



**Zod is a runtime library for validating values and shapes in TypeScript.**

**Source Of Types**

<https://tkdodo.eu/blog/type-safe-react-query>

타입을 "갖는 것"과 타입이 "안전한 것"과는 차이가 있다

**There is a big difference between "having types" and "being type-safe"**

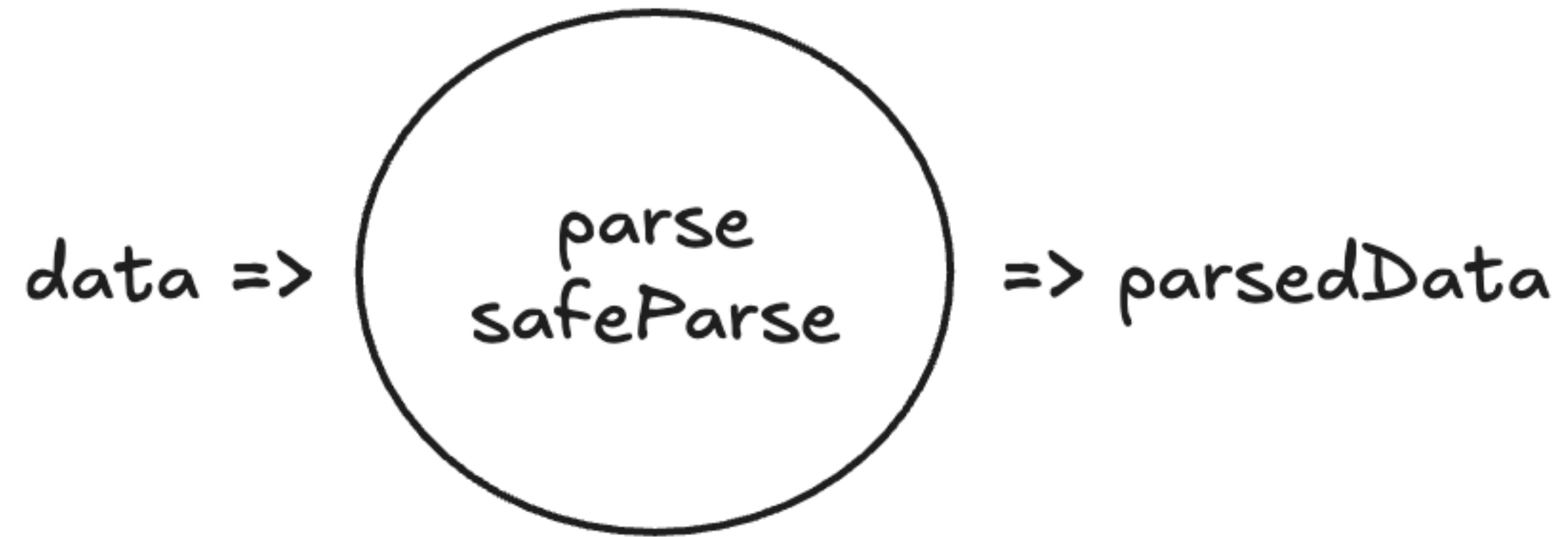


# Concepts

- validation: 데이터가 valid 한가?
- parsing: 데이터가 valid한지 체크하고 파싱된 데이터를 반환



# Concepts





# Concepts

```
▶ Uncaught ZodError: [  
  {  
    "code": "invalid_type",  
    "expected": "string",  
    "received": "number",  
    "path": [  
      "firstName"  
    ],  
    "message": "Expected string, received number"
```

우리는 우리가 선언한 타입을 믿는다.

하지만 우리의 애플리케이션 외부(API, URL, FORM, STORAGE)에서 오는 값들을 모두 신뢰할 수는 없다.

우리가 선언한 타입이 외부에서 들어올 것이라고 바라는 것 뿐이다.

# type assertion



```
const fetchTodo = async (id: number) => {  
  const response = await axios.get<Todo>(`/todos/${id}`)  
  return response.data  
}
```



```
function get<T = any>(url: string): Promise<{ data: T, status: number}>
```



# The golden rule of Generics

I learned this rule from [@danvdk's](#) great book *Effective TypeScript*. It basically states:

**For a Generic to be useful, it must appear at least twice.**

The so called "return-only" generics are nothing more than a type assertion in disguise. The (slightly simplified) type signature for `axios.get` reads:

axios-get-type-signature

TS

Copy

```
1 function get<T = any>(url: string): Promise<{ data: T, status: number}>
```

The Type `T` only appears in one place - the return type. So it's a lie! We could've just as well written:

explicit-type-assertion

TS

Copy

```
1 const fetchTodo = async (id: number) => {  
2   const response = await axios.get(`/todos/${id}`)  
3   return response.data as Todo  
4 }
```



```
type Todo = { id: number; name: string; done: boolean }
```

```
// ✅ 직접 리턴 타입 명시
```

```
const fetchTodo = async (id: number): Promise<Todo> => {  
  const response = await axios.get(`/todos/${id}`)  
  return response.data  
}
```

```
// ✅ 이러면 데이터에 대한 타입 추론
```

```
const query = useQuery({  
  queryKey: ['todos', id],  
  queryFn: () => fetchTodo(id),  
})
```

```
query.data
```

```
//    ^?(property) data: Todo | undefined
```

백엔드의 응답값을 믿지만, 항상 완벽하게 합의한대로 응답이 오지는 않을 수 있다.

Zod의 목표는 어떤 값이 내가 원한대로 생겼는지를 보장하는 것이다.(in runtime)

- The goal of zod is to guarantee that your expectations of how this thing is shaped are true.
- Typescript is not a guarantee. typescript doesn't run. Any type enforcement you do in typescript gets lost after it's been compiled.

## Usecase

- FE, BE anywhere
- Parsing requests
- Validating forms (w/ react-hook-form. Zod-resolver)
- Validating external apis (hopefully you don't need this)
- Parsing URL (Query Params)

# 백엔드의 응답값을 믿지만, 항상 완벽하게 합의한대로 응답이 오지는 않을 수 있다.

## 프론트의 잘못된인가? 아니다. 하지만 유저는 상관하지 않는다. 그냥 기능이 안되는거니까

←→↺local.berriz.in:3000/ko/ive/artist/9

📁 git📁 jira📁 팬플랫폼🔵 마이워크📁 start📁 아지트🔵 kakao 시작페이지📁 내자리📁 EC5🔴 YouTube📁 dev📁 Berriz 모니터링📁 오디션 모니터링🟢 [베리즈샵] WBS🟢 shop api

모든 북마크

469 messages101 user messages83 errors332 warnings47 info<other> 17hot-re... 307 verbose

Hide networkPreserve logSelected context onlyGroup similar messages in consoleShow CORS errors in consoleLog XMLHttpRequestsEager evaluationAutocomplete from historyTreat code evaluation as user action

▶ The above error occurred in the <NotFoundErrorBoundary> component: [hook.js:608](#)

at ActivitySection (webpack-internal:///.../domains/community/artist/presentational/ActivitySection.client.tsx:17:11)  
at ArtistActivitySectionContainer (webpack-internal:///.../domains/community/artist/container/ArtistActivitySectionContainer.client.tsx:17:11)  
at ArtistProfileContainer (Server)  
at main  
at CommonLayout (Server)  
at div  
at div  
at FlexIsolateWrapper (Server)  
at CommunityLayoutContainer (Server)  
at ArtistProfilePage (Server)  
at InnerLayoutRouter (webpack-internal:///.../node\_modules/.pnpm/next@14.2.20\_@babel+core@7.26.9\_@opentelemetry+api@1.9.0\_react-dom@18.3.1\_react@18.3.1/node\_modules/next/dist/client/components/layout-router.js:243:11)  
at RedirectErrorBoundary (webpack-internal:///.../node\_modules/.pnpm/next@14.2.20\_@babel+core@7.26.9\_@opentelemetry+api@1.9.0\_react-dom@18.3.1\_react@18.3.1/node\_modules/next/dist/client/components/redirect-boundary.js:74:9)  
at RedirectBoundary (webpack-internal:///.../node\_modules/.pnpm/next@14.2.20\_@babel+core@7.26.9\_@opentelemetry+api@1.9.0\_react-dom@18.3.1\_react@18.3.1/node\_modules/next/dist/client/components/redirect-boundary.js:82:11)  
at NotFoundBoundary (webpack-internal:///.../node\_modules/.pnpm/next@14.2.20\_@babel+core@7.26.9\_@opentelemetry+api@1.9.0\_react-dom@18.3.1\_react@18.3.1/node\_modules/next/dist/client/components/not-found-boundary.js:84:11)  
at LoadingBoundary (webpack-internal:///.../node\_modules/.pnpm/next@14.2.20\_@babel+core@7.26.9\_@opentelemetry+api@1.9.0\_react-dom@18.3.1\_react@18.3.1/node\_modules/next/dist/client/components/layout-router.js:349:11)  
at ErrorBoundary (webpack-internal:///.../node\_modules/.pnpm/next@14.2.20\_@babel+core@7.26.9\_@opentelemetry+api@1.9.0\_react-dom@18.3.1\_react@18.3.1/node\_modules/next/dist/client/components/error-boundary.js:160:11)  
at InnerScrollAndFocusHandler (webpack-internal:///.../node\_modules/.pnpm/next@14.2.20\_@babel+core@7.26.9\_@opentelemetry+api@1.9.0\_react-dom@18.3.1\_react@18.3.1/node\_modules/next/dist/client/components/layout-router.js:153:9)  
at ScrollAndFocusHandler (webpack-internal:///.../node\_modules/.pnpm/next@14.2.20\_@babel+core@7.26.9\_@opentelemetry+api@1.9.0\_react-dom@18.3.1\_react@18.3.1/node\_modules/next/dist/client/components/layout-router.js:228:11)

!

서비스 오류가 발생했습니다

페이지 새로고침 후 이용해주세요

새로고침



# API 반환값에 type 검증을 추가한다면? (필요한 곳에)

throwOnError 옵션에 의해 에러바운더리로 던져질 것

parsing-with-zod

TS

Copy

```
1 import { z } from 'zod'
2
3 // 👁 define the schema
4 const todoSchema = z.object({
5   id: z.number(),
6   name: z.string(),
7   done: z.boolean(),
8 })
9
10 const fetchTodo = async (id: number) => {
11   const response = await axios.get(`/todos/${id}`)
12   // 🚀 parse against the schema
13   return todoSchema.parse(response.data)
14 }
15
16 const query = useQuery({
17   queryKey: ['todos', id],
18   queryFn: () => fetchTodo(id),
19 })
```