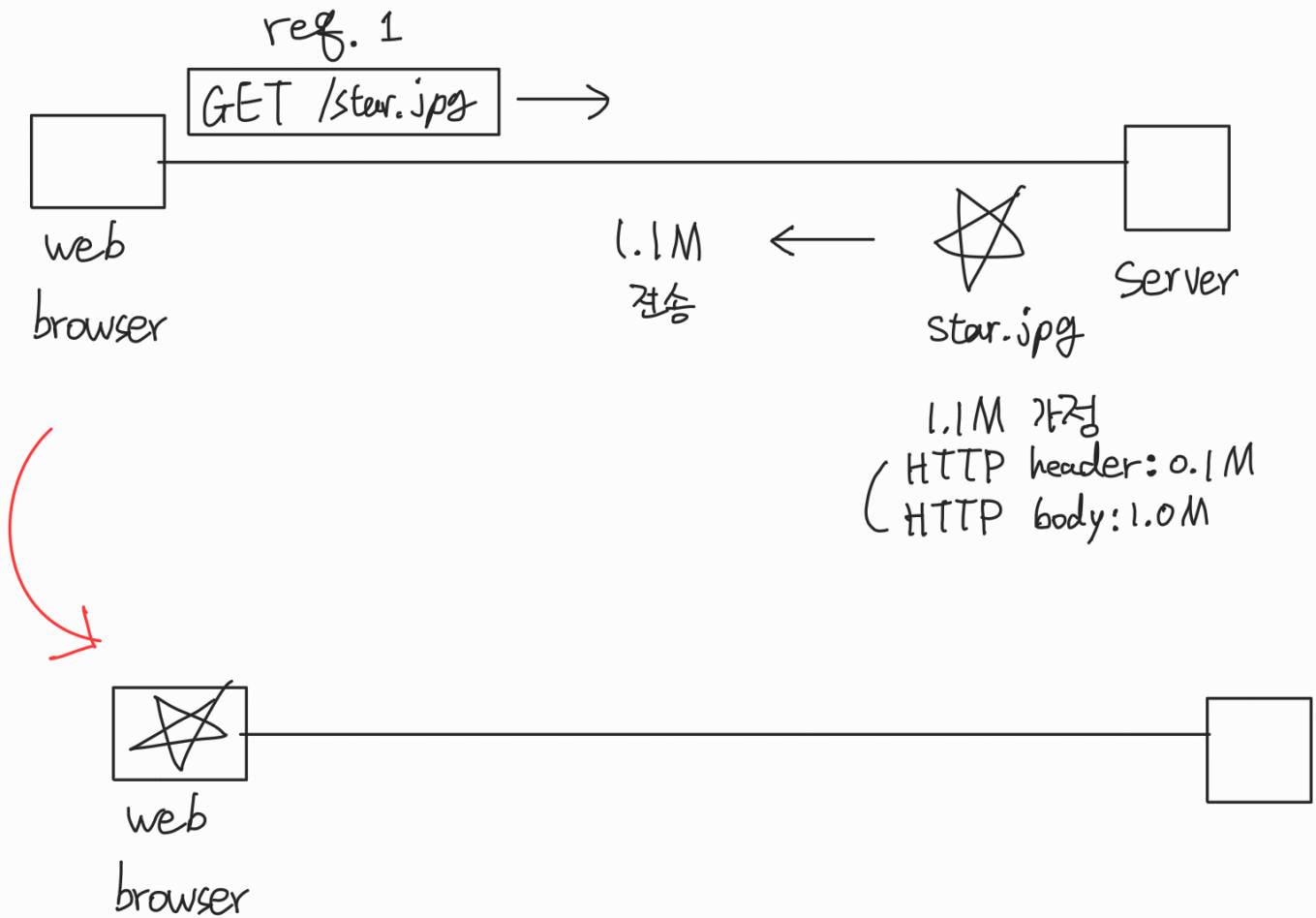


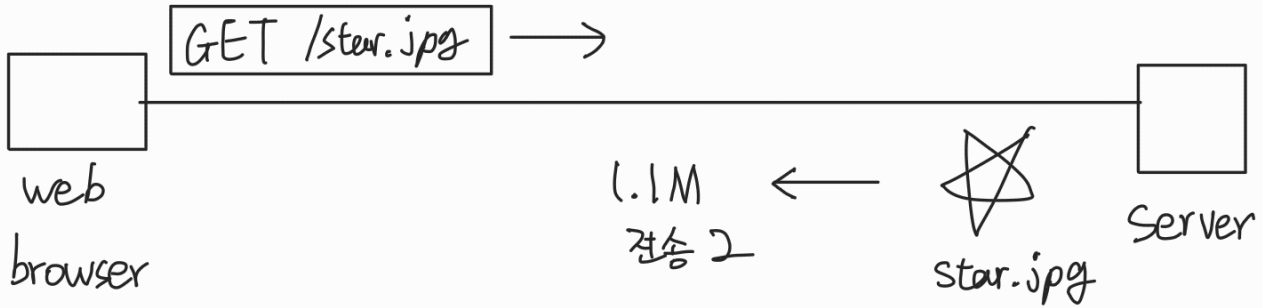
캐시 기본 동작

캐시가 없을 때 ; 첫 번째 요청



캐시가 없을 때 ; 두 번째 요청

req. 2 (똑같은 req. 한 번 더 함)



1.1M 가령
(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
```

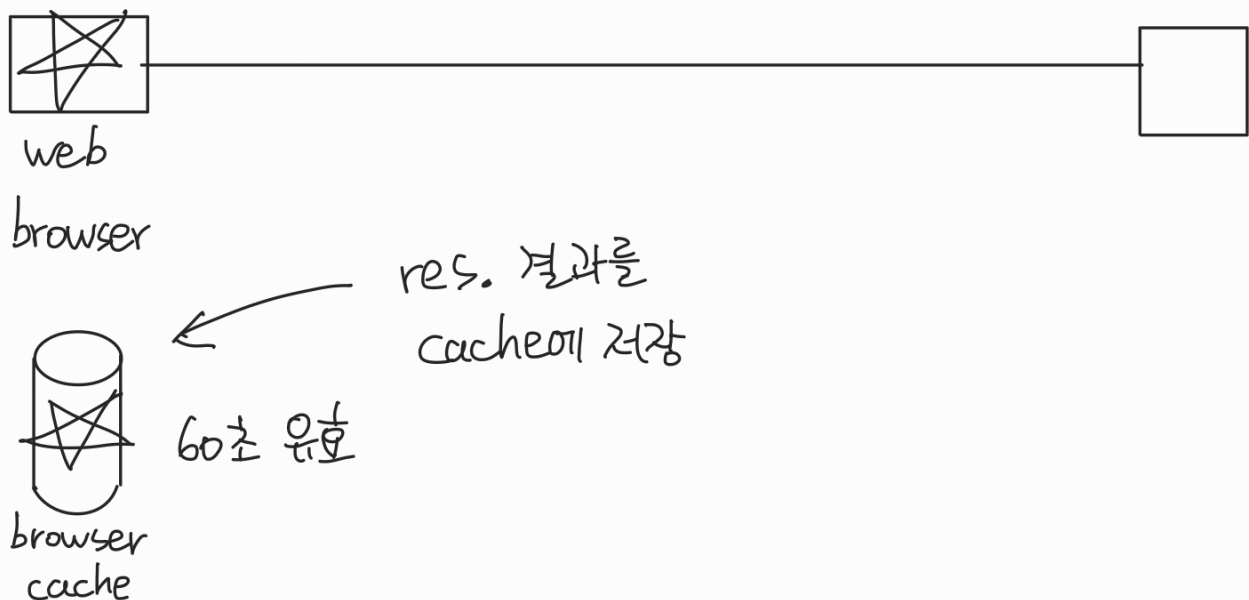
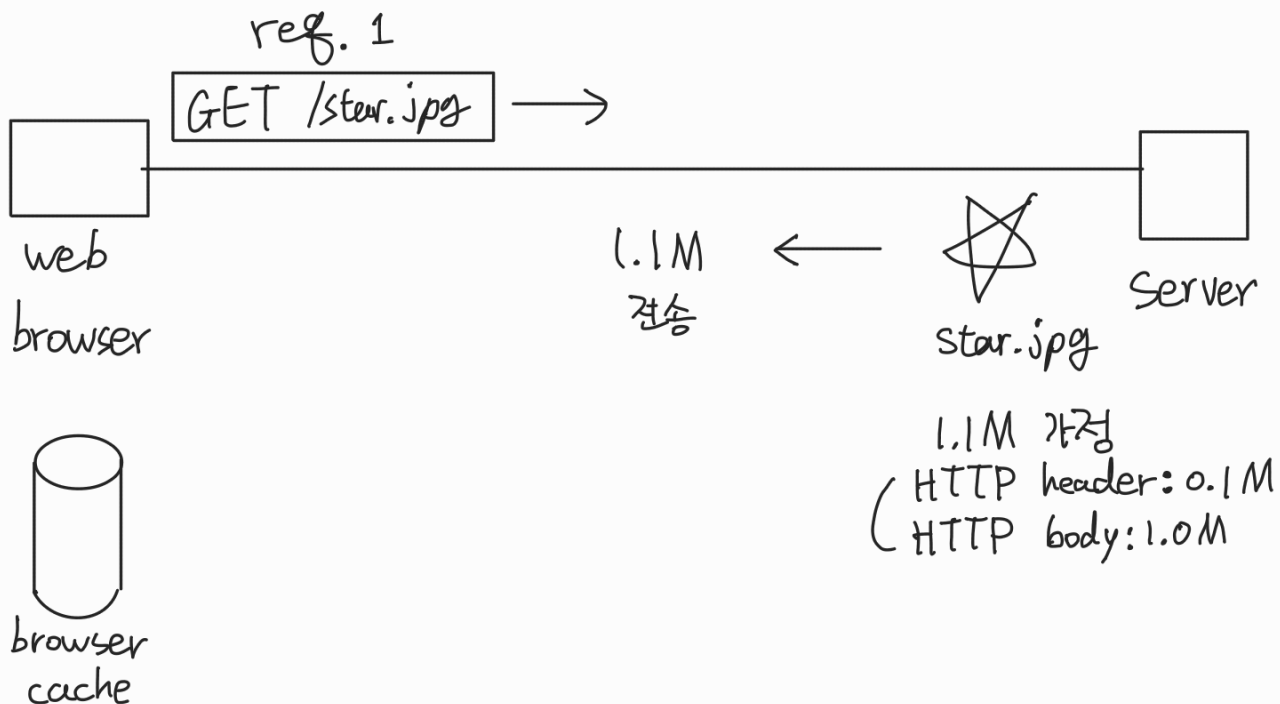
```
Content-Type: image/jpeg
```

```
cache-control: max-age=60
```

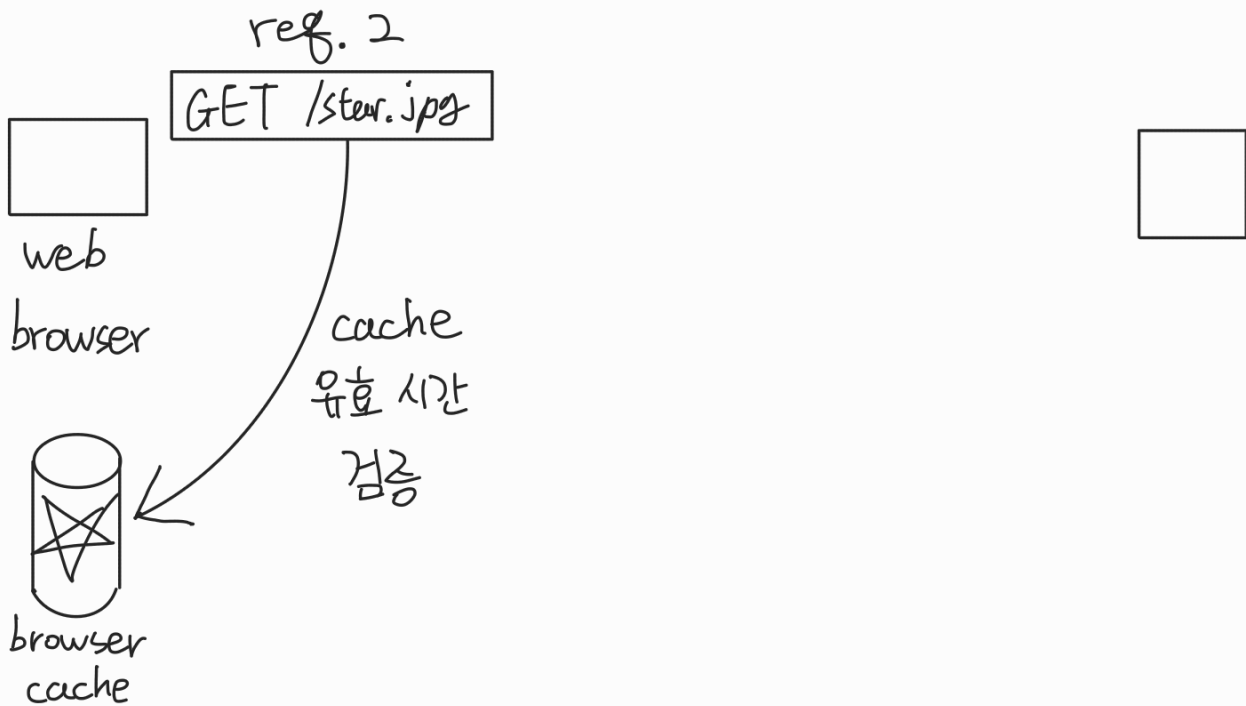
```
Content-Length: 34012
```

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

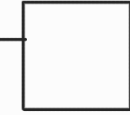


web
browser

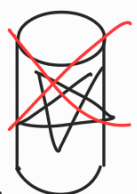
1.1M
전송



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

검증 헤더와 조건부 요청 1

캐시 시간 초과

- cache 유효 시간이 초과해서
server에 다시 req.하면

다음 2가지 상황 나타남

- ① serv.에서 기존 data 변경함

~~☆~~ → ☆

- ② serv.에서 기존 data 변경하지 않음

~~☆~~
↳ (기존의랑 똑같은 걸 굳이 또 down?)

캐시 시간 초과 ; 캐시 만료 후에도

서버에서 데이터 변경 X
(~~☆~~)

- data 전송 대신에,
저장해두었던 cache 재사용 가능
- 단, cli.의 data와
serv.의 data가 같다는 사실 확인하는
방법 필요

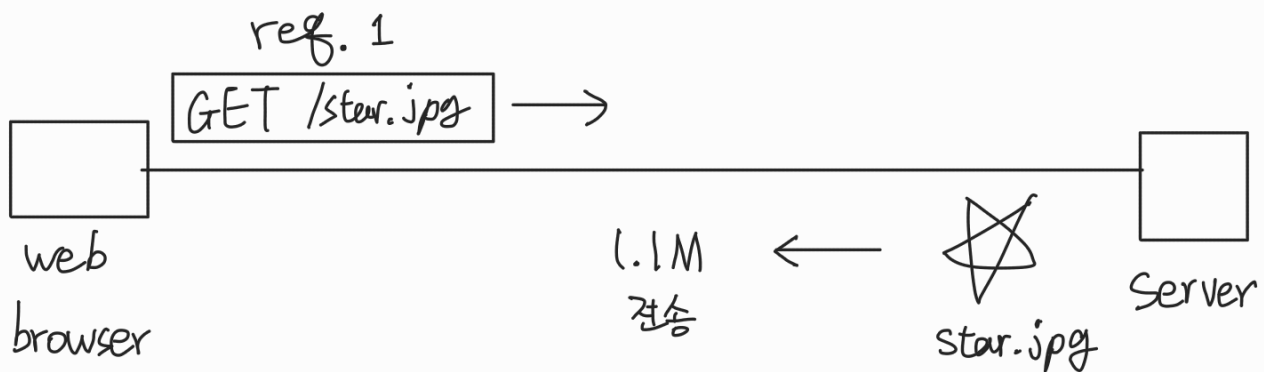
검증 헤더 추가

→ 첫 번째 req.

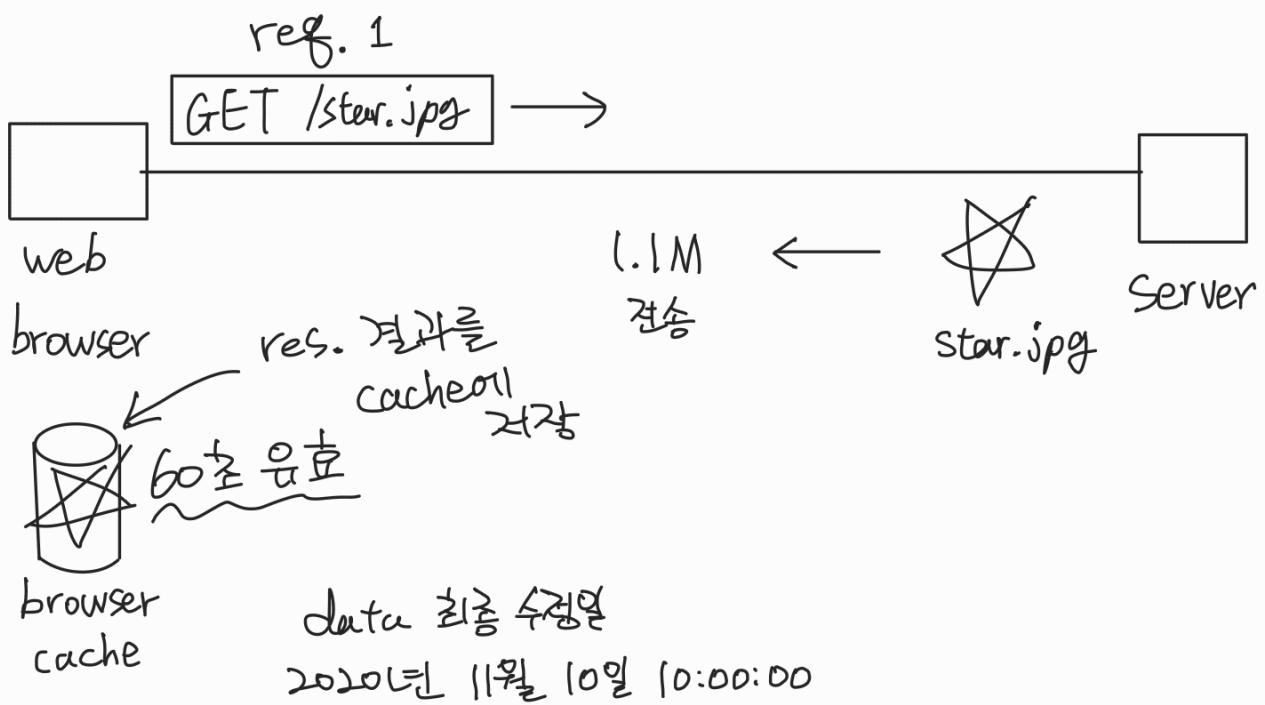
ex)

~~~~~  
~~~~~  
cache-control: max-age=60
Last-Modified: 2020년 11월 10일 10:00:00
~~~~~  
~~~~~

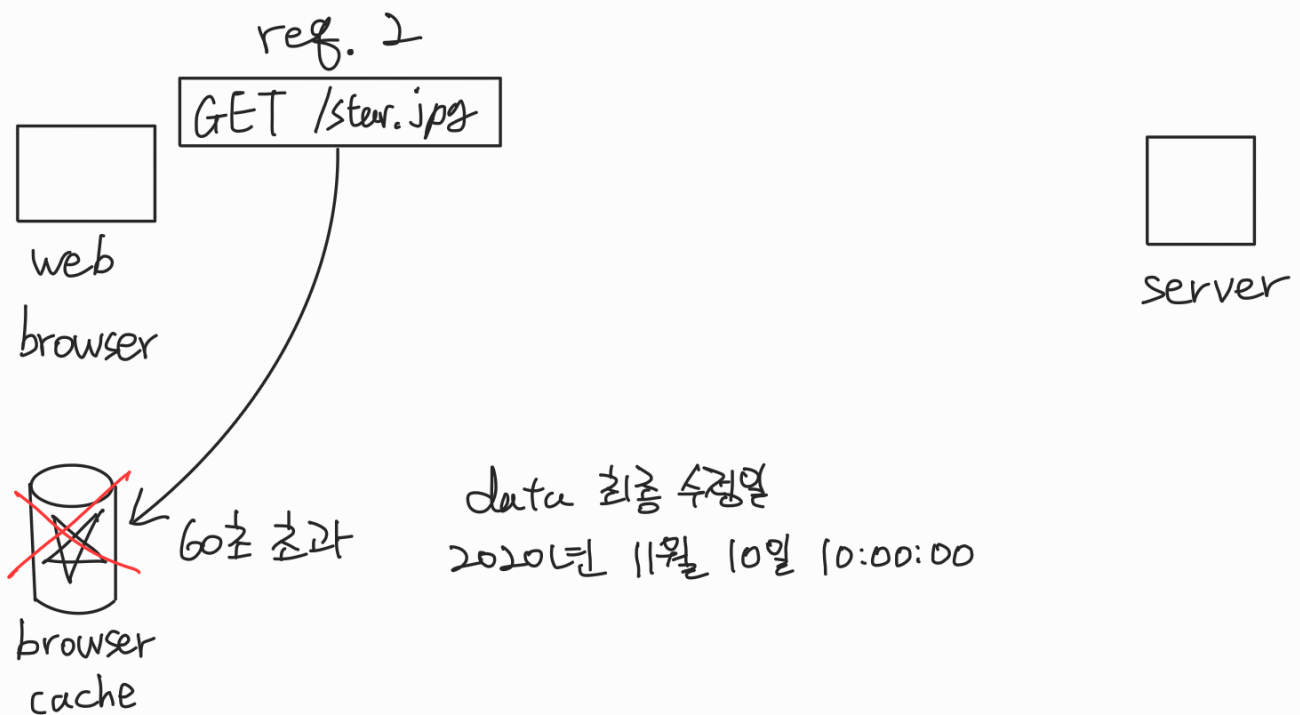
데이터가
마지막에
수정된 시간

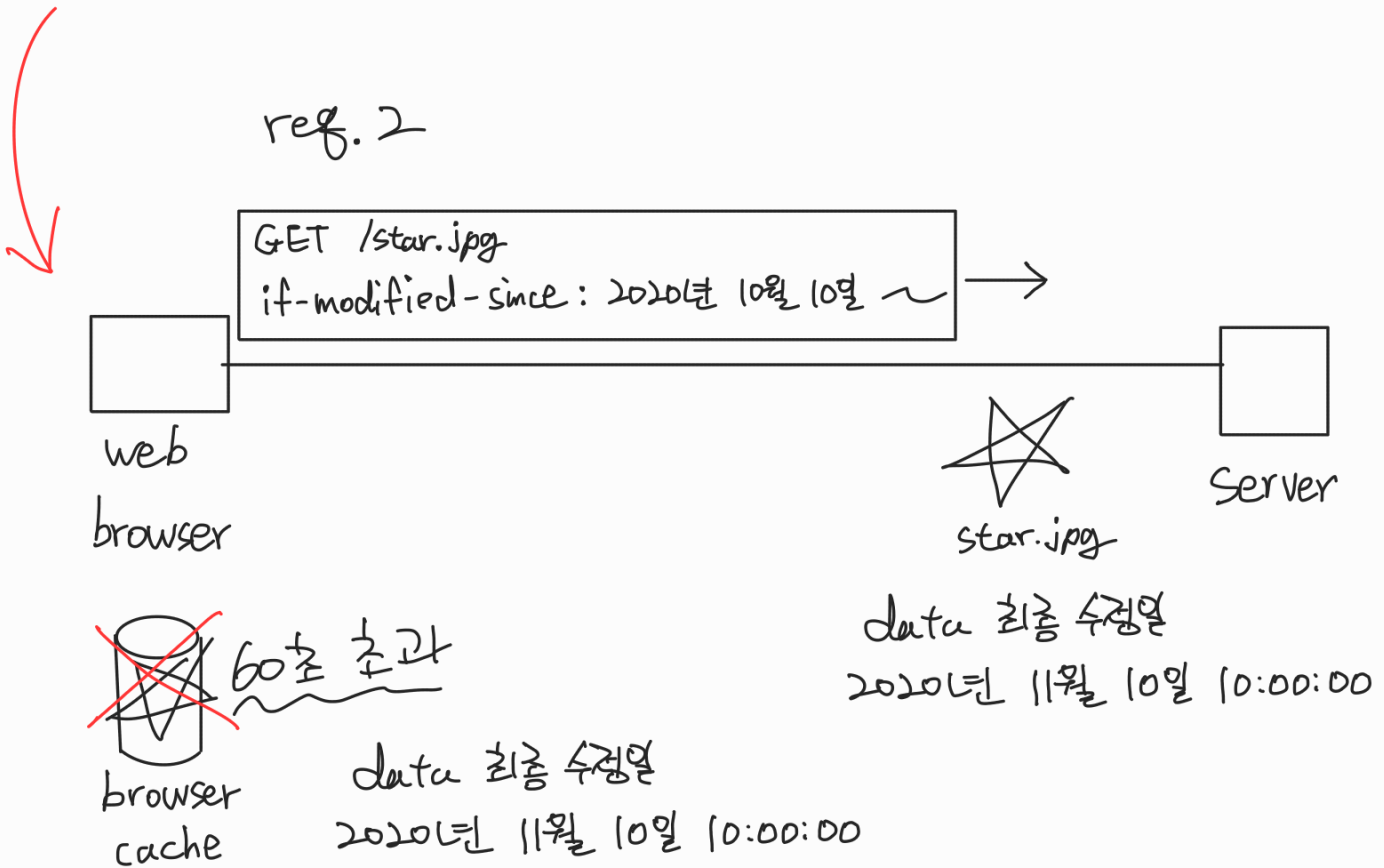
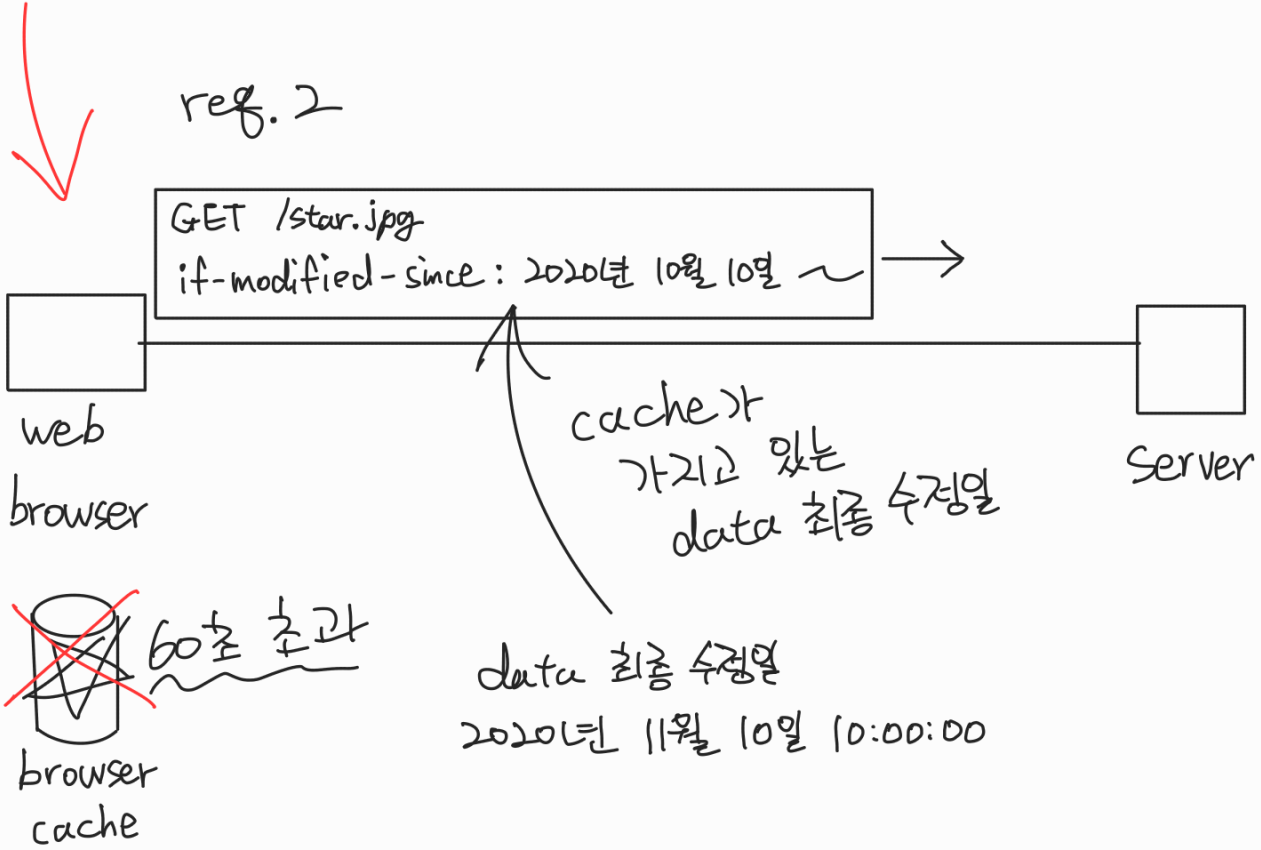


data 최종 수정일
2020년 11월 10일 10:00:00

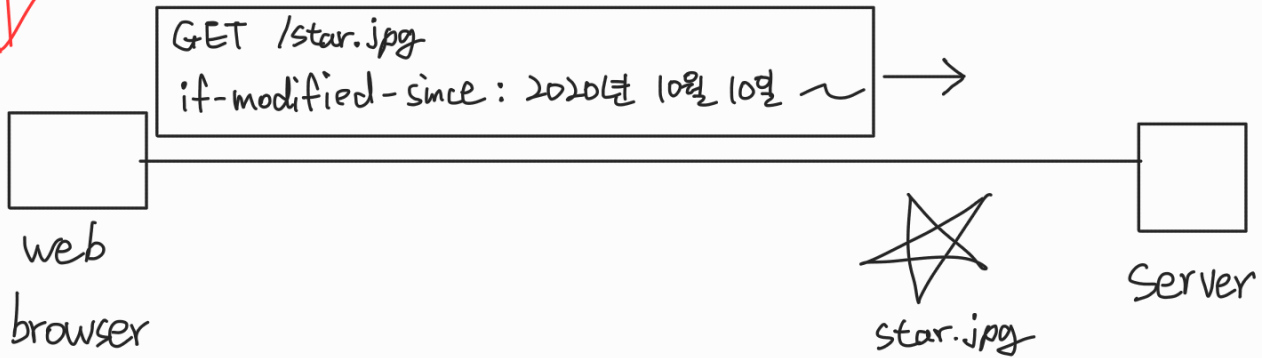


→ 두 번째 요청 ; 캐시 시간 초과





ref. 2

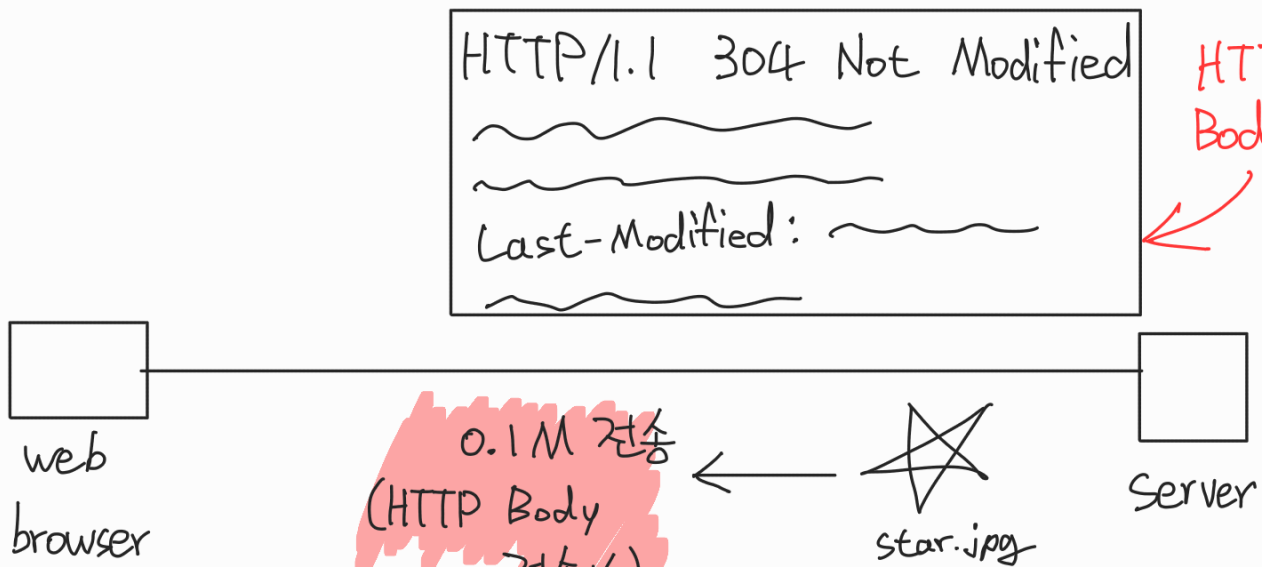


~~60초 초과~~
browser cache

data 최종 수정일
2020년 11월 10일 10:00:00

data 최종 수정일
2020년 11월 10일 10:00:00

data가 아직
수정되지 않았다.



HTTP
Body가 없음

0.1M 전송
(HTTP Body
전송X)

~~60초 초과~~
browser cache

data 최종 수정일
2020년 11월 10일 10:00:00

1.1M 가림
HTTP header: 0.1M
HTTP body: 1.0M

