

Http 06.

2022-07-21

캐시

캐시가 없다면?: 데이터가 변경되지 않아도 계속 네트워크를 통해서 데이터를 다운로드 받아야함, 인터넷 네트워크는 느리고 비싸다 효율성이 매우 떨어지게 된다.

캐시 적용

캐시 유효시간을 지정해서 응답하게 되면 응답 결과를 캐시에 유효시간만큼 저장한다.

만약 요청을 하면 우선 웹 브라우저는 캐시부터 찾아서 빠르게 응답해준다.

- 유효시간이 초과하면 다시 네트워크 다운로드가 발생한다. 이때 페이지 속 이미지등이 바뀌지 않았다면?

캐시 시간 초과

두가지 상황이 나타날 수 있다.

1. 다른 데이터 (기존과 다름)
2. 같은 데이터 (기존과 같음) - 비효율적
 - 데이터를 바꾸지 않았다면 저장해둔 캐시를 재사용하기 위해 검증 헤더를 추가해서 데이터가 바뀌지 않았다고 검증해준다. [Last-Modified : 2022-xx-xx]
 - 요청에 **if-modified-since**라는 헤더를 보내주면 서버는 **304 Not Modified**를 HTTP Body가 없는 응답을 보내준다. 응답이 되면 부하가 줄어들고 캐시속 응답결과를 재사용하면서 헤더를 갱신
 - 네트워크 다운로드가 발생하긴 하지만 헤더 정보만 다운로드 하기 때문에 굉장히 실용적인 해결책이 된다.

검증 헤더와 조건부 요청

If-Modified-Since

- 데이터가 변경이 안되면 : **304 Not Modified** 캐시로 리다이렉션
- 데이터가 변경이 되었다면 **200 OK** 모든 데이터 전송

단점 : 1초 미만 단위로 캐시 조정이 불가능, 날짜 기반의 로직 사용, 날짜가 다르지만 같은 결과로 수정을 했을 경우? 내부 컨텐츠가 일정할 때도 모든 데이터를 전송받는다.

ETag, If-None-Match

ETag(Entity Tag)

- 캐시용 데이터에 임의의 고유한 버전 이름을 달아둠
- Hash를 다시 생성하게 만든다.
- 그러므로 ETag만 보내주면 그것만 사용하면 된다. If-None-Match [이름 같으면 실패 : 304 캐시에서 가져옴][성공시 200 OK]

캐시 제어 헤더 종류

- Cache-Control

- max-age : 캐시 유효 시간, 초 단위
- no-cache : 데이터는 캐시해도 되지만 항상 원(origin) 서버에 검증하고 사용
- no-store : 데이터에 민감한 정보가 있으므로 저장하면 안됨 (ASAP 빨리 삭제)
- must-revalidate : 캐시 만료후 최초 조회시 원 서버에 검증해야함 , 실패시 반드시 오류가 발생해야한다.

- Expires

- 캐시 만료일을 지정 (하위 호환)

- Pragma

검증 헤더 (**Validator**) 종류

- **ETag** - [요청 : **If-Match** , **If-None-Match**]
- **Last-Modified** - [요청 : **If-Modified-Since** , **If Unmodified-Since**]

프록시 캐시

원 서버 직접 접근시 응답이 너무 느리다 그러므로 프록시 캐시 서버를 둔다

원 서버 접근 요청시 프록시 캐시 서버에서 가져오게 만든다. (CDN 서비스)

CDN - 콘텐츠 전송 네트워크 Content Delivery Network

캐시 지시어

public 캐시에 저장되어도 됨 private 응답이 사용자만을 위한 것 (기본)

s-maxage 프록시 캐시에만 적용되는 max-age , 프록시 캐시에 머무른 시간(Age)

캐시 무효화

캐시 무효화 응답

Cache-Control: no-cache, no-store, must-revalidate

Pragma : no-cache (하위 호환) -HTTP 1.0

를 전부 써주어야 확실하게 캐시 무효화 된다.

no-cache vs must-revalidate

항상 검증하는데 왜 같이 써주나 ?

no-cache 의 경우 프록시 서버가 원 서버로 넘긴다. 만약 no-cache 요청시 원서버가 데이터가 오류가 발생하는 것보다는 데이터를 보내는 것을 선택할 수도 있다.

그렇기 때문에 원 서버랑 통신이 문제가 생기면 **504 Gateway Timeout** 을 발생시키는 **must-revalidate** 를 사용해 준다.