결을 끊고 나서 다시 연결하여 그림을 전송 받음
 - 1995년 5월에 완성
 - 메소드는 GET, HEAD, POST 방식만 지원
- HTTP 1.1
 - HTML 문서를 받고 난 후 연결을 끊지 않고 바로 그림 파일을 요청하여 전송 받음
 - 2001년에 공식 발표되었음
 - 메소드는 OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT 방식 지원

웹 애플리케이션 구조



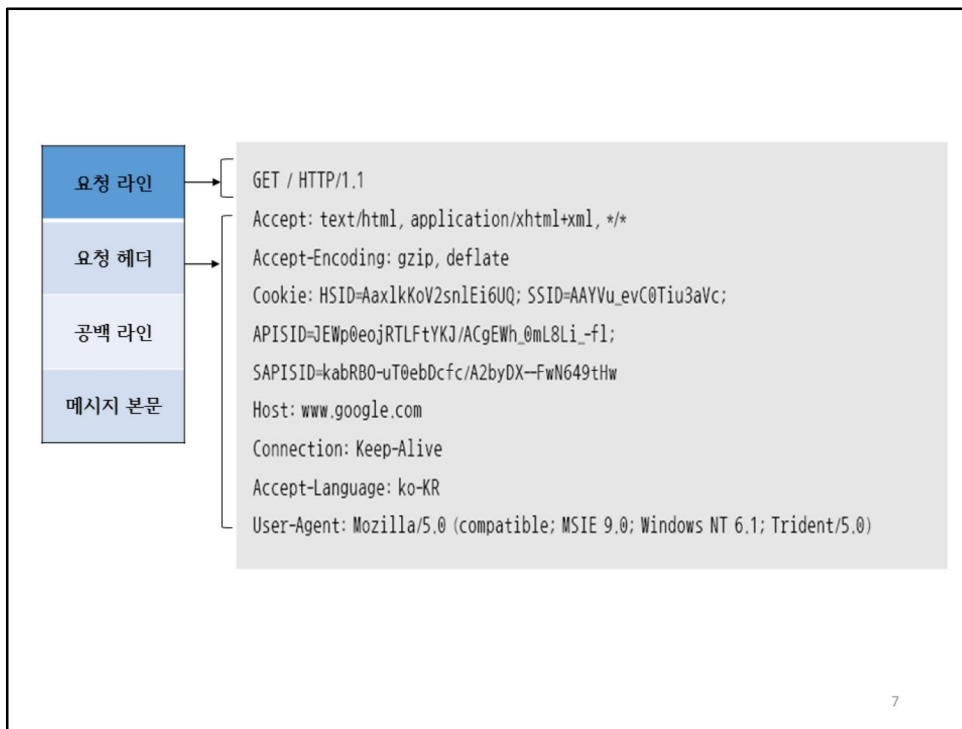
5

- 웹 클라이언트
 - 사용자(클라이언트)가 요청한 웹 서버의 결과를 화면에 출력해 보여주는 애플리케이션
 - 인터넷 익스플로러를 비롯해 크롬, 파이어폭스 등의 웹브라우저 사용
- 웹 서버
 - 웹브라우저의 요청을 받아 들어서 이에 알맞은 결과를 웹 브라우저에 응답해 주는 역할
 - 정적인 페이지 처리를 담당
 - 요청에 필요한 페이지의 로직이나 데이터베이스와의 연동을 위해서 애플리케이션 서버에 처리 요청
 - HTTP 요청과 응답을 처리하는 것으로 Apache와 IIS 등이 있음
- 웹 애플리케이션 서버(WAS)
 - 요청한 페이지의 로직이나 데이터베이스와의 연동을 처리
 - 동적인 페이지 처리 담당
 - 데이터를 가공하거나 화면을 유동적으로 처리
 - 톰캣(Tomcat), WebLogin, WebSphere 등
 - 플러그인 : Perl, C/C++, JSP 등
- 데이터베이스 관리 시스템(DBMS)
 - 데이터를 저장하고 관리하는 것
 - Oracle, MS-SQL, MySQL 등

HTTP Method

1) Request Method

타입	설명
GET	<ul style="list-style-type: none">• 요청 데이터에 대한 인수를 URL을 통해 전송• 이름과 값을 &로 결합하며 글자수를 255자로 제한• 아이디나 패스워드가 인수로 저장되어 전달• 데이터가 주소 입력란에 표시되어 보안에 취약
POST	<ul style="list-style-type: none">• URL에 요청 데이터를 기록하지 않고 HTTP 헤더에 데이터를 전송• 인수 값을 URL을 통해 전송하지 않음• 서버가 각 구분자에 대한 내용을 해석하여 데이터를 처리하기 때문에 GET 방식에 비해 상대적으로 처리속도가 느림



GET /HTTP/1.1

GET [request-uri]?query_string HTTP/1.1

-데이터 처리 방식, 경로 및 파일명(URI), HTTP 프로토콜 버전

Accept : text/html, application/xhtml+xml, */*

응답 메시지에 대해 허용할 수 있는 미디어 종류

웹 서버로부터 수신되는 데이터 중 웹 브라우저 가 처리할 수 있는 데이터 타입

-text/html : text, html 형태의 문서

-*/* : 모든 문서

Accept-Encoding: gzip, deflate

클라이언트가 인식 할 수 있는 인코딩(압축) 방법

- gzip, deflate : 압축한 리소스를 클라이언트가 해석 가능

- 서버에서 압축을 했을 경우 응답헤더에 content-encoding 헤더에 해당 압축 방법을 명시

Cookie : name = value

프로토콜 자체가 세션을 유지하지 않는 state-less(접속 상태를 유지 않는) 방식이기 때문에 사용자 정보를 기억하기 위해 만든 인위적인 값

서버가 클라이언트에 전송한 인자 값에 추가 정보를 보낼 때 사용

Host : www.google.com

응답을 요청한 호스트, 사용자가 요청한 도메인 정보가 포함

Accept-Language : ko-KR

클라이언트 웹브라우저가 인식 할 수 있는 언어

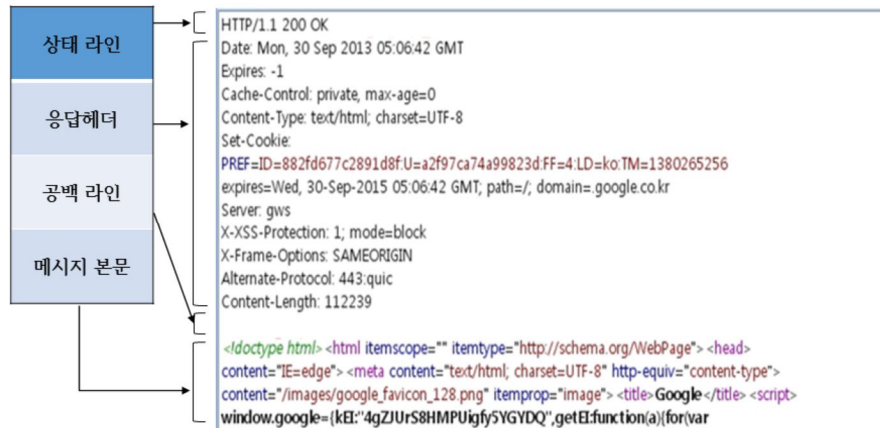
응답에 대해 선호하는 언어로 우선 순위 지정 가능

User-Agent : Mozilla/5.0(compatible; MSIE 9.0; Window NT 6.1; Trident/5.0)

응답 내용에 대해 응답할 수 있는 브라우저 종류

사용자의 웹 브라우저 종류 및 버전 정보가 포함

2) Response



8

- 헤더 정보 뒤에 빈 줄이 하나 들어가고, 그 다음에 실제 데이터가 전달됨
- 실제 데이터는 HTML이나 그림 파일이며, 데이터 전달이 종료되면 서버는 연결을 끊음

[http response header 구성]

HTTP /1.1 200 OK

- HTTP 버전과 응답 코드

Date

- 응답시간

Server

- 응답 웹서버 정보

Content-Type

- 내용의 형식

- 요청한 파일의 MIME 타입, Text/html 은 text 중 html 파일

Last-Modified

- 요청한 파일의 최종 수정일, 최근 응답 페이지 수정일

Content-Length

-헤더 이후 이어지는 데이터의 길이(바이트 단위)

Entity Body

- 클라이언트가 요청한 페이지 내용과 같은 전달할 값

❖ HTTP response 상태 코드

웹 서버 상태 코드	함축적 의미	내용
100번 대	정보 전송	임시 응답을 나타내는 것은 Status-Line과 선택적인 헤더들로 이루어져 있고 빈 줄로 끝을 맺는다. HTTP/1.0까지는 계열에 대한 어떤 정의도 이루어지지 않았기 때문에 시험용 외에는 서버 쪽의 추가 응답은 없다.
200번 대	성공	클라이언트의 요청이 성공적으로 수신되어 처리되었음을 뜻한다.
300번 대	리다이렉션	클라이언트의 요구 사항을 처리하기 위해서는 다른 곳에 있는 자원이 필요하다는 것을 의미한다.
400번 대	클라이언트 측 에러	클라이언트가 서버에 보내는 요구 메시지를 완전히 처리하지 못한 경우 등 클라이언트 측에서 오류가 발생한 것을 의미한다.
500번 대	서버 측 에러	서버 자체에서 생긴 오류 상황이나 클라이언트 요구 사항을 제대로 처리 할 수 없을 때 발생한다.

- 200 OK : 클라이언트의 요청이 성공했다는 것을 나타냄
- 201 Created : 클라이언트의 PUT 요청이 성공적이라는 것을 나타냄
- 301 Moved Permanently : 브라우저의 요청을 다른 URL로 항상 전달한다는 것을 의미함. 다른 URL에 대한 정보는 Location 헤더에 나타남
- 400 Bad Request : 클라이언트가 서버에 잘못된 요청을 했다는 것을 나타냄
- 401 Unauthorized : 서버가 클라이언트의 요청에 대해 HTTP 인증 확인을 요구하는 것을 의미함
- 403 Forbidden : 클라이언트의 요청에 대해 접근을 차단하는 것을 나타냄
- 404 Not Found : 클라이언트가 서버에 요청한 자료가 존재하지 않음을 나타냄
- 500 Internal Server Error : 서버가 클라이언트의 요청을 실행할 수 없을 때 500 상태 코드가 발생함. 일반적으로 SQL 인젝션 취약점이 존재하는지 확인할 때 500 에러가 유용하게 사용됨