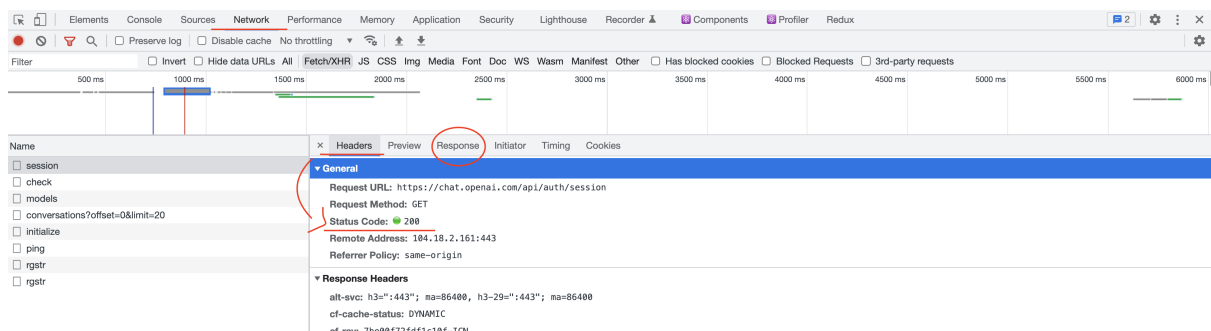
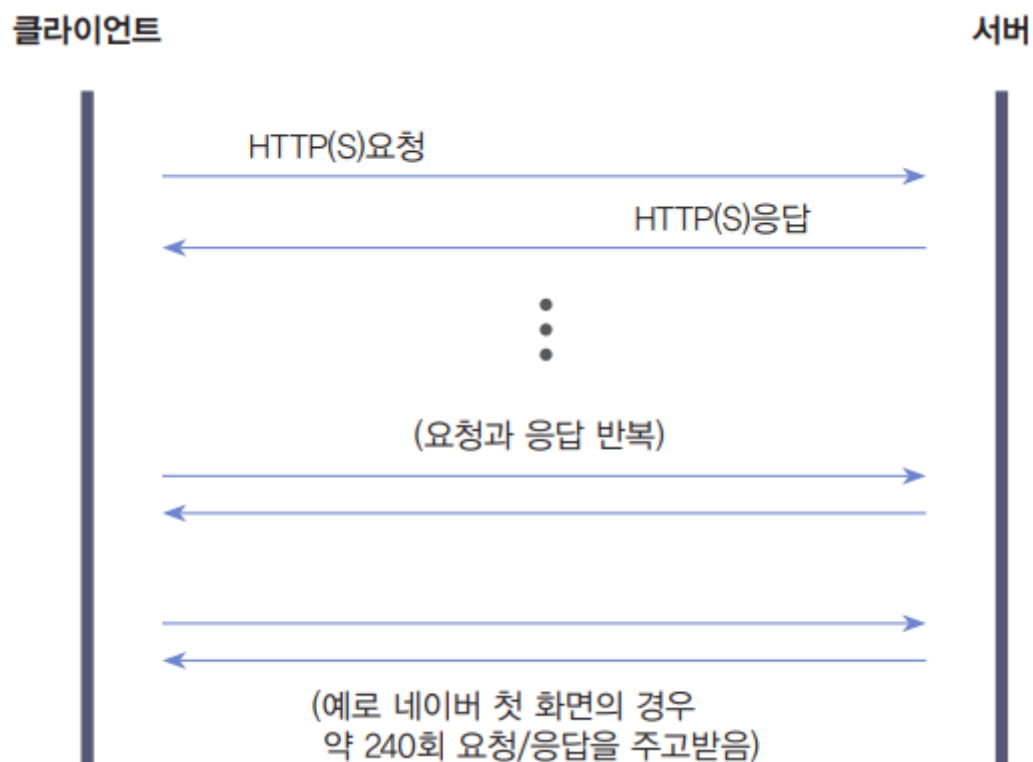


# HTTP Status Code

## HTTP Status Code란?

클라이언트가 서버에 요청을 보냈을 때, 서버가 응답하는 상태를 나타내는 세자리 숫자로 이루어진 코드이다. 이 코드는 클라이언트에게 요청이 성공했는지 실패했는지, 그리고 실패했다면 어떤 이유로 실패했는지를 알려준다.



개발자도구에서 Network에 응답코드는 Headers에서 확인가능하며 응답완료된 내용은 Response에서 확인 할 수 있다.

1xx (Informational) - 요청이 수신되어 처리 중임을 나타낸다.

2xx (Successful) - 요청이 성공적으로 처리되었음을 나타낸다.

3xx (Redirection) - 클라이언트가 추가 작업을 수행해야 함을 나타낸다.

4xx (Client Error) - 클라이언트의 요청이 잘못되었거나 처리할 수 없음을 나타낸다.

5xx (Server Error) - 서버에서 요청을 처리할 수 없음을 나타낸다.

---

1XX - 정보 전달: 요청을 받았고, 작업을 진행중이다.

100 Continue - 클라이언트가 서버로 보낸 요청에 문제가 없으니 다음 요청을 이어서 보내도 된다는 것을 의미한다. 일시적인 상태 코드이며, 클라이언트가 요청을 계속 보낼 수 있음을 나타내기 때문에 다음 상태 코드인 200 OK와 함께 사용된다.

101 Switching Protocols - 클라이언트가 Upgrade 헤더를 통해 요청한 것에 따라 서버가 프로토콜을 바꾼다는 것을 알려주는 코드다.

102 Processing - 사용자가 수신 요청을 하여 처리하고는 있지만 아직은 제대로 된 응답을 할 수 없다. 예를 들어, 클라이언트가 대용량 파일 업로드 요청을 서버에 보냈다면, 서버는 이 요청을 처리하는 동안 중간 응답으로 HTTP 상태 코드 102를 보낼 수 있다. 이 상태 코드는 서버에서 파일 업로드 작업이 계속 진행 중임을 알리며, 클라이언트는 이 상태 코드를 수신하면 요청을 계속 보낼 수 있다.

---

2XX - 성공: 작업을 성공적으로 처리하였다.

**200 OK** - 서버에서 클라이언트 요청에 대한 성공적인 응답을 나타내는 상태 코드이다. 이러한 상태 코드는 클라이언트가 요청한 작업이나 리소스를 성공적으로 처리했으며, 요청된 데이터가 응답 본문에 포함되어 있다는 것을 의미한다.

서버에서 클라이언트 요청에 대한 성공적인 응답을 나타내는 상태 코드이다. 이러한 상태 코드는 클라이언트가 요청한 작업이나 리소스를 성공적으로 처리했으며, 요청된 데이터가 응답 본문에 포함되어 있다는 것을 의미한다.

프로그래머라면 가장 흔하게 볼 수 있는 응답 코드이다.

**201 Created** - 요청이 성공적으로 처리되어 리소스가 생성되었다.

예를들어 POST요청을 해서 새로운 게시글을 생성했다면 새로운 리소스가 생성된것이므로 201응답을 보낸다.

이때 응답과 함께 uri를 제공하는데, 이 uri는 생성된 자원을 조회하는데 사용된다.

**202 Accepted** - 요청이 성공적으로 수신되었으며, 서버에서 처리가 진행 중임을 의미한다. 서버는 작업이 완료되면 클라이언트에 대한 응답을 생성하거나, 다른 상태 코드를 반환할 수도 있다.

클라이언트와 서버 간의 비동기 통신에서 자주 사용된다. 이러한 상태 코드는 클라이언트가 요청을 보낸 후 서버에서 수행되는 작업이 시간이 오래 걸리거나, 작업이 완료되기 전에 다른 요청이 처리되는 경우에 사용되며 클라이언트는 작업이 완료될 때까지 기다려야 함을 알 수 있다.

**204 No Content** - 요청을 받아들여 성공적으로 처리했지만 콘텐츠를 제공하지는 않는다.

이러한 상태 코드는 일반적으로 DELETE 요청이나 PUT 요청으로 콘텐츠를 업데이트할 때 사용된다.

**206 Partial Content** - 콘텐츠의 일부 부분만 제공한다.클라이언트가 대용량 파일을 다운로드할 때 사용된다. 예를 들어, 클라이언트가 1GB의 파일을 다운로드하려고 할 때, 서버는 이 파일을 100MB 단위로 분할하여 전송할 수 있다. 이 경우, 클라이언트는 서버에서 반환된 각 부분을 받은 후, 이들을 병합하여 전체 파일을 생성할 수 있다.

**207 Multi-Status** 다수의 동작이 일어난 클라이언트에서 볼 수 있으며, 여러 응답인 상태에서 적절한 정도를 사용자에게 제공하도록 하는 응답코드다.이러한 응답은 XML 또는 JSON 과 같은 데이터 형식으로 반환된다. 멀티 상태 응답은 WebDAV (Web Distributed Authoring and Versioning)와 같은 특수한 프로토콜에서 사용되며, 일반적인 HTTP 요청 및 응답에서는 잘 사용되지 않는다.

---

**3XX: 리다이렉션** - 이 요청을 완료하기 위해서는 리다이렉션이 이루어져야 한다.

**301 Moved Permanently** - 클라이언트가 요청한 리소스가 완전히 새로운 위치로 옮겨졌을 경우 사용됩니다. 이 경우, 클라이언트는 새로운 위치의 URL을 사용하여 리소스에 접근해야 합니다.

**302 Found** - 일시적으로 콘텐츠가 이동하였다.

**303 See Other** - 서버가 사용자의 GET요청을 처리하여 다른 URL에서 요청된 정보를 가져올 수 있도록 응답한다.

**304 Not Modified** - "수정되지 않음(Not Modified)"을 나타낸다. 이 코드는 클라이언트의 캐시된 버전이 여전히 최신인 경우, 서버가 요청된 리소스를 변경하지 않았음을 알리기 위해 사용된다.

일반적으로 브라우저 캐시와 함께 사용된다. 클라이언트가 요청한 페이지나 파일이 수정되지 않았다면, 서버는 304 코드를 반환하여 캐시된 버전을 사용하도록 허용한다. 이렇게 함으로써 불필요한 대역폭 사용을 줄일 수 있다.

예를 들어, 사용자가 인터넷 브라우저에서 웹 페이지를 방문했을 때, 브라우저는 이미 해당 페이지의 캐시된 버전을 가지고 있는 경우 서버에 새로운 버전을 요청하기보다 캐시된 버전을 사용할 수 있다. 이 경우 서버는 304 코드를 반환하여 브라우저가 캐시된 페이지를 사용하도록 허용한다.

**305 Use Proxy** - 말 그대로. 프록시를 사용하지 않으면 접근할 수 없다. 다만 보안상의 이유로 현재는 사용되지 않는 응답 코드다.

**307 Temporary Redirect** - 요청한 리소스가 일시적으로 다른 URL로 이동되었지만, 클라이언트는 이전 URL을 계속 사용해야 하는 경우에 사용된다.

307 코드는 302 코드와 유사하지만, 다른 점은 클라이언트가 이전 URL을 계속 사용해야 한다는 것이다. 302 코드는 클라이언트가 새 URL을 사용하도록 요청하지만, 307 코드는 클라이언트가 이전 URL을 사용하도록 요청한다.

307 코드는 브라우저 캐시와 함께 사용된다. 예를 들어, 웹 서버가 유지 보수 중이며 일시적으로 다른 서버로 이동해야 할 경우, 서버는 307 코드를 반환하여 브라우저가 이전 URL을 계속 사용하도록 요청한다. 이렇게 함으로써, 클라이언트는 이전 URL에서 캐시된 리소스를 계속 사용할 수 있으며, 서버는 유지 보수 작업을 수행하면서 새 URL로 이동할 수 있다.

---

4XX 클라이언트 오류: 요청이 올바르지 않다.

**400 Bad Request** (잘못된 요청) - 클라이언트 측에서 전송한 요청이 잘못되었음을 나타내는 상태 코드이다.

예를 들어, 요청 매개변수가 잘못된 경우, 필수 매개변수가 누락된 경우, 잘못된 요청 형식을 사용한 경우, 요청 URI가 잘못된 경우 등이 있다.

예시)

```
POST /api/create_user HTTP/1.1
Host: example.com
Content-Type: application/json
Content-Length: 34

{ "username": "john_doe", "email": "" }
```

위의 요청에서는 email 필드에 빈 문자열이 전송되어 HTTP 상태 코드 400이 반환된다. 서버는 이 요청을 처리할 수 없기 때문에, 클라이언트는 서버에서 반환한 오류 메시지를 기반으로 요청을 수정해야 한다.

**401 Unauthorized(권한 없음):** 인증이 실패한 경우를 나타내는 상태 코드로 클라이언트가 요청한 자원에 대한 인증이 필요하다는 것을 나타낸다. 예를 들어, 요청한 자원에 대한 인증 정보가 없는 경우, 잘못된 인증 정보를 제공한 경우, 만료된 인증 토큰을 사용한 경우 등이 있으며, 서버와 클라이언트 간의 통신에서 매우 중요한 역할을 한다.

403과의 차이점은 403의 경우 단순히 권한이 없음을 의미한다.

**402 Payment Required(결제 필요):** 비용이 발생하는 서비스에 대한 요청을 처리할 수 없음을 나타내는 상태 코드, 클라이언트가 사용 중인 서비스가 유료 서비스인 경우, 사용량에 따라 요금이 부과될 수 있다. 이러한 경우, 402를 반환하여 클라이언트가 서비스 사용에 필요한 요금을 지불하도록 유도할 수 있다. 상태 코드 중에서는 그다지 많이 사용되지 않는 편이며, 대부분의 서비스에서는 요금 부과와 관련된 내용을 다른 방식으로 처리한다.

**403 Forbidden(거부됨 혹은 금지됨):** 서버가 클라이언트의 요청을 처리하지 않을 것임을 나타낸다. 클라이언트가 요청한 자원에 대한 권한이 없거나, 서버에서 이를 거부한 경우에 HTTP 상태 코드 403이 반환한다. 서버와 클라이언트 간의 권한 및 보안 정책을 위해 사용된다.

만약 사용자가 권한이 없는 자원에 접근을 시도할 경우, 서버는 HTTP 상태 코드 403을 반환할 수 있다. 예를 들어, 관리자 권한을 가진 사용자만 접근 가능한 자원에 일반 사용자가 접근하려고 시도한 경우, 서버는 다음과 같은 응답을 반환할 수 있다.

**404 Not Found(찾을 수 없음):** 클라이언트가 요청한 자원을 서버가 찾을 수 없을 때 반환된다. 즉, 요청받은 URL에 해당하는 리소스가 서버에 존재하지 않을 때 사용된다.

일반적으로 사용자가 올바른 URL을 입력했지만, 그 URL에 해당하는 리소스가 존재하지 않는 경우 발생한다. 예를 들어, 잘못된 URL을 입력하거나, 파일이 삭제된 경우, 혹은 파일이

옮겨진 경우 등이 해당되며, 이 응답은 요청한 리소스를 찾을 수 없다는 것을 알리며, 사용자에게 이를 알리기 위해 "Not Found" 메시지와 함께 설명을 제공한다.

**405 Method Not Allowed**(허용되지 않은 메소드): 클라이언트 요청이 서버에서 허용되지 않는 HTTP 메서드를 사용했을 때 반환된다. 즉, 요청한 URL은 존재하지만, 사용한 HTTP 메서드가 지원되지 않을 때 발생한다.

**406 Not Acceptable**(받아들일 수 없음): 서버가 클라이언트의 요청을 수락할 수 없는 응답 형식을 요구했을 때 반환된다. 즉, 클라이언트가 Accept 헤더에 지정한 미디어 타입이 서버에서 지원되지 않을 때 발생한다.

**408 Request Timeout**(요청 시간 초과): 말 그대로. 요청 중 시간이 초과하였다.

**409 Conflict**(충돌): 클라이언트 요청이 서버 상태와 충돌하여 요청이 수행되지 않았음을 나타낸다. 대개 충돌은 리소스 상태가 요청된 조작과 일치하지 않는 경우에 발생한다.

서버 상태와 충돌이 발생하는 경우는 주로 다음과 같은 상황에서 발생한다.

- 다른 클라이언트 또는 프로세스가 동일한 리소스에 대한 동시 요청을 보냈고, 이로 인해 리소스의 상태가 충돌 상태가 되었다.
- 클라이언트가 요청한 리소스를 수정하려고 했으나, 그 동안 리소스의 상태가 변경되었거나 다른 클라이언트가 이미 해당 리소스를 수정한 경우.
- 요청한 작업이 리소스 상태와 일치하지 않는 경우.

이러한 경우에 서버는 클라이언트에게 409 Conflict 상태 코드를 반환하여 충돌 상황임을 알리고, 클라이언트는 요청을 수정하여 충돌을 해결해야 한다.

**410 Gone**(사라짐): 요청한 리소스가 서버에서 영구적으로 삭제되었음을 나타냅니다. 이 상태 코드는 404 상태 코드와는 달리 해당 리소스가 존재하지 않는 것이 아니라, 서버에서 영구적으로 삭제되었으므로 더 이상 요청을 처리할 수 없다는 것을 나타낸다.

**411 Length Required**(길이 필요): 사용자가 서버로 요청할 때 헤더에 Content-Length 필드 값이 없을 경우 발생하는 오류다.

**412 Precondition Failed**(전제조건 실패): 사용자가 서버로 조건부 요청을 할 때 서버와 사용자의 전제조건이 일치하지 않아 발생하는 오류.

**413 Request Entity Too Large**(요청된 것이 너무 큼): 이 코드는 클라이언트가 서버에 보내는 요청의 크기가 서버에서 처리할 수 있는 최대 크기를 초과했을 때 발생한다.

이러한 상황에서는 클라이언트가 요청의 크기를 줄이거나, 서버 측에서 처리할 수 있는 요청 크기를 늘리도록 구성해야한다.

414 Request-URL Too Long:클라이언트가 서버로 보낸 요청의 URL이 너무 길때 요청이 처리되지 않았음을 알린다.

415 Unsupported Media Type(지원하지 않는 미디어 타입): 사용자가 요청한 미디어타입이 서버에서 지원하지 않는다.

416 Requested Range Not Satisfiable(요청된 범위가 만족스럽지 않음): 요청 헤더의 Range로 지정한 범위가 잘못되었다.

417 Expectation Failed(예측 실패): 요청 헤더의 Expect값이 서버에서는 적절하지 못하다.

418 I'm a teapot(찻주전자로 커피를 만들 수 없음):

이건 그냥 만우절 이벤트로 만든 장난 응답코드

임. <https://www.google.com/teapot> <https://tools.ietf.org/html/rfc2324>

이래 보여도 엄연한 표준 응답코드임.

420 Enhance Your Calm: 트위터에서만 쓰는 비표준 응답코드로, 뜻은 후술할 429와 같음.

421 Misdirected Request(잘못된 요청): 요청이 응답을 생성할 수 없는 서버로 전달되었다.

422 Unprocessable Entity(처리할 수 없는 개체): 요청 자체는 정상적으로 수리되었으나, 문법 오류로 인하여 응답할 수 없다.

423 Locked(잠김): 요청한 리소스가 잠겨있다.

424 Failed Dependency(실패한 종속성): 이전의 요청이 실패한 상태에서 지금의 요청도 실패했다.

425 Too Early(너무 일찍 요청): 이게 뭔지는 나도 잘 모르겠다.

426 Upgrade Required(업그레이드 필요): 클라이언트에서 보낸 요청의 프로토콜이 맞지 않아 현재 서버에서 처리할 수 없다.

**429 Too Many Requests(너무 많은 요청):** 클라이언트가 일정 시간 동안 너무 많은 요청을 보낸 경우에 반환된다.

서버에서 클라이언트의 요청을 처리할 수 있는 능력을 초과했을 때, 클라이언트에게 잠시 대기하라는 신호를 보내는 역할을 하며, 이것은 서버의 과부하를 줄이기위해 설정될 수있는 요청 제한기능을 가르킨다.

예를 들어, 1분에 100개의 요청을 처리할 수 있는 서버에 클라이언트가 1분에 150개의 요청을 보낸 경우, 서버는 429 응답을 반환하여 클라이언트가 일정 시간 동안 요청을 중단하도록 한다.

431 Request Header Fields Too Large(요청 헤더 필드가 너무 큼): 요청한 헤더 값이 너무 커서 서버에서 처리를 하지 않는다.

451 Unavailable for Legal Reasons(법적인 이유로 차단됨): 국가 검열에 의거하여 차단되었다. 유래는 당연히 미국 소설 화씨 451.

---

5XX(서버 오류): 오류코드인것은 4XX와 같다. 차이점은 4XX와 달리 요청은 올바르지만 이에 대해 서버가 응답할 수 없음을 뜻한다.

**500 Internal Server Error**(내부 서버 에러): 서버에 오류가 발생해 작업을 수행할 수 없다. 주로 서버 측 프로그램의 오류나 서버 자원의 문제로 발생합니다. 예를 들어, 데이터베이스 연결 오류, 메모리 부족, 파일 시스템 오류 등이 이에 해당한다.

**501 Not Implemented**(요청한 기능 미지원): 요청된 기능이 서버에 구현되어 있지 않은 경우에 사용된다.

즉, 클라이언트가 요청한 메서드(예: GET, POST)나 헤더 필드(예: If-Match, Range) 등이 서버에서 지원되지 않을 때 501 코드가 반환된다.

**502 Bad Gateway**: 서버와 게이트웨이 사이에서 발생한 문제로 인해 서버가 클라이언트 요청을 처리할 수 없는 경우에 사용한다.

일반적으로 게이트웨이는 서버와 클라이언트 간의 트래픽을 중개하고 프로토콜을 변환하는 역할을 한다. 게이트웨이는 클라이언트로부터의 요청을 서버로 전달하고, 서버로부터의 응답을 클라이언트에게 전달한다. 만약 게이트웨이에서 서버로의 요청이 실패하거나, 서버에서 응답을 받지 못한 경우, 502 코드가 발생한다.

**503 Service (Temporarily) Unavailable**(서비스를 일시적으로 이용할 수 없음): 말 그대로.

**504 Gateway Timeout**(게이트웨이 시간초과): 이것도 말 그대로.

**506 Variant Also Negotiates**: 서버 내부 구성값에 오류가 있어 반환된다.

**507 Insufficient Storage**: 서버 내부 구성에 오류가 있다.

**508 Loop Detected**: 서버가 요청을 처리하는 동안 무한 루프를 발견하였다.

이것이 무슨 뜻인지는 나도 모르겠다.



510 Not Extended: 서버가 요청을 처리할 때 요청에 대한 추가 확장이 필요하다.

:

511 Network Authentication Required: 사용자가 네트워크 액세스 권한이 필요하다.