

3장.HTTP 메시지

3.1 메시지의 흐름

3.1.1 메시지는 원 서버 방향을 인바운드로 하여 송신된다.

3.1.2 다운스트림으로 흐르는 메시지.

3.2 메시지의 각 부분

3.2.1 메시지 문법

3.2.2 시작줄

3.2.3 헤더

3.2.4 엔터티 본문

3.3 메서드

3.3.1 안전한 메서드

3.3.2 GET

3.3.3 HEAD

3.3.4 PUT

3.3.5 POST

3.3.6 TRACE

3.3.7 OPTIONS

3.3.8 DELETE

3.3.9 확장 메서드

3.4 상태 코드

3.4.1 정보성 상태 코드(100-199)

3.4.2 성공 상태 코드(200-299)

3.4.3 리다이렉션 상태 코드(300-399)

3.4.4 클라이언트 에러 상태 코드(400-499)

3.4.5 서버 에러 상태 코드(500-599)

3.5 헤더

3.5.1 일반 헤더

3.5.2 요청 헤더

3.5.3 응답 헤더

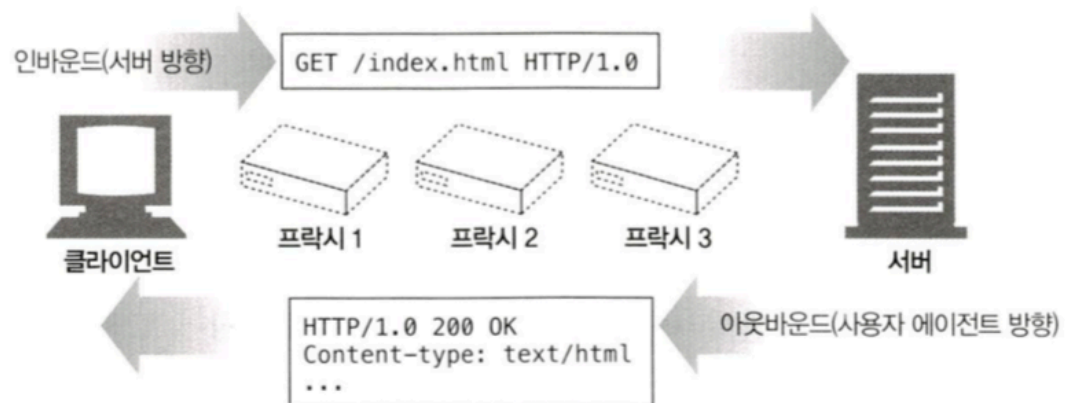
3.5.4 엔터티 헤더

3.1 메시지의 흐름

- HTTP 메시지는 HTTP 애플리케이션 간에 주고받은 데이터의 블록들이다.
- 이 데이터의 블록들은 메시지의 내용과 의미를 설명하는 텍스트 메타 정보로 시작하고 그 다음에 선택적으로 데이터가 올 수 있다.

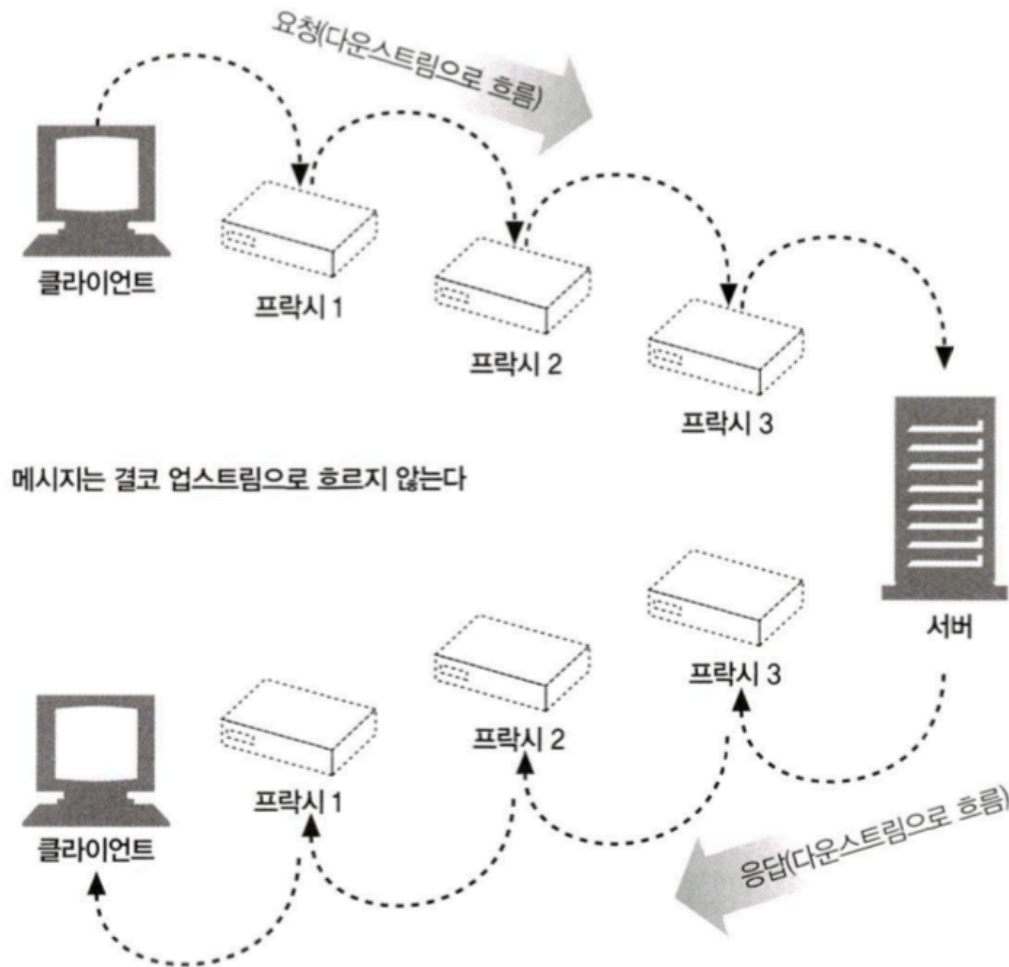
3.1.1 메시지는 원 서버 방향을 인바운드로 하여 송신한다

- 메시지가 원 서버로 향하는 것은 인바운드로 이동하는 것이고, 모든 처리가 끝난 뒤에 메시지가 사용자 에이전트로 돌아오는 것은 아웃바운드로 이동하는 것이다.



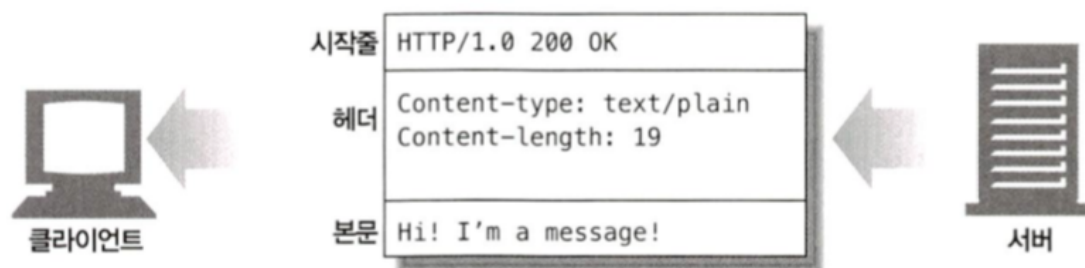
3.1.2 다운스트림으로 흐르는 메시지

- 모든 메시지는 다운스트림으로 흐른다. 메시지의 발송자는 수신자의 업스트림이다.



3.2 메시지의 각 부분

- 메시지는 클라이언트로부터의 요청이나 서버로부터의 응답 중 하나를 포함한다. 메시지는 시작줄, 헤더 블록, 본문 세 부분으로 이루어진다. 시작줄은 이것이 어떤 메시지인지 서술하며, 헤더 블록은 속성을, 본문은 데이터를 담고 있다. 본문은 아예 없을 수도 있다.



3.2.1 메시지 문법

- 모든 HTTP 메시지는 요청 메시지나 응답 메시지로 분류된다
- 요청 메시지는 웹 서버에 어떤 동작을 요구한다
- 답 메시지는 요청의 결과를 클라이언트에게 돌려준다.



요청 메시지의 형식

<메서드> <요청 URL> <버전>
 <헤더>
 <엔티티 본문>

응답 메시지의 형식

<버전> <상태 코드> <사유 구절>
 <헤더>
 <엔티티 본문>

- 메서드
 - 클라이언트 측에서 서버가 리소스에 대해 수행해주길 바라는 동작이다. 'GET', 'HEAD', 'POST'와 같이 한 단어로 되어 있다.
- 요청 URL
 - 요청 대상이 되는 리소스를 지칭하는 완전한 URL 혹은 URL의 경로 구성요소다.

- 서버는 URL에서 생략된 호스트/포트가 자신을 가리키는 것으로 간주할 것이다.
- 버전
 - 이 메시지에서 사용 중인 HTTP의 버전이다. 형식은 다음과 같다.

| HTTP/<메이저>.<마이너>

- 상태 코드
 - 요청 중에 무엇이 일어났는지 설명하는 세 자리 숫자다.
 - 각 코드의 첫 번째 자릿수는 상태의 일반적인 분류(성공, 에러 등)를 나타낸다.
- 사유 구절
 - 숫자로 된 상태 코드의 의미를 사람이 이해할 수 있게 설명해주는 짧은 문구로, 상태 코드 이후부터 줄바꿈 문자열까지가 사유 구절이다.
 - 사유 구절은 오로지 사람에게 읽히기 위한 목적으로만 존재한다.
- 헤더들
 - 이름, 콜론(:), 선택적인 공백, 값, CRLF가 순서대로 나타나는 0개 이상의 헤더들.
 - 이 헤더의 목록은 빈 줄(CRLF)로 끝나 헤더 목록의 끝과 엔티티 본문의 시작을 표시한다.
- 엔티티 본문
 - 엔티티 본문은 임의의 데이터 블록을 포함한다.
 - 모든 메시지가 엔티티 본문을 갖는 것은 아니므로, 때때로 메시지는 그냥 CRLF로 끝나게 된다.

3.2.2 시작줄

- 요청 메시지의 시작줄은 무엇을 해야 하는지 말해준다.
- 응답 메시지의 시작줄은 무슨 일이 일어났는지 말해준다.
- 요청줄
 - 요청 메시지는 서버에게 리소스에 대해 무언가를 해달라고 부탁한다.

- 요청 메시지의 시작줄, 혹은 요청줄에는 서버에서 어떤 동작이 일어나야 하는지 설명해주는 메서드와 그 동작에 대한 대상을 지칭하는 요청 URL이 들어 있다.

- 응답줄

- 응답 메시지는 수행 결과에 대한 상태 정보와 결과 데이터를 클라이언트에게 돌려준다.
- 응답 메시지의 시작줄 혹은 응답줄에는 응답 메시지에서 쓰인 HTTP의 버전, 숫자로 된 상태 코드, 수행 상태에 대해 설명해주는 텍스트로 된 사유 구절이 들어 있다.

- 메서드

요청의 시작줄은 메서드로 시작하며, 서버에게 무엇을 해야 하는지 말해준다.

메서드	설명	메시지 본문이 있는가?
GET	서버에서 어떤 문서를 가져온다	X
HEAD	서버에서 어떤 문서에 대해 헤더만 가져온다	X
POST	서버가 처리해야 할 데이터를 보낸다	O
PUT	서버에 요청 메시지의 본문을 저장한다	O
TRACE	메시지가 프록시 서버를 거쳐 서버에 도달하는 과정을 추적한다	X
OPTIONS	서버가 어떤 메서드를 수행할 수 있는지 확인한다	X
DELETE	서버에서 문서를 제거한다	X

- 상태 코드

- 상태 코드는 클라이언트에게 무엇이 일어났는지 말해준다.
- 상태 코드는 응답의 시작줄에 위치한다. 예를 들어, `HTTP/1.0 200 OK` 라는 줄에서 상태 코드는 200이다.

상태 코드들은 세 자리 숫자로 된 그들의 코드값을 기준으로 묶인다.

전체 범위	정의된 범위	분류
100-199	100-101	정보
200-299	200-206	성공
300-399	300-305	리다이렉션
400-499	400-415	클라이언트 에러
500-599	500-505	서버 에러

- 사유 구절
 - 사유 구절은 응답 시작줄의 마지막 구성요소다.
 - 사유 구절은, 애플리케이션 개발자들이 그들의 사용자에게 요청 중에 무슨 일이 일어났는지 알려주기 위해 넘겨줄 수 있는, 상태 코드의 사람이 이해하기 쉬운 버전이다.
- 버전 번호
 - 버전 번호는 HTTP/x.y 형식으로 요청과 응답 메시지 양쪽 모두에 기술된다.
 - HTTP 애플리케이션들이 자신이 따르는 프로토콜의 버전을 상대방에게 말해주는 위한 수단이 된다.
 - 버전 번호는 어떤 애플리케이션이 지원하는 가장 높은 HTTP 버전을 가리킨다.

3.2.3 헤더

- 시작줄 다음에는 0개~여러 개의 HTTP 헤더가 온다.
- HTTP 헤더 필드는 요청과 응답 메시지에 추가 정보를 더한다. 기본적으로 이름/값 쌍 목록이다.

헤더 분류

- 일반 헤더: 요청과 응답 양쪽에 모두 나타날 수 있음
- 요청 헤더: 요청에 대한 부가 정보를 제공
- 응답 헤더: 응답에 대한 부가 정보를 제공
- Entity 헤더: 본문 크기와 콘텐츠, 혹은 리소스 그 자체를 서술
- 확장 헤더: 명세에 정의되지 않은 새로운 헤더

3.2.4 엔티티 본문

- HTTP 메시지의 세 번째 부분은 선택적인 엔티티 본문이다.
- HTTP 메시지는 이미지, 비디오, HTML 문서, 소프트웨어 애플리케이션 등 여러 종류의 디지털 데이터를 실어 나를 수 있다.

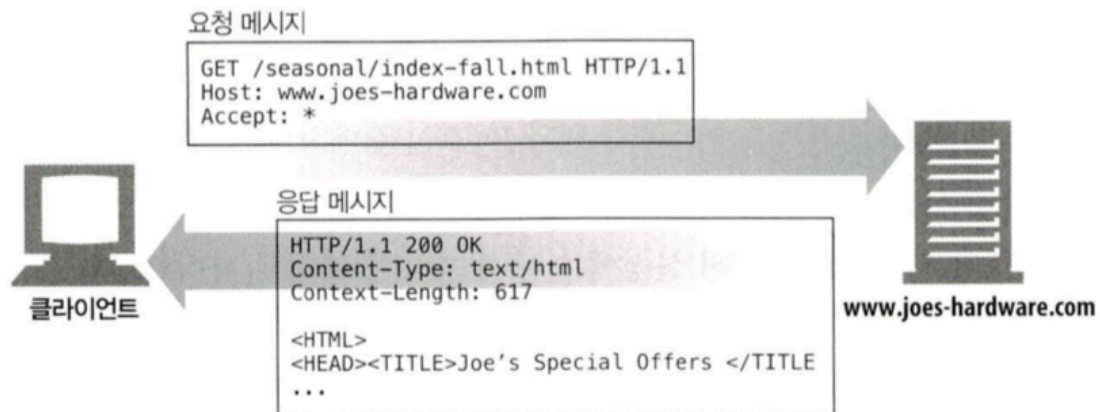
3.3 메서드

3.3.1 안전한 메서드(Safe Method)

- HTTP는 안전한 메서드라 불리는 메서드의 집합을 정의한다.
- GET과 HEAD 메서드는 안전하다고 할 수 있는데, 이는 GET이나 HEAD 메서드를 사용하는 HTTP 요청의 결과로 서버에 어떤 작용도 없음을 나타낸다.

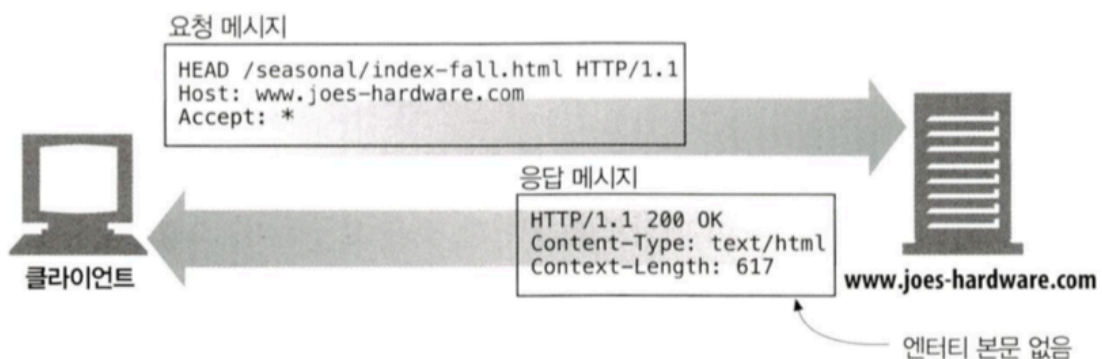
3.3.2 GET

- GET은 주로 서버에게 리소스를 달라고 요청하기 위해 쓰인다.



3.3.3 HEAD

- HEAD 메서드는 정확히 GET처럼 동작하지만, 서버는 응답으로 헤더만을 돌려준다. 엔티티 본문은 결코 반환되지 않는다. HEAD를 사용하면
 - 리소스를 가져오지 않고도 그에 대해 무엇인가(타입이라거나)를 알아낼 수 있다.
 - 응답의 상태 코드를 통해, 개체가 존재하는지 확인할 수 있다.
 - 헤더를 확인하여 리소스가 변경되었는지 검사할 수 있다.



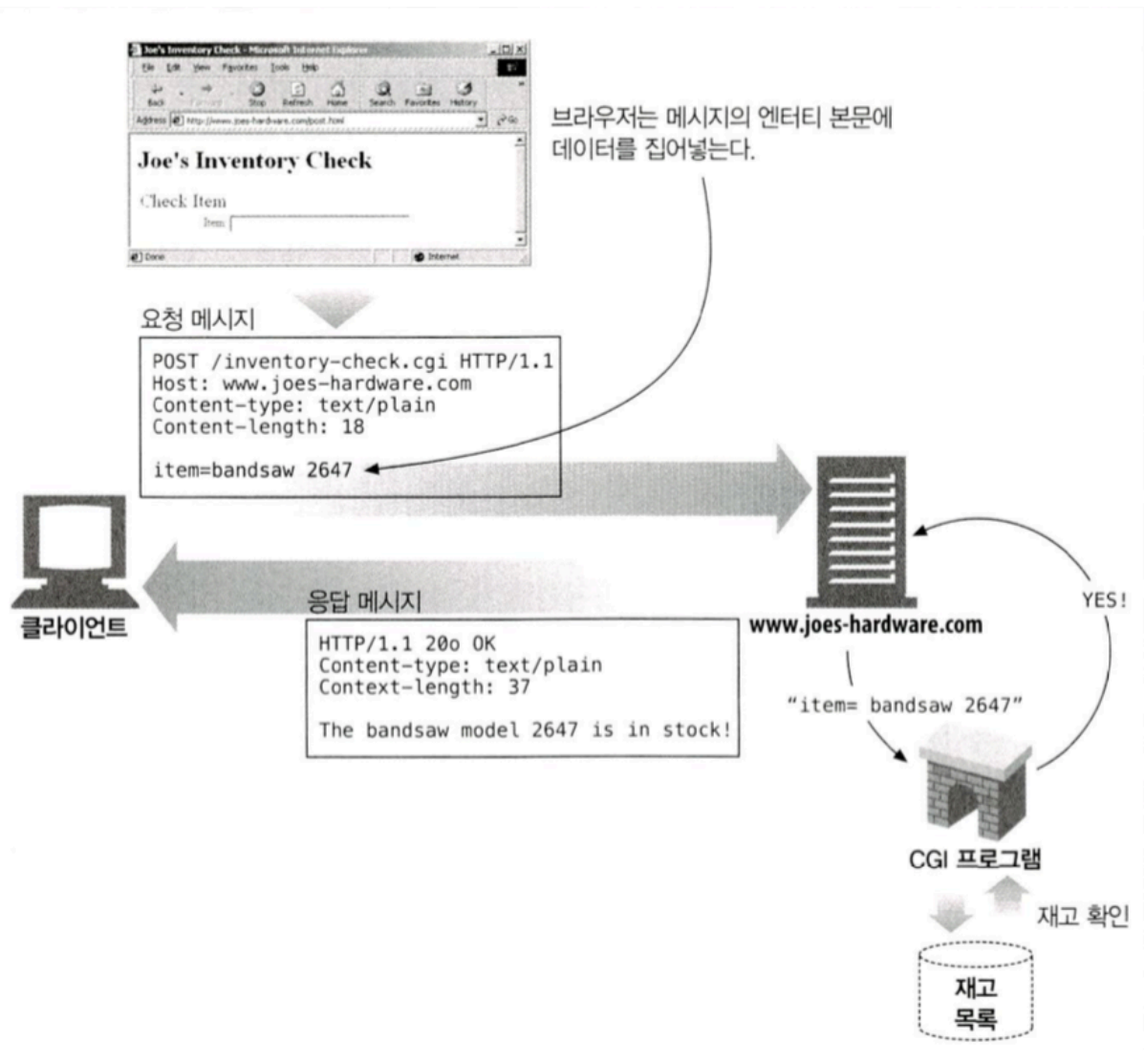
3.3.4 PUT

- PUT 메서드의 의미는, 서버가 요청의 본문을 가지고 요청 URL의 이름대로 새 문서를 만들거나, 이미 URL이 존재한다면 본문을 사용해서 교체하는 것이다.



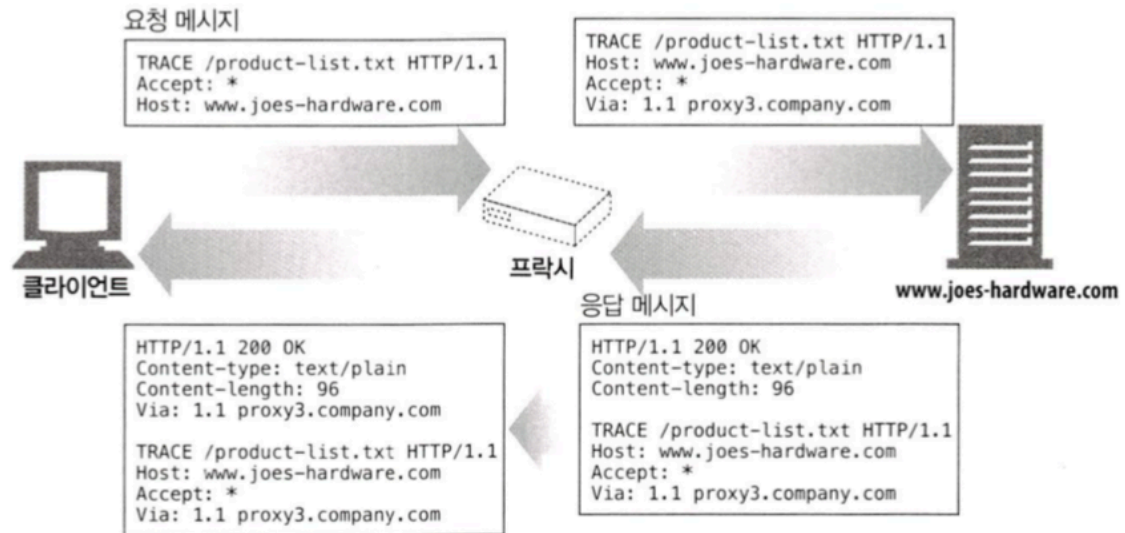
3.3.5 POST

- POST 메서드는 서버에 입력 데이터를 전송하기 위해 설계되었다. 채워진 폼에 담긴 데이터는 서버로 전송되며, 서버는 이를 모아서 필요로 하는 곳에 보낸다.



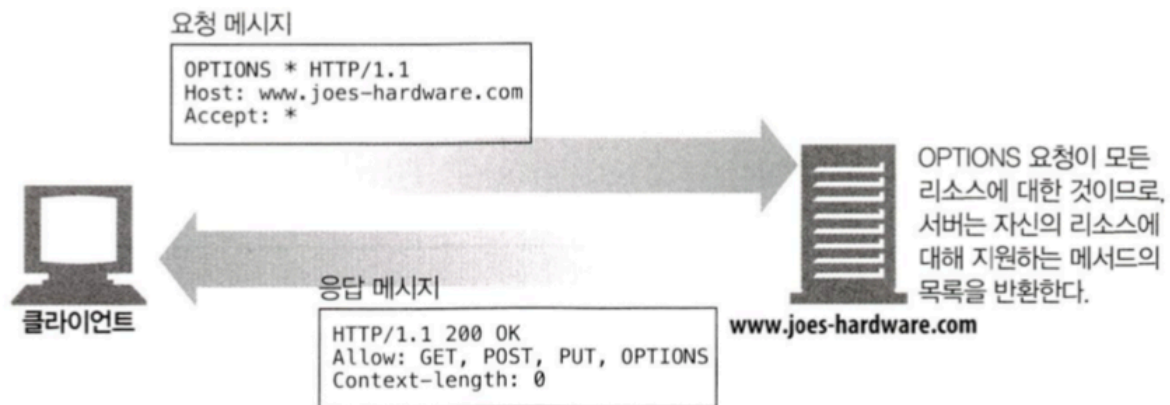
3.3.6 TRACE

- 클라이언트가 어떤 요청을 할 때, 그 요청은 방화벽, 프록시, 게이트웨이 등의 애플리케이션을 통과할 수 있다. TRACE 메서드는 클라이언트에게 자신의 요청이 서버에 도달했을 때 어떻게 보이게 되는지 알려준다.
- 요청 전송의 마지막 단계에 있는 서버는 자신이 받은 요청 메시지를 본문에 넣어 TRACE 응답을 되돌려준다. 클라이언트는 자신과 목적지 서버 사이에 모든 HTTP 애플리케이션의 요청/응답 연쇄를 따라가면서 자신이 보낸 메시지가 망가졌거나 수정되었는지, 만약 그렇다면 어떻게 변경되었는지 확인할 수 있다.



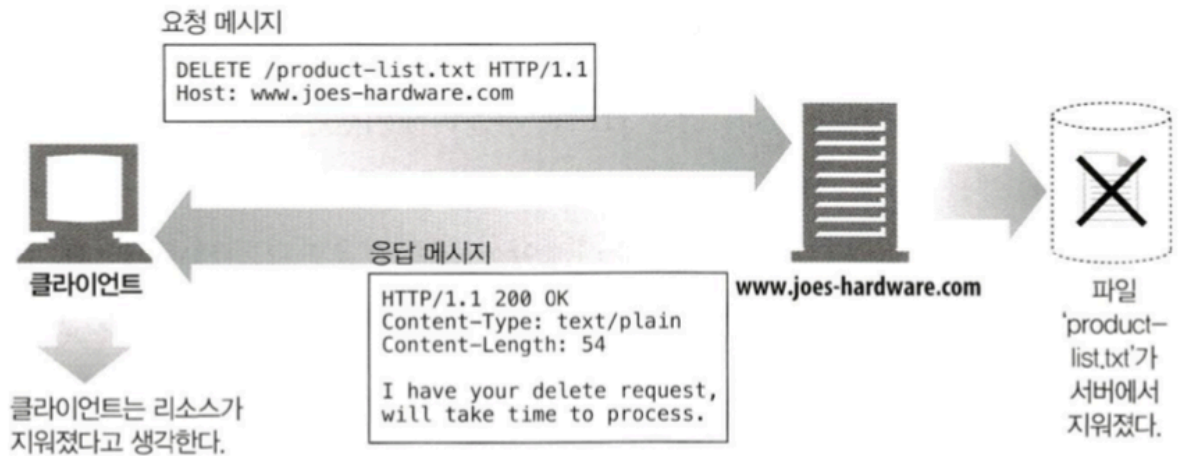
3.3.7 OPTIONS

- OPTIONS 메서드는 웹 서버에게 여러 가지 종류의 지원 범위에 대해 물어본다.



3.3.8 DELETE

- DELETE 메서드는 서버에게 요청 URL로 지정한 리소스를 삭제할 것을 요청한다.



3.4 상태 코드

- 상태 코드는 클라이언트에게 그들의 트랜잭션을 이해할 수 있는 쉬운 방법을 제공한다.

3.4.1 100-199: 정보성 상태 코드

상태코드	사유 구절	의미
100	Continue	요청의 시작 부분 일부가 받아들여졌으며, 클라이언트는 나머지를 계속 이어서 보내야 함을 의미한다. 이것을 보낸 후, 서버는 반드시 요청을 받아 응답해야 한다.
101	Switching Protocols	클라이언트가 Upgrade 헤더에 나열한 것 중 하나로 서버가 프로토콜을 바꾸었음을 의미한다.

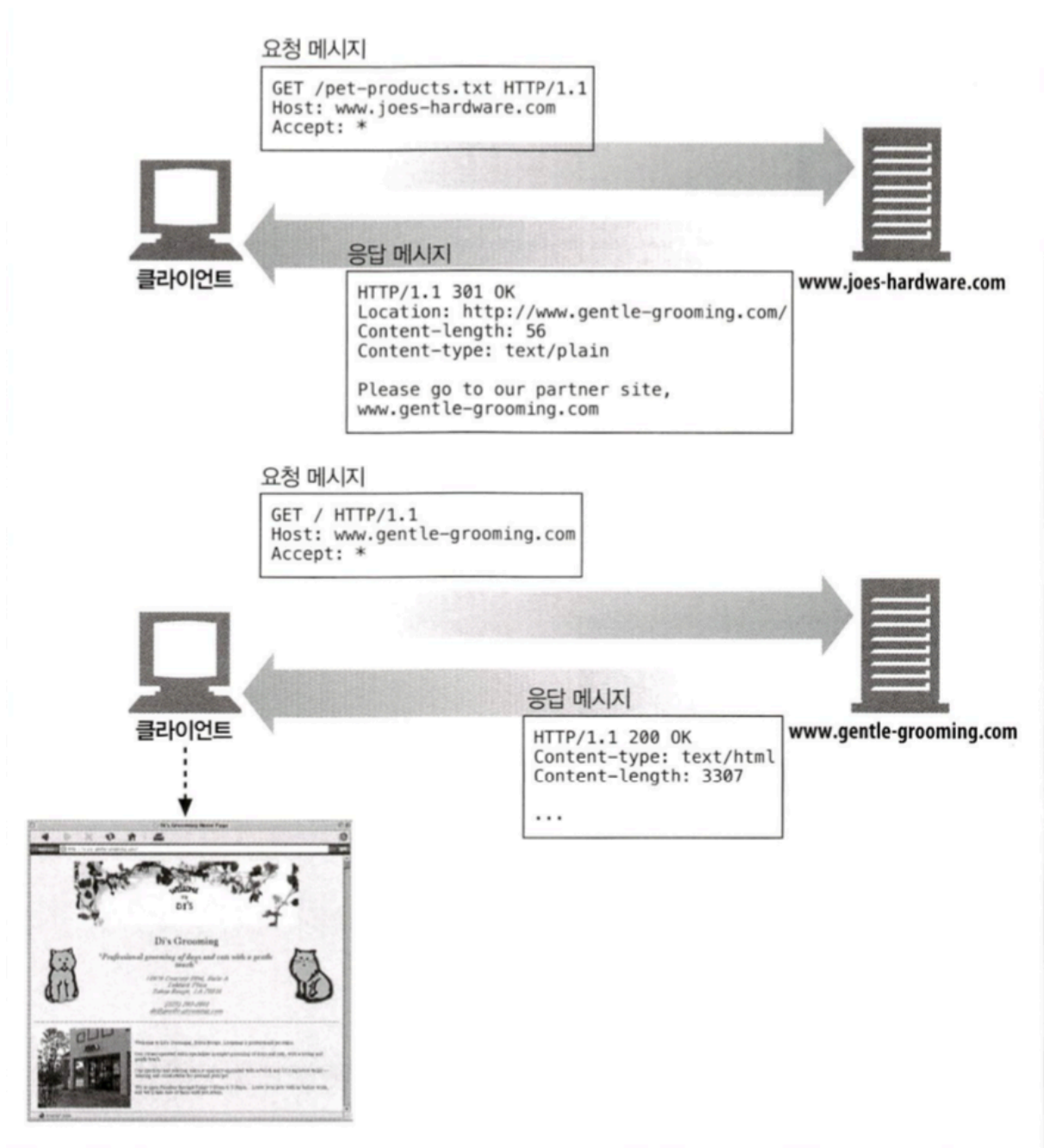
3.4.2 200-299: 성공 상태 코드

상태 코드	사유 구절	의미
200	OK	요청은 정상이고, 엔티티 본문은 요청된 리소스를 포함하고 있다
201	Created	서버 개체를 생성하라는 요청을 위한 것. 서버는 상태 코드를 보내기에 앞서 반드시 객체를 생성해야 한다.
202	Accepted	요청은 받아들여졌으나 서버는 아직 그에 대한 어떤 동작도 수행하지 않았다. 서버는 엔티티 본문에 요청에 대한 상태와 가급적이면 요청의 처리가 언제 완료될 것인지에 대한 추정도 포함해야 한다.

203	Non-Authoritative Information	중개자가 리소스의 사본을 갖고 있었지만 리소스에 대한 메타 정보(헤더)를 검증하지 못한 경우 이런 일이 발생할 수 있다.
204	No Content	응답 메시지는 헤더와 상태줄을 포함하지만 엔티티 본문은 포함하지 않는다. 주로 웹 브라우저를 새 문서로 이동시키지 않고 갱신하고자 할 때 사용한다.
205	Reset Content	주로 브라우저를 위해 사용되는 또 하나의 코드. 브라우저에게 현재 페이지에 있는 HTML 폼에 채워진 모든 값을 비우라고 말한다.
206	Partial Content	부분 혹은 범위 요청이 성공했다. 206 응답은 Content-Range와 Date 헤더를 반드시 포함해야 하며, Etag와 Content-Location 중 하나의 헤더도 반드시 포함해야 한다.

3.4.3 300-399: 리다이렉션 상태 코드

- 리다이렉션 상태 코드는 클라이언트가 관심 있어 하는 리소스에 대해 다른 위치를 사용하라고 말해주거나 그 리소스의 내용 대신 다른 대안 응답을 제공한다.



상태 코드	사유 구절	의미
300	Multiple Choices	클라이언트가 동시에 여러 리소스를 가리키는 URL을 요청한 경우, 그 리소스의 목록과 함께 반환한다.
301	Moved Permanently	요청한 URL이 옮겨졌을 때 사용한다. 응답은 Location 헤더에 현재 리소스가 존재하고 있는 URL을 포함해야 한다.
302	Found	301 상태 코드와 같다. 그러나 클라이언트는 Location 헤더로 주어진 URL을 임시로 가리키기 위한 목적으로 사용해야 한다.

303	See Other	클라이언트에게 리소스를 다른 URL에서 가져와야 한다고 말해주고자 할 때 쓰인다. 이 상태 코드의 주 목적은 POST 요청에 대한 응답으로 클라이언트에게 리소스의 위치를 알려주는 것이다.
304	Not Modified	클라이언트는 헤더를 이용해 조건부 요청을 만들 수 있다.
305	Use Proxy	리소스가 반드시 프록시를 통해서 접근되어야 함을 나타내기 위해 사용한다.
307	Temporary Redirect	301 상태 코드와 비슷하다. 그러나 클라이언트는 Location 헤더로 주어진 URL을 리소스를 임시로 가리키기 위한 목적으로 사용해야 한다.

3.4.4 400-499: 클라이언트 에러 상태 코드

상태 코드	사유 구절	의미
400	Bad Request	클라이언트가 잘못된 요청을 보냈다고 말해준다
401	Unauthorized	리소스를 얻기 전에 클라이언트에게 스스로를 인증하라고 요구하는 애용의 응답을 적절한 헤더와 함께 반환한다
402	Payment Required	현재 이 상태 코드는 쓰이지 않지만, 미래에 사용될 가능성을 위해 준비해두었다
403	Forbidden	요청이 서버에 의해 거부되었음을 알려주기 위해 사용한다.
404	Not Found	서버가 요청한 URL을 찾을 수 없음을 알려주기 위해 사용한다.
405	Method Not Allowed	요청한 URL에 대해, 지원하지 않는 메서드로 요청받았을 때 사용한다.
406	Not Acceptable	클라이언트는 자신이 어떤 종류의 엔티티를 받아들이고자 하는지에 대해 매개변수로 명시할 수 있다.
407	Proxy Authentication Required	401 상태 코드와 같으나, 리소스에 대해 인증을 요구하는 프록시 서버를 위해 사용된다.
408	Request Timeout	클라이언트의 요청을 완수하기에 시간이 너무 많이 걸리는 경우, 서버는 이 상태 코드로 응답하고 연결을 끊을 수 있다.

3.4.5 500-599: 서버 에러 상태 코드

상태 코드	사유 구절	의미
-------	-------	----

500	Internal Server Error	서버가 요청을 처리할 수 없게 만드는 에러를 만났을 때 사용한다
501	Not Implemented	클라이언트가 서버의 능력을 넘는 요청을 했을 때 사용한다.
502	Bad Gateway	프록시나 게이트웨이처럼 행동하는 서버가 그 요청 응답 연쇄에 있는 다음 링크로부터 가짜 응답에 맞닥뜨렸을 때 사용한다.
503	Service Unavailable	현재는 서버가 요청을 처리해 줄 수 없지만 나중에는 가능함을 의미하고자 할 때 사용한다.
504	Gateway Timeout	상태 코드 408과 비슷하지만, 다른 서버에게 요청을 보내고 응답을 기다리다 타임아웃이 발생한 게이트웨이나 프록시에서 온 응답이라는 점이 다르다.
505	HTTP Version Not Supported	서버가 지원할 수 없거나 지원하지 않으려고 하는 버전의 프로토콜로 된 요청을 받았을 때 사용한다

3.5 헤더

- 헤더와 메서드는 클라이언트와 서버가 무엇을 하는지 결정하기 위해 함께 사용된다.
- 헤더에는 특정 종류의 메시지에만 사용할 수 있는 헤더와, 더 일반 목적으로 사용할 수 있는 헤더, 그리고 응답과 요청 메시지 양쪽 모두에서 제공하는 헤더가 있다.

3.5.1 일반 헤더(General Headers)

- 일반 헤더는 클라이언트와 서버 양쪽 모두가 사용한다. 이들은 클라이언트, 서버, 그리고 어딘가에 메시지를 보내는 다른 애플리케이션들을 위해 다양한 목적으로 사용된다.

헤더	설명
Connection	클라이언트와 서버가 요청/응답 연결에 대한 옵션을 정할 수 있게 해준다
Date	메시지가 언제 만들어졌는지에 대한 날짜와 시간을 제공한다
MIME-Version	발송자가 사용한 MIME 버전을 알려준다
Trailer chunked transfer	인코딩으로 인코딩된 메시지의 끝 부분에 위치한 헤더들의 목록을 나열한다
Transfer-Encoding	수신자에게 안전한 전송을 위해 메시지에 어떤 인코딩이 적용되었는지 말해준다
Upgrade	발송자가 업그레이드 하길 원하는 새 버전이나 프로토콜을 알려준다
Via	이 메시지가 어떤 중개자(프록시, 게이트웨이)를 거쳐 왔는지 보여준다

일반 캐시 헤더

- HTTP/1.0은 HTTP 애플리케이션에게 매번 원 서버로부터 객체를 가져오는 대신 로컬 복사본으로 캐시할 수 있도록 해주는 최초의 헤더를 도입했다.

헤더	설명
Cache-Control	메시지와 함께 캐시 지시자를 전달하기 위해 사용한다
Pragma	메시지와 함께 지시자를 전달하는 또 다른 방법. 캐시에 국한되지 않는다

3.5.2 요청 헤더

- 요청 헤더는 요청 메시지에서만 의미를 갖는다.
- 요청이 최초 발생한 곳에서 누가 혹은 무엇이 그 요청을 보냈는지에 대한 정보나 클라이언트의 선호나 능력에 대한 정보를 준다.

헤더	설명
Client-IP	클라이언트가 실행된 컴퓨터의 IP를 제공한다
From	클라이언트 사용자의 메일 주소를 제공한다
Host	요청의 대상이 되는 서버의 호스트 명과 포트를 준다
Referer	현재의 요청 URI가 들어있었던 문서의 URL을 제공한다
User-Agent	요청을 보낸 애플리케이션의 이름을 서버에게 말해준다

Accept 관련 헤더

- 클라이언트는 Accept 관련 헤더들을 이용해 서버에게 자신의 선호와 능력을 알려줄 수 있다.
- 클라이언트가 무엇을 원하고 무엇을 할 수 있는지, 그리고 무엇보다도 원치 않는 것은 무엇인지 알려줄 수 있다.
- 클라이언트는 그들이 원하는 것을 얻을 수 있으며, 서버는 클라이언트가 사용할 수도 없는 것을 전송하는 데 시간과 대역폭을 낭비하지 않을 수 있다.

헤더	설명
Accept	서버에게 서버가 보내도 되는 미디어 종류를 말해준다
Accept-Charset	서버에게 서버가 보내도 되는 문자집합을 말해준다
Accept-Encoding	서버에게 서버가 보내도 되는 인코딩을 말해준다
Accept-Language	서버에게 서버가 보내도 되는 언어를 말해준다
TE	서버에게 서버가 보내도 되는 확장 전송 코딩을 말해준다

조건부 요청 헤더

- 클라이언트는 요청에 몇몇 제약을 넣기도 한다.
- 클라이언트가 이미 어떤 문서의 사본을 갖고 있는 상태라면, 클라이언트는 서버에게 그 문서를 요청할 때 자신이 갖고 있는 사본과 다를 때만 전송해 달라고 요청하고 싶을 수 있을 것이다.
- 조건부 요청 헤더를 사용하면, 클라이언트는 서버에게 요청에 응답하기 전에 먼저 조건이 참인지 확인하게 하는 제약을 포함시킬 수 있다.

헤더	설명
Expect	클라이언트가 요청에 필요한 서버의 행동을 열거할 수 있게 해준다
If-Match	문서의 엔티티 태그가 주어진 엔티티 태그와 일치하는 경우에만 문서를 가져온다
If-Modified-Since	주어진 날짜 이후에 리소스가 변경되지 않았다면 요청을 제한한다
If-None-Match	문서의 엔티티 태그가 주어진 엔티티 태그와 일치하지 않는 경우에만 문서를 가져온다
If-Range	문서의 특정 범위에 대한 요청을 할 수 있게 해준다
If-Unmodified-Since	주어진 날짜 이후에 리소스가 변경되었다면 요청을 제한한다
Range	서버가 범위 요청을 지원한다면, 리소스에 대한 특정 범위를 요청한다

요청 보안 헤더

- HTTP는 자체적으로 요청을 위한 간단한 인증요구/응답 체제를 갖고 있다.
- 그것은 요청하는 클라이언트가 어느 정도의 리소스에 접근하기 전에 자신을 인증하게 함으로써 트랜잭션을 약간 더 안전하게 만들고자 한다.

헤더	요청
Authorization	클라이언트가 서버에게 제공하는 인증 그 자체에 대한 정보를 담고 있다
Cookie	클라이언트가 서버에게 토큰을 전달할 때 사용한다.
Cookie2	요청자가 지원하는 쿠키의 버전을 알려줄 때 사용한다

프록시 요청 헤더

헤더	설명
Max-Forwards	요청이 원 서버로 향하는 과정에서 다른 프록시나 게이트웨이로 전달될 수 있는 최대 횟수. TRACE 메서드와 함께 사용된다

Proxy-Authorization	Authorization과 같으나 프록시에서 인증을 할 때 쓰인다
Proxy-Connection	Connection과 같으나 프록시에서 연결을 맺을 때 쓰인다

3.5.3 응답 헤더

- 응답 헤더는 클라이언트에게 부가 정보를 제공한다.
- 누가 응답을 보내고 있는지 혹은 응답자의 능력은 어떻게 되는지 알려주며, 더 나아가 응답에 대한 특별한 설명도 제공할 수 있다.
- 이 헤더들은 클라이언트가 응답을 잘 다루고 나중에 더 나은 요청을 할 수 있도록 도와준다.

헤더	설명
Age	응답이 얼마나 오래되었는지
Public	서버가 특정 리소스에 대해 지원하는 요청 메서드의 목록
Retry-After	현재 리소스가 사용 불가능한 상태일 때, 언제 가능해지는지 날짜 또는 시각
Server	서버 애플리케이션의 이름과 버전
Title	HTML 문서에서 주어진 것과 같은 제목
Warning	사유 구절에 있는 것보다 더 자세한 경고 메시지

협상 헤더

헤더	설명
Accept-Ranges	서버에 자원에 대해 받아들일 수 있는 범위의 형태
Vary	서버가 확인해 보아야 하고 그렇기 때문에 응답에 영향을 줄 수 있는 헤더들의 목록. ex) 서버가 클라이언트에게 보내줄 리소스의 가장 적절한 버전을 선택하기 위해 살펴보아야 하는 헤더들의 목록

응답 보안 헤더

헤더	설명
Proxy-Authenticate	프록시에서 클라이언트로 보낸 인증요구의 목록
Set-Cookie	진짜 보안 헤더는 아니지만, 보안에 영향을 줄 수 있다. 서버가 클라이언트를 인증할 수 있도록 클라이언트 측에 토큰을 설정하기 위해 사용한다
Set-Cookie2	Set-Cookie와 비슷하게 RFC 2965로 정의된 쿠키

WWW-Authenticate	서버에서 클라이언트로 보낸 인증요구의 목록
------------------	-------------------------

3.5.4 엔티티 헤더

- 엔티티 헤더는 엔티티와 그것의 내용물에 대한, 개체의 타입부터 시작해서 주어진 리소스에 대해 요청할 수 있는 유효한 메서드들까지, 광범위한 정보를 제공한다.

헤더	설명
Allow	이 엔티티에 대해 수행될 수 있는 요청 메서드들을 나열한다
Location	클라이언트에게 엔티티가 실제로 어디에 위치하고 있는지 말해준다. 수신자에게 리소스에 대한 위치(URL)를 알려줄 때 사용한다

콘텐츠 헤더

- 콘텐츠 헤더는 엔티티의 콘텐츠에 대한 구체적인 정보를 제공한다.
- 콘텐츠의 종류, 크기, 기타 콘텐츠를 처리할 때 유용하게 활용될 수 있는 것들이다.

헤더	설명
Content-Base	본문에서 사용된 상대 URL을 계산하기 위한 기저 URL
Content-Encoding	본문에서 적용된 어떤 인코딩
Content-Language	본문을 이해하는데 가장 적절한 자연어
Content-Length	본문의 크기나 길이
Content-Location	리소스가 실제로 어디에 위치하는지
Content-MD5	본문의 MD5 체크섬(checksum)
Content-Range	전체 리소스에서 이 엔티티가 해당하는 범위를 바이트 단위로 표현
Content-Type	이 본문이 어떤 종류의 객체인지

엔티티 캐싱 헤더

- 일반 캐싱 헤더는 언제 어떻게 캐시가 되어야 하는지에 대한 지시자를 제공한다.
- 엔티티 캐싱 헤더는 엔티티 캐싱에 대한 정보를 제공한다.

헤더	설명
ETag	이 엔티티에 대한 엔티티 태그
Expires	이 엔티티가 더 이상 유효하지 않아 원본을 받아와야 하는 일시
Last-Modified	가장 최근 이 엔티티가 변경된 일시