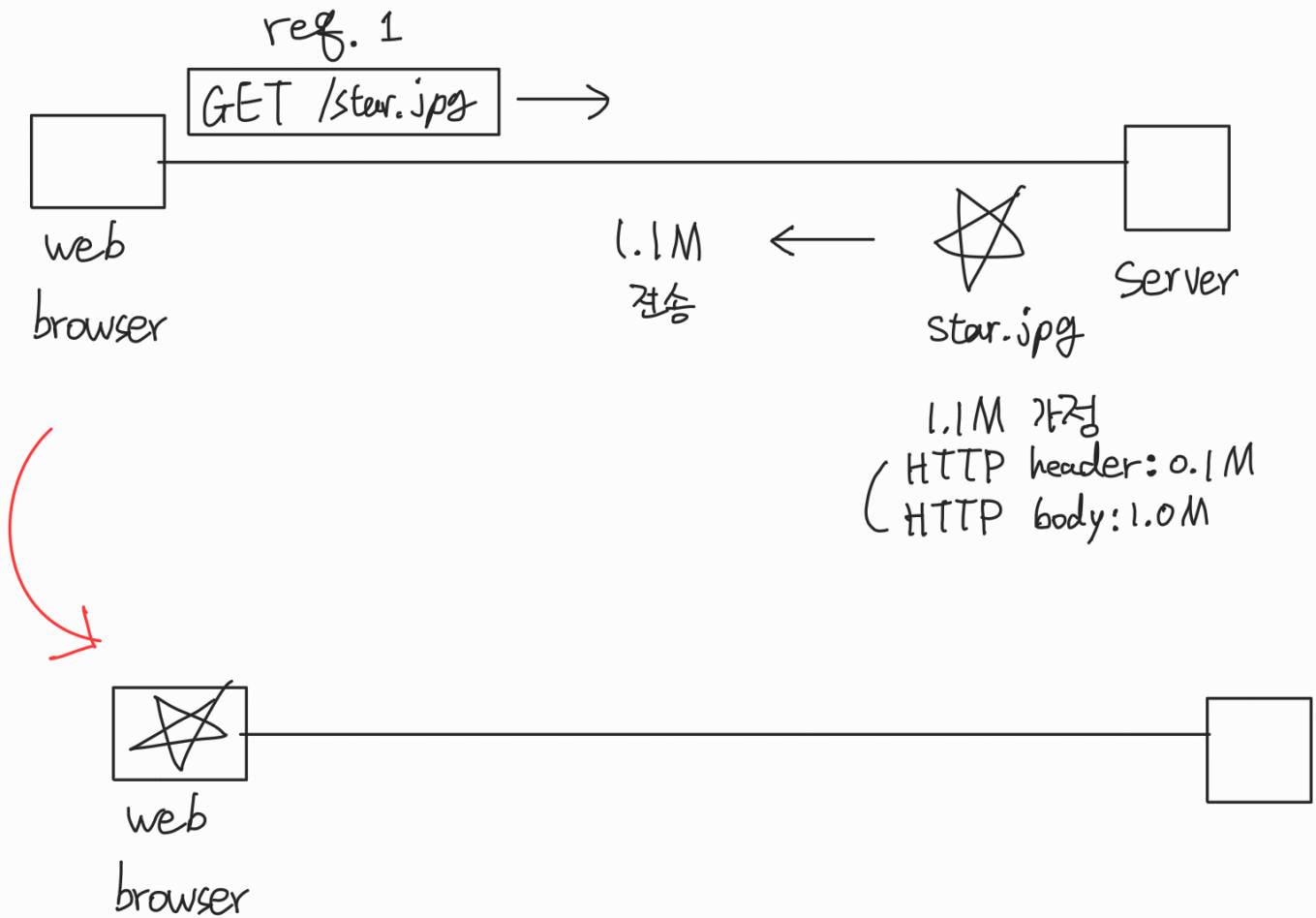


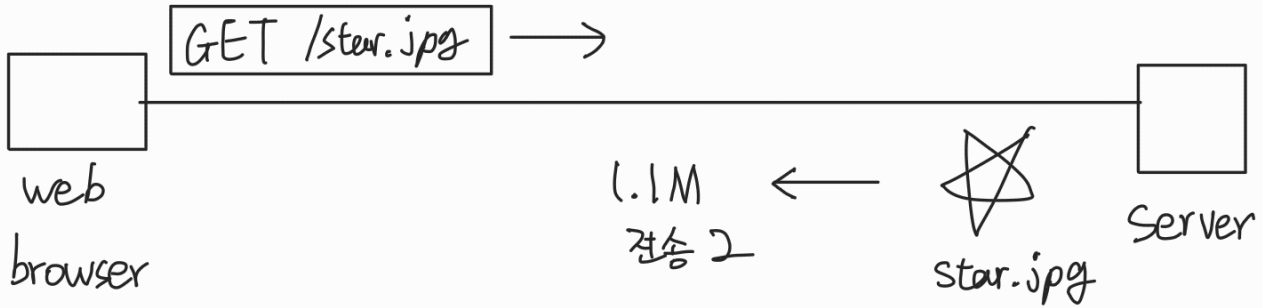
캐시 기본 동작

캐시가 없을 때 ; 첫 번째 요청



캐시가 없을 때 ; 두 번째 요청

req. 2 (똑같은 req. 한 번 더 함)



1.1M 가정
(HTTP header: 0.1M
HTTP body: 1.0M)



- data가 변경되지 않아도 계속 network 통해서 data down 받아야 됨
- Internet network 는 매우 느리고 비싸다
- browser loading 속도가 느리다.
- 느린 user experience.

캐시 적용; 첫 번째 요청

ex)

HTTP/1.1 200 OK

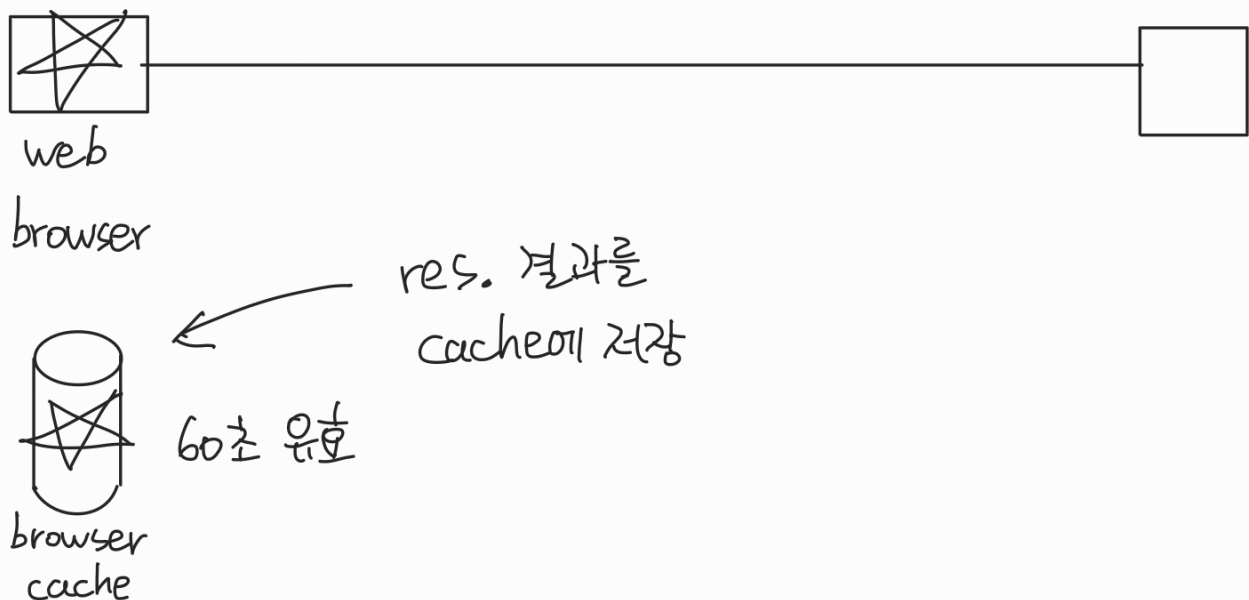
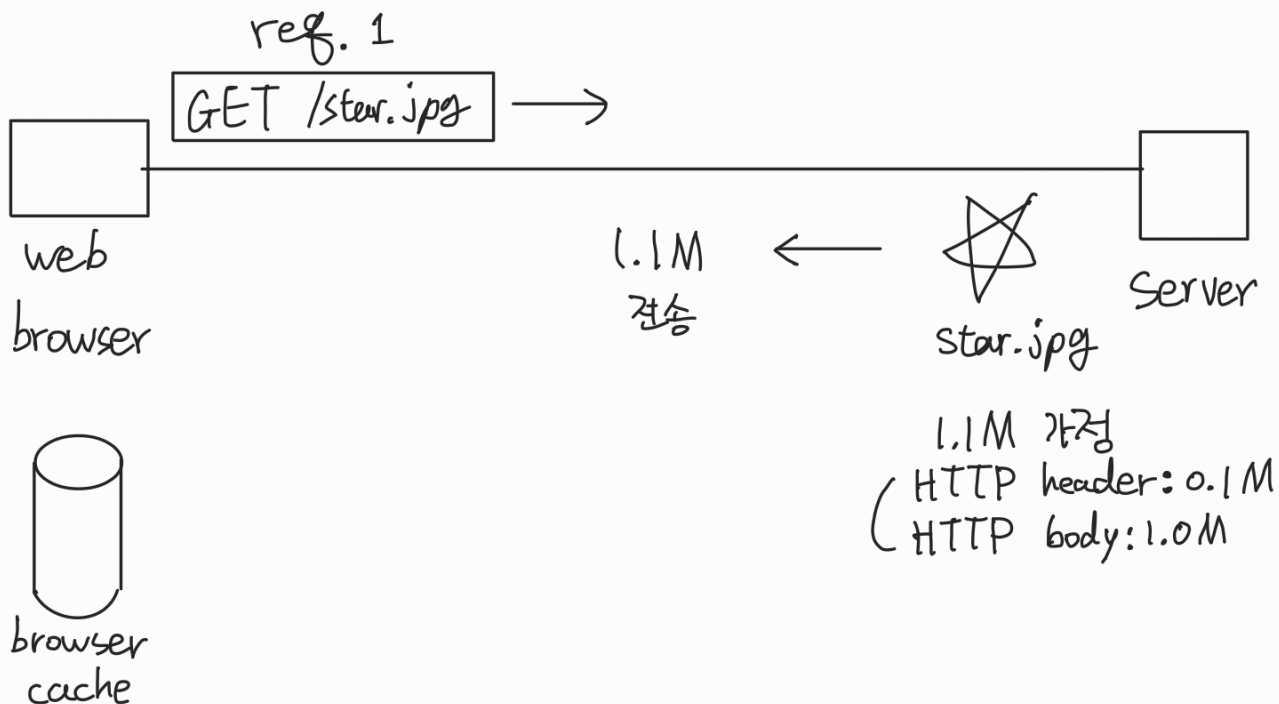
Content-Type: image/jpeg

cache-control: max-age=60

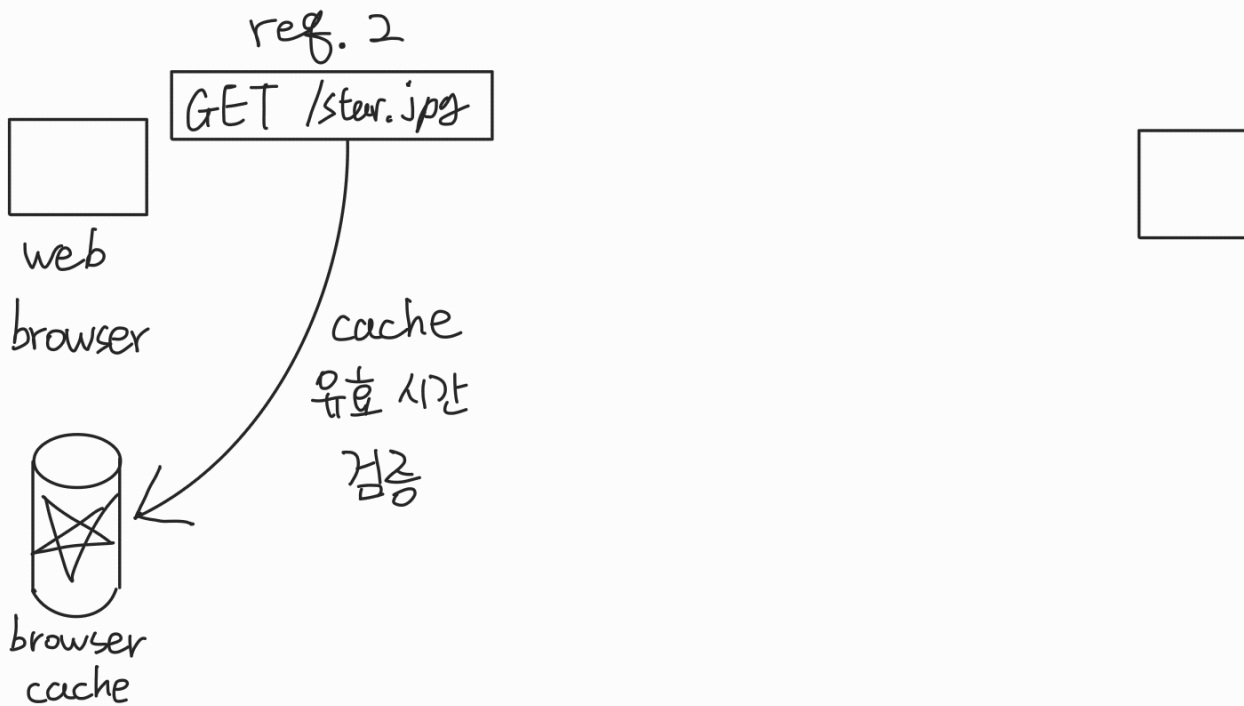
Content-Length: 34012

cache가

유효한 시간 (초)



캐시 적용 ; 두 번째 요청



- cache 덕분에 cache 가능 시간 동안 network 사용하지 않아도 된다.
- 비싼 network 사용량 줄일 수 있다.
- browser loading 속도가 매우 빠르다
- 빠른 user experience

캐시 적용 ; 세 번째 요청 - 캐시 시간 초과

req. 2

GET /star.jpg



web
browser

cache
유효 시간
검증



browser
cache

60초 초과

req. 3

GET /star.jpg

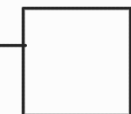


web
browser

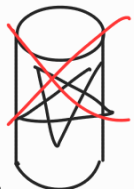
1.1M
전송



star.jpg



Server



browser
cache

60초 초과

1.1M 가중
(HTTP header: 0.1M
HTTP body: 1.0M



- cache 유효 시간이 초과되면,
server 통해 data 다시 조회하고
cache 갱신한다.

↳ 이때 다시 network down- 발생

But,

여기서 /star.jpg 는 내용이 바뀌