HTTP 상태 코드

HTTP status code를 설명해 주세요.

[핵심 답변]

HTTP status code는 클라이언트가 보낸 HTTP 요청에 대한 서버의 응답 코드로,
상태 코드를 통해 요청의 성공/실패 여부를 판단할 수 있습니다.

100번대부터 500번대까지 총 5개의 클래스로 구분되어 HTTP 요청에 대한 상태를 알려줍니다.

종종 나오는 면접질문 중에 하나 입니다. status code무엇인지, 알고있는 status code를 모두 설명해 달라는 질문도 하곤 합니다. 자주 사용하는 대표적인 status code만 외워가도 충분한 답을 할 수 있습니다.
(200, 201, 400, 401, 403, 404, 500)

Status code

웹 개발시 서버와 클라이언트가 HTTP 통신할 때 주고받아야 할 값중에 하나입니다. 클라이언트로 부터 받은 request에 대한 서버의 response에 대한 간략할 설명 이라고 볼 수 있습니다. 상황에 알맞는 status code를 response에 담아서 클라이언트에 넘겨주면 이를 토대로 클라이언트는 알맞는 대응을 할 수 있습니다.
모든 HTTP status code는 5개의 클래스로 구분됩니다.

- 1xx (정보): 요청을 받았으며 작업을 계속한다.
- 2xx (성공): 클라이언트가 요청한 동작을 성공적으로 수신하여 이해했고 성공적으로 처리하였다.
- 3xx (리다이렉션): 요청을 완료하기 위해 추가 작업 조치가 필요하다.
- 4xx (클라이언트 오류): 클라이언트의 요청에 문제가 있다.
- 5xx (서버 오류): 서버가 유효한 요청의 수행을 실패했다.

[참고] 자주 등장하는 HTTP 응답코드

status code	message	
200	OK	요청이 성공함 (ex. 잔액조회 성공)
201	Created	리소스 생성 성공 (ex. 게시글 작성 성공, 회원가입 성공)
400	Bad Request	데이터의 형식이 올바르지 않는 등 서버가 요청을 이해할 수 없음 (ex. 올바르지 않은 형식의 데이터 입력 등)
401	Unauthorized	인증되지 않은 상태에서 인증이 필요한 리소스에 접근함 (ex. 로그인 전에 사용자 정보 요청 등)
403	Forbidden	인증된 상태에서 권한이 없는 리소스에 접근함 (ex. 일반 유저가 관리자 메뉴 접근 등)
404	Not Found	요청한 route가 없음. 찾는 리소스가 없음 (ex. www.naver.com/nossi 등 존재하지 않는 route에 요청 등)
502	Bad Gateway	서버에서 예상하지 못한 에러가 발생함 (ex. 예외처리를 하지 않은 오류가 발생 등)

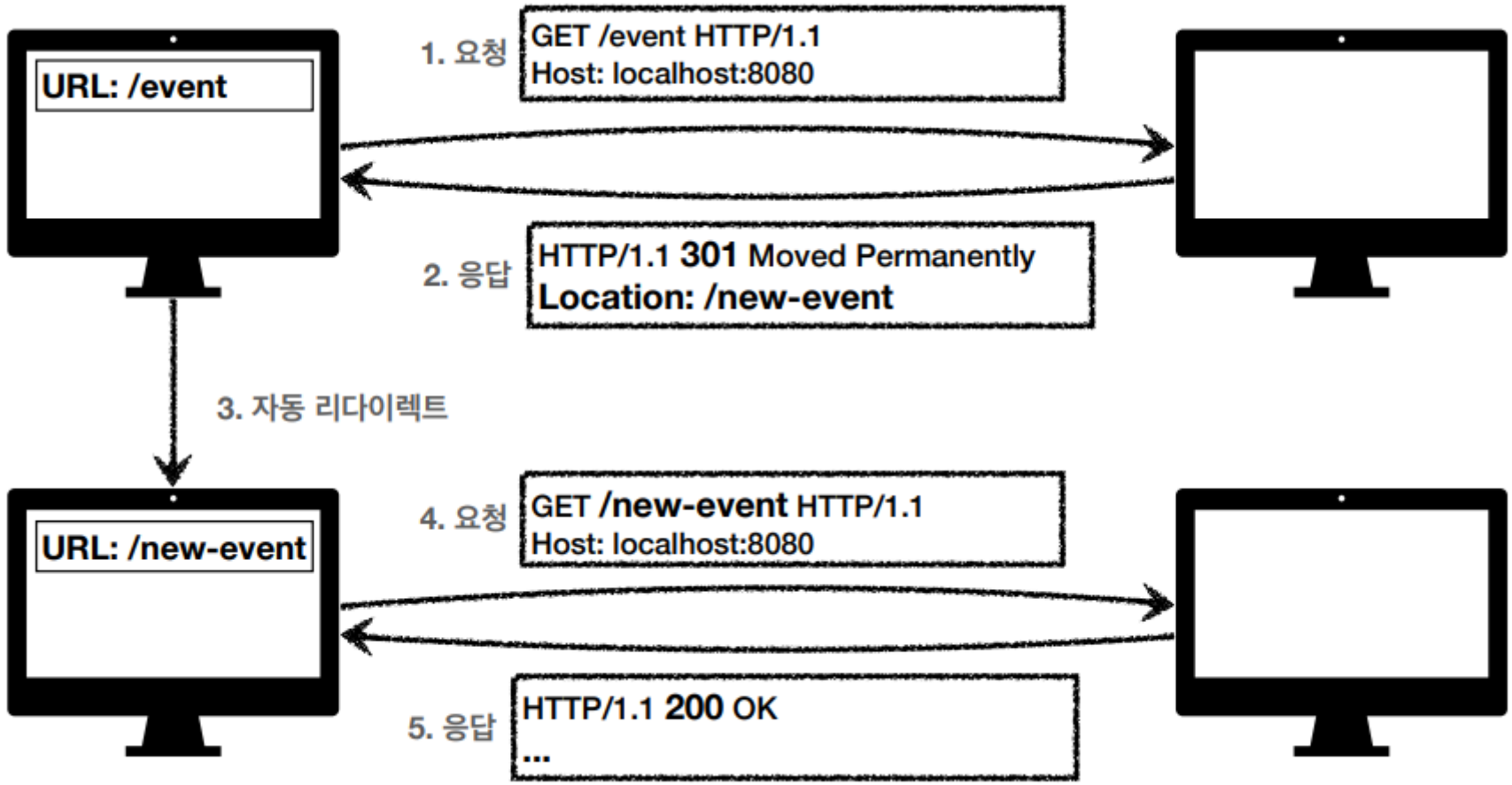
브라우저별 지원

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
100	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
103	✓ 103	✓ 103	✓ 120	✓ 89	✓ 17	✓ 103	? ?	? ?	? ?	? ?	? ?
rel=preconnect	✓ 103	✓ 103	✓ 120	✓ 89	✓ 17 *	✓ 103	? ?	? ?	? ?	? ?	? ?
rel=preload	✓ 103	✓ 103	✓ 123	✓ 89	✗ No	✓ 103	? ?	? ?	? ?	? ?	? ?
200	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
201	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
204	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
206	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
301	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
302	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
303	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
304	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
307	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
308	✓ 36	✓ 12	✓ 14	✓ 24	✓ 7	✓ 36	✓ 14	✓ 24	✓ 7	✓ 3.0	✓ 37
401	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
403	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
404	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
406	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
407	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
409	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
410	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
412	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
416	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
418	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
425	? ?	? ?	✓ 58	? ?	? ?	? ?	✓ 58	? ?	? ?	? ?	? ?
451	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
500	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
501	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
502	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
503	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
504	✓ Yes	✓ 12	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes

3XX

- 300 Multiple Choices
- 301 Moved Permanently
- 302 Found
- 303 See Other
- 304 Not Modified
- 307 Temporary Redirect
- 308 Permanent Redirect

웹 브라우저는 3xx 응답의 결과에 Location 헤더가 있으면, Location 위치로 자동 이동 (리다이렉트)


URL: /event
1. 요청
GET /event HTTP/1.1
Host: localhost:8080
2. 응답
HTTP/1.1 301 Moved Permanently
Location: /new-event
3. 자동 리다이렉트
URL: /new-event
4. 요청
GET /new-event HTTP/1.1
Host: localhost:8080
5. 응답
HTTP/1.1 200 OK
...

리다이렉션 이해
종류

- 영구 리다이렉션 - 특정 리소스의 URI가 영구적으로 이동
    - 예) /members -> /users
    - 예) /event -> /new-event
- 일시 리다이렉션 - 일시적인 변경
    - 주문 완료 후 주문 내역 화면으로 이동
    - PRG: Post/Redirect/Get
- 특수 리다이렉션
    - 결과 대신 캐시를 사용

영구 리다이렉션
301, 308

- 리소스의 URI가 영구적으로 이동
- 원래의 URL를 사용X, 검색 엔진 등에서도 변경 인지

- 301 Moved Permanently
- 리다이렉트시 요청 메서드가 GET으로 변하고, 본문이 제거될 수 있음(MAY)

- 308 Permanent Redirect
- 301과 기능은 같음
- 리다이렉트시 요청 메서드와 본문 유지(처음 POST를 보내면 리다이렉트도 POST 유지)

일시적인 리다이렉션
302, 307, 303

- 리소스의 URI가 일시적으로 변경
- 따라서 검색 엔진 등에서 URL을 변경하면 안됨

- **302 Found**
  - 리다이렉트시 요청 메서드가 GET으로 변하고, 본문이 제거될 수 있음(MAY)
- **307 Temporary Redirect**
  - 302와 기능은 같음
  - 리다이렉트시 요청 메서드와 본문 유지(요청 메서드를 변경하면 안된다. MUST NOT)
- **303 See Other**
  - 302와 기능은 같음
  - 리다이렉트시 요청 메서드가 GET으로 변경

그래서 뭘 써야 하나요?
302, 307, 303

- 정리
  - 302 Found -> GET으로 변할 수 있음
  - 307 Temporary Redirect -> 메서드가 변하면 안됨
  - 303 See Other -> 메서드가 GET으로 변경
- 역사
  - 처음 302 스펙의 의도는 HTTP 메서드를 유지하는 것
  - 그런데 웹 브라우저들이 대부분 GET으로 바꾸어버림(일부는 다르게 동작)
  - 그래서 모호한 302를 대신하는 명확한 307, 303이 등장함(301 대응으로 308도 등장)
- 현실
  - 307, 303을 권장하지만 현실적으로 이미 많은 애플리케이션 라이브러리들이 302를 기본값으로 사용
  - 자동 리다이렉션시에 GET으로 변해도 되면 그냥 302를 사용해도 큰 문제 없음

기타 리다이렉션
300 304

- 300 Multiple Choices: 안씀
- 304 Not Modified
    - 캐시를 목적으로 사용
    - 클라이언트에게 리소스가 수정되지 않았음을 알려준다. 따라서 클라이언트는 로컬PC에 저장된 캐시를 재사용한다. (캐시로 리다이렉트 한다.)
    - 304 응답은 응답에 메시지 바디를 포함하면 안된다. (로컬 캐시를 사용해야 하므로)
    - 조건부 GET, HEAD 요청시 사용