

P R E S E N T A T I O N

# 34주차 주제 예외와 예러 처리

~ 에러 상태코드 ~

by mun

# Error

## 에러 / 오류

- 에러 또는 오류는 부정확하거나 유효하지 않은 동작

## HTTP

- HyperText Transfer Protocol
- 웹 환경에서 정보를 주고받기 위한 프로토콜

## HTTP 상태 코드란?

- HTTP 요청이 성공했는지 실패했는지를 서버에서 알려주는 코드
- 클라이언트는 HTTP의 상태 코드를 확인하여 요청의 성공/실패를 확인할 수 있음

상태 코드로 에러 상황을 파악 가능

# 4XX: Client Error

클라이언트의 원인으로 에러가 발생

- 종류가 많다

상태 코드	상태 텍스트	한국어 뜻	서버 측면에서의 의미
4XX	Client Error	클라이언트 에러	클라이언트의 요청에 오류가 있다.
400	Bad Request	잘못된 요청	요청의 구문이 잘못되었다. 클라이언트가 모르는 4xx 계열 응답 코드가 반환된 경우에도 클라이언트는 400과 동일하게 처리하도록 규정하고 있습니다.
401	Unauthorized	권한 없음	지정한 리소스에 대한 액세스 권한이 없다. 응답 헤더 WWW-Authenticate에 필요한 인증 방식을 지정합니다.
402	Payment Required	결제 필요	지정한 리소스를 액세스하기 위해서는 결제가 필요하다. 이 응답 코드는 실제로는 사용되지 않습니다.
403	Forbidden	금지됨	지정한 리소스에 대한 액세스가 금지되었다. 401 인증 처리 이외의 사유로 리소스에 대한 액세스가 금지되었음을 의미합니다. 리소스의 존재 자체를 은폐하고 싶은 경우는 404 응답 코드를 사용할 수 있습니다.
404	Not Found	찾을 수 없음	지정한 리소스를 찾을 수 없다.
405	Method Not Allowed	허용되지 않은 메소드	요청한 URI가 지정한 메소드를 지원하지 않는다. 응답 헤더 Allow에 이 URI가 지원하는 메소드 목록을 기록합니다.
406	Not Acceptable	수용할 수 없음	클라이언트가 Accept-* 헤더에 지정한 항목에 관해 처리할 수 없다. 응답 바디에는 300 응답처럼 서버가 수용 가능한 다른 선택지 리스트가 기록됩니다.

# 400 Bad Request

클라이언트의 요청이 유효하지 않아 더 이상 작업을 진행하지 않는 경우

## 유효성 검증 가능

1. 필수 여부
2. 유효 여부 (범위, 패턴 등)

```
HTTP/1.1 400 Bad Request
```

어디가 문제지 ?

# 400 Bad Request

클라이언트의 요청이 유효하지 않아 더 이상 작업을 진행하지 않는 경우

## 유효성 검증 가능

1. 필수 여부
2. 유효 여부 (범위, 패턴 등)

```
HTTP/1.1 400 Bad
```

```
HTTP/1.1 400 Bad Request
{
  "errors": [
    {
      "location": "body",
      "param": "name",
      "value": 123,
      "error": "TypeError",
      "msg": "must be String"
    }
  ]
}
```

오류 발생 시의 파라미터의 위치, 사용자 입력 값, 에러 이유를 꼭 명시하는 것이 좋다

# 401 Unauthorized, 403 Forbidden

인증과 권한

## Unauthorized 401

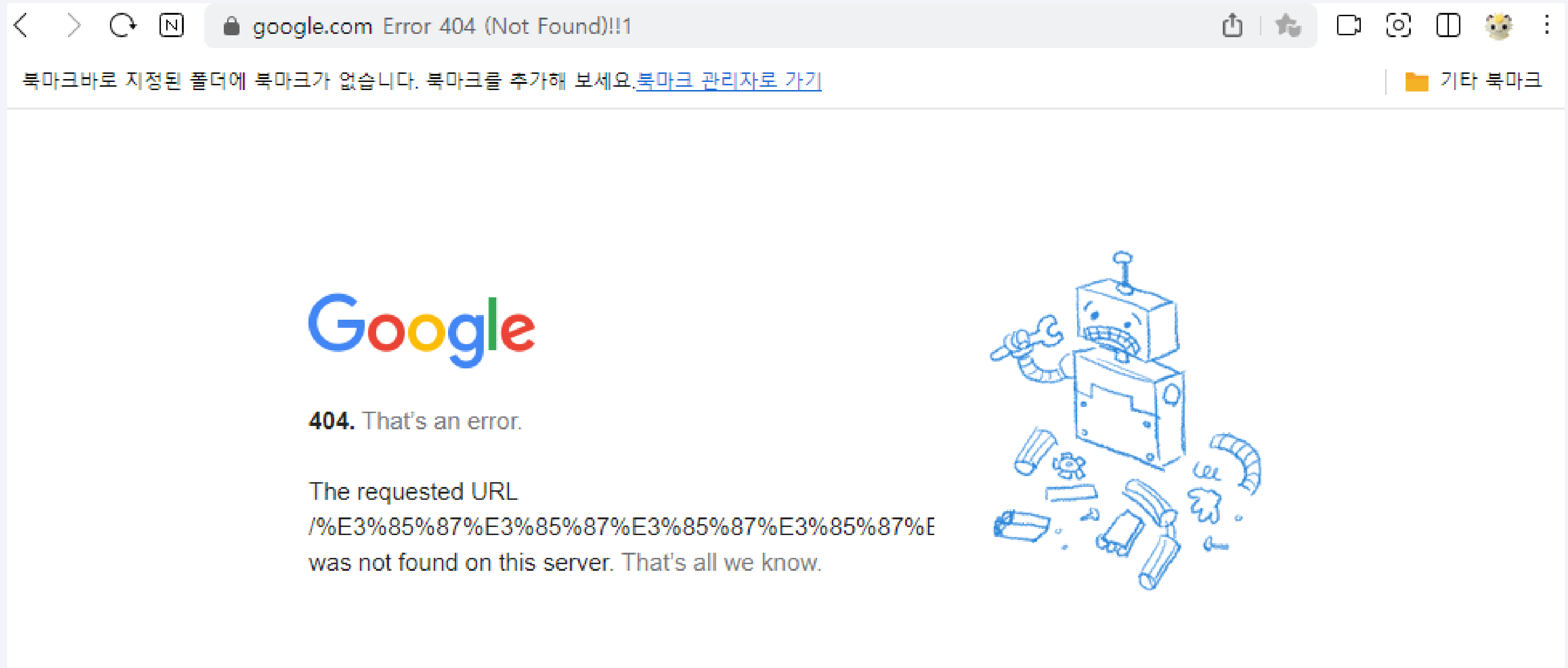
- 인증이 안돼 자원을 이용할 수 없는 상태
- 로그인이 안된 상태

## Forbidden 403

- 클라이언트가 권한이 없기 때문에 작업을 진행할 수 없는 경우
- 로그인 된 상태지만, 해당 권한이 없는 경우

# 404 Not Found

클라이언트가 요청한 자원이 존재하지 않는 경우



# 404 Not Found

클라이언트가 요청한 자원이 존재하지 않는 경우

```
GET /users/1 HTTP/1.1
```



자원 1이 없는 경우 404

- GET /users/:id 라는 경로는 존재
- id 자원이 있는지를 파악



# 405 Method Not Allowed

클라이언트의 요청이 허용되지 않는 메소드인 경우

## Http Method

- GET
- POST
- DELETE
- PUT
- 등등...

```
GET /users/1 HTTP/1.1
```

유저 1이 없는 경우 404

```
POST /users/1 HTTP/1.1
```

제공하지 않는 메소드 = 405

# 409 Conflict

클라이언트의 요청이 서버의 상태와 충돌이 발생한 경우

- 앞의 상태 코드들과는 달리 애매...
- 많이 정의되는 규칙: 해당 요청의 처리 중 비즈니스 로직상 불가능하거나 모순이 생긴 경우

```
DELETE /users/1 HTTP/1.1
X-TOKEN: password
```

- 자원 존재
- 형식 유효
- 사용자 존재
- 인증 완료
- 권한 있음

근데... 해당 사용자가 작성한 글이 있는 경우  
사용자를 삭제할 수 없어서 불가능

비즈니스 로직상의 문제



# 5XX Server errors

서버 오류로 인해 요청을 수행할 수 없다는 의미

- 앞 클라이언트의 요청에는 문제가 없지만, 작업 중 문제가 생긴 경우
- 일어나면 안됨... 개발자의 문제
- 그냥 5xx 반환이 안되도록 설계를 잘 해야됨

```
POST /users HTTP/1.1
{
  "name": 123
}
```

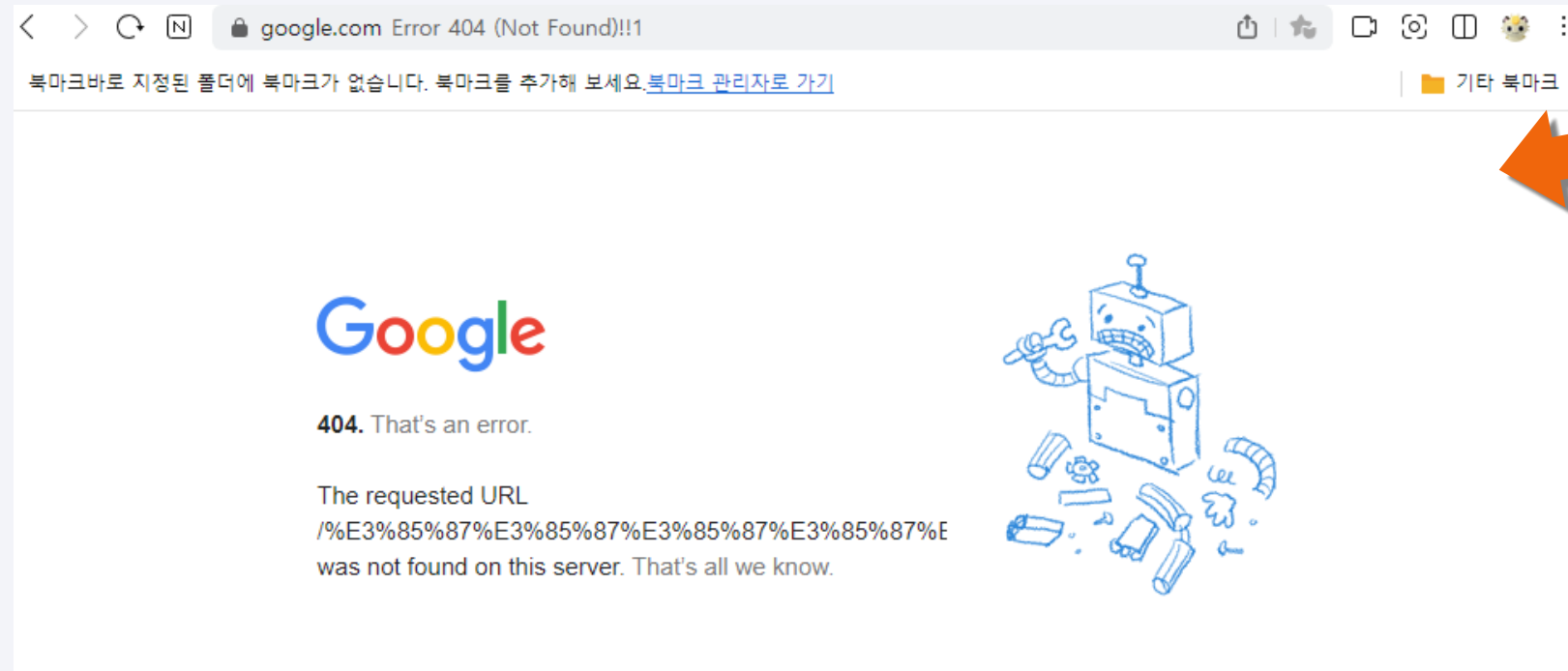
유효성 검사를 하지 않는다면 서버에서 오류가 발생

완벽한 예외처리를 통해  
5xx 상태를 방지해야 한다!

# 생각해 볼 점: 데이터가 없는 경우의 상태코드

## 404 Not Found

클라이언트가 요청한 자원이 존재하지 않는 경우



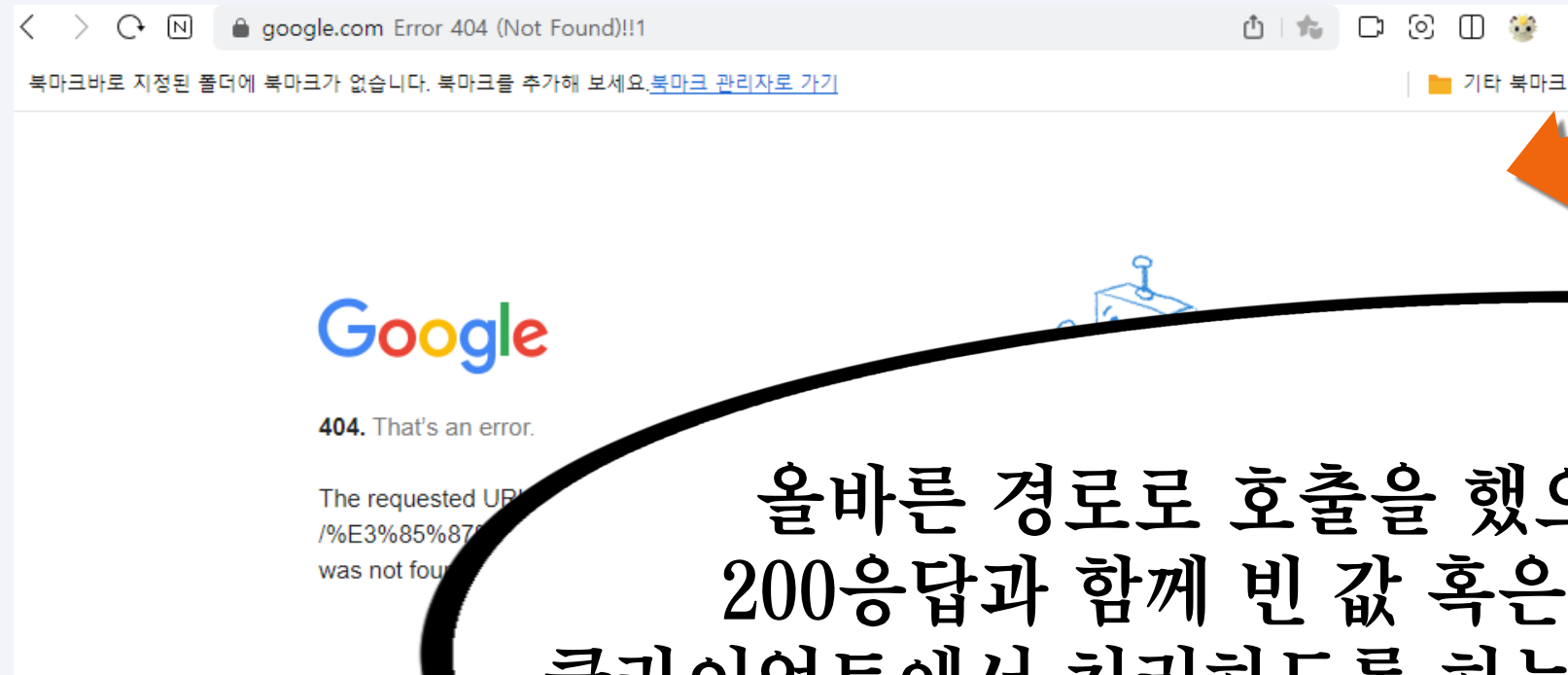
- 경로 없음과 자원 없음의 의미를 모두 포함
- 브라우저: 잘못된 경로로 접근했다
- 서버: 리소스를 찾을 수 없다
- 같은 404여도 해석의 차이가 발생할 수 있음

- 404는 리소스가 없어졌다는 것을 의미하지 않음
- 리소스가 영구적으로 삭제되었다면 404 상태 코드 대신 410(Gone) 상태 코드가 쓰여야 한다

# 생각해 볼 점: 데이터가 없는 경우의 상태코드

## 404 Not Found

클라이언트가 요청한 자원이 존재하지 않는 경우



- 경로 없음과 자원 없음의 의미를 모두 포함
- 브라우저: 잘못된 경로로 접근했다
- 서버: 리소스를 찾을 수 없다
- 같은 404여도 해석의 차이가 발생할 수 있음

올바른 경로로 호출을 했으나 데이터가 없을 때는  
200응답과 함께 빈 값 혹은 약속된 코드를 반환하여  
클라이언트에서 처리하도록 하는 것이 맞는 방향이라 생각하고  
앞으로 통일해 나갈 생각입니다.

- 404는 리소스가 없어
- 리소스가 영구적으로 삭제되었다는  
410(Gone) 상태 코드가 쓰여야 한다

요기요

# 느낀점

올바른 경로로 호출을 했으나 데이터가 없을 때는  
200응답과 함께 빈 값 혹은 약속된 코드를 반환하여  
클라이언트에서 처리하도록 하는 것이 맞는 방향이라 생각하고  
앞으로 통일해 나갈 생각입니다.

요기요

결국 같이 프로젝트를 하는 팀원들과 함께 상의 후  
룰을 정하고 통일시키는 것이 중요하다고 생각이 들었습니다  
계속 의견을 나누고 생각해보면서 최선의 방향을 찾아가는 것이 좋을 것 같습니다!

**충분한 논의를 통해**

**좋은 상태 코드를 반환해보아요!**

# 감사합니다



## 참고

- <https://sanghaklee.tistory.com/61>
- <https://hongong.hanbit.co.kr/http-%EC%83%81%ED%83%9C-%EC%BD%94%EB%93%9C-%ED%91%9C-1xx-5xx-%EC%A0%84%EC%B2%B4-%EC%9A%94%EC%95%BD-%EC%A0%95%EB%A6%AC/>
- <https://techblog.yogiyo.co.kr/%EB%8D%B0%EC%9D%B4%ED%84%B0%EA%B0%80-%EC%97%86%EC%9D%84-%EB%95%8C-200%EC%9D%B8%EA%B0%80-404%EC%9D%B8%EA%B0%80-f1c8c39ca9df>
- <https://guti-coding.tistory.com/134>