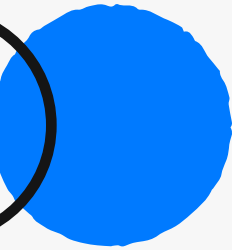


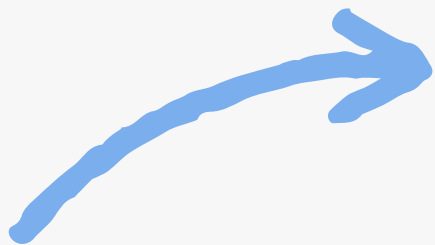


## Effective TypeScript

Item 47.



공개 API에 등장하는 모든 타입을 익스포트 하기



@Bori-github

# 목차

01

익스포트 되지 않은 타입

02

타입 숨기기

03

숨겨진 타입 추출하기

04

마무리

O'REILLY®

## Effective TypeScript

62 Specific Ways to Improve Your TypeScript



Dan Vanderkam

## 익스포트 되지 않은 타입

타입스크립트에서 서드 파티 라이브러리를 사용하려면?

- 각 기능에 대한 타입 정의 필요
- 라이브러리의 **익스포트**<sup>export</sup> **되지 않은 타입 정보**가 필요한 경우 발생  
⇒ 타입 매핑 도구를 사용하거나 필요한 타입을 참조하는 방법을 사용



**라이브러리 제작자 관점**에서 프로젝트 초기에 **타입 익스포트**부터 **작성**해야 함을 의미

☑ 특정 타입을 숨기기 위해 익스포트하지 않은 경우



```
1 interface SecretName {  
2   first: string;  
3   last: string;  
4 }  
5  
6 interface SecretSanta {  
7   name: SecretName;  
8   gift: string;  
9 }  
10  
11 export function getGift(name: SecretName, gift: string): SecretSanta {  
12   // ...  
13 }
```

☑ импорт 가능

- getGift 함수

🚫 импорт 불가능

- SecretName 타입

- SecretSanta 타입

## 숨겨진 타입 추출하기

✅ Parameters/ReturnType 제네릭 타입을 사용하여 추출하기

```
1 export function getGift(name: SecretName, gift: string): SecretSanta {  
2     // ...  
3 }  
4  
5 type MySanta = ReturnType<typeof getGift>; // SecretSanta  
6 type MyName = Parameters<typeof getGift>[0]; // SecretName
```

각 타입은 익스포트된 함수 시그니처에 등장

⇒ Parameters/ReturnType 제네릭 타입 사용 가능

## 숨겨진 타입 추출하기



[ 사용자 ]



안전하게 타입 추출 가능



[ 제작자 ]



숨기려고 했던 타입이 노출

⇒ 라이브러리 사용자를 위해 모든 타입을 명시적으로 익스포트하기

- 공개 라이브러리에 등장한 모든 타입은 **익스포트** 하는 것이 좋습니다.
- 라이브러리 사용자가 타입을 추출할 수 있으므로, **익스포트하기 쉽게 만드는 것**이 좋습니다.



# Thanks!

**Do you have any questions?**

qhflrnfl4324@gamil.com

<https://github.com/Bori-github>

