

Item 35.

데이터가 아닌,
API와 명세를 보고 타입 만들기

Effective TypeScript

목 차

1. 명세를 참고해 타입을 생성하기
2. API 명세로부터 타입을 생성하기
3. 데이터로부터 타입을 생성하기
4. 마무리

O'REILLY®

Effective TypeScript

62 Specific Ways to Improve Your TypeScript



Dan Vanderkam

1. 명세를 참고해 타입을 생성하기

프로젝트 외부에서 비롯된 타입은 **명세를 참고해 타입을 생성**

✓ 실수 방지

명세를 참고해 타입을 생성하여 사용자의 실수를 줄임

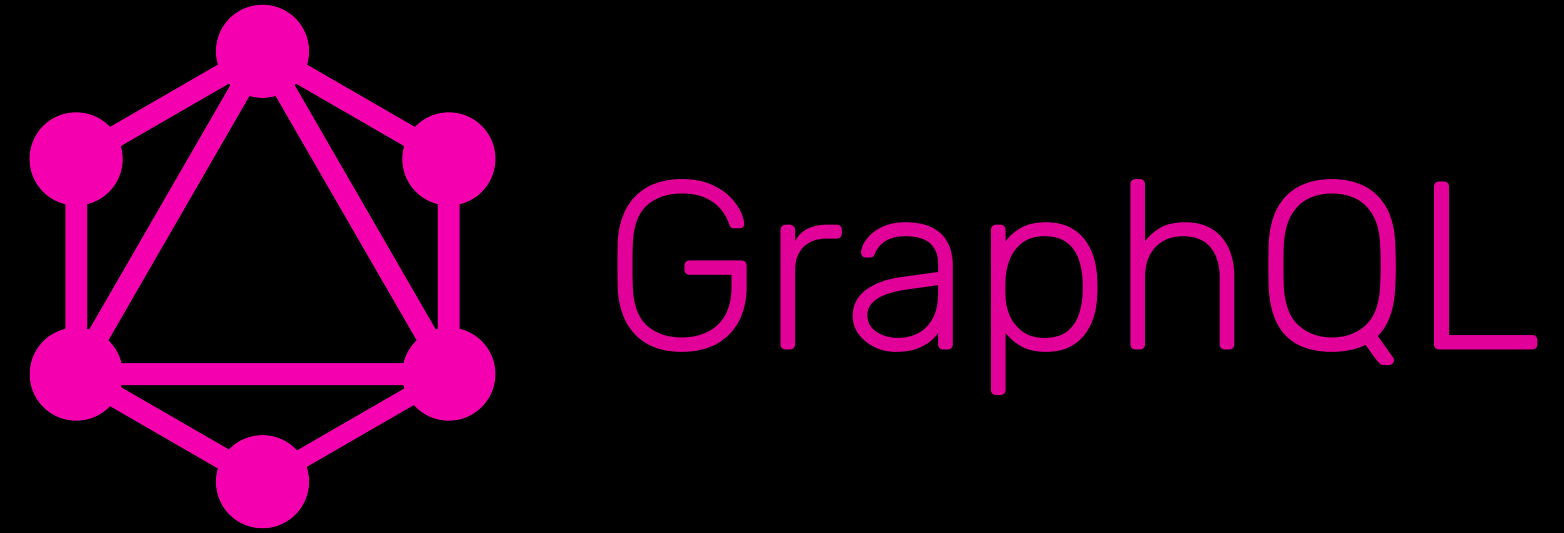
✓ 오류 방지

예시 데이터를 참고하여 타입을 생성하면 보이는 데이터만 고려하게 되므로 예기치 않은 오류 발생

⇒ 명세를 기반으로 타입을 작성하면
사용 가능한 모든 값에 대해 작동한다는 **확신**을 가질 수 있다.

2. API 명세로부터 타입을 생성하기

API의 타입은 어떻게 생성하는 것이 좋을까요?



- API를 위한 쿼리 언어
- 타입스크립트와 비슷한 **타입 시스템**을 사용
- 클라이언트는 GraphQL 쿼리를 통해 필요한 데이터를 서버에 요청,
서버는 해당 쿼리를 해석하여 데이터를 반환

2. API 명세로부터 타입을 생성하기

GraphQL 스키마

- API에서 사용되는 **데이터 모델의 구조와 규칙**을 정의

⇒ 클라이언트와 서버 간의 데이터 요청과 응답을 정확하게 매핑

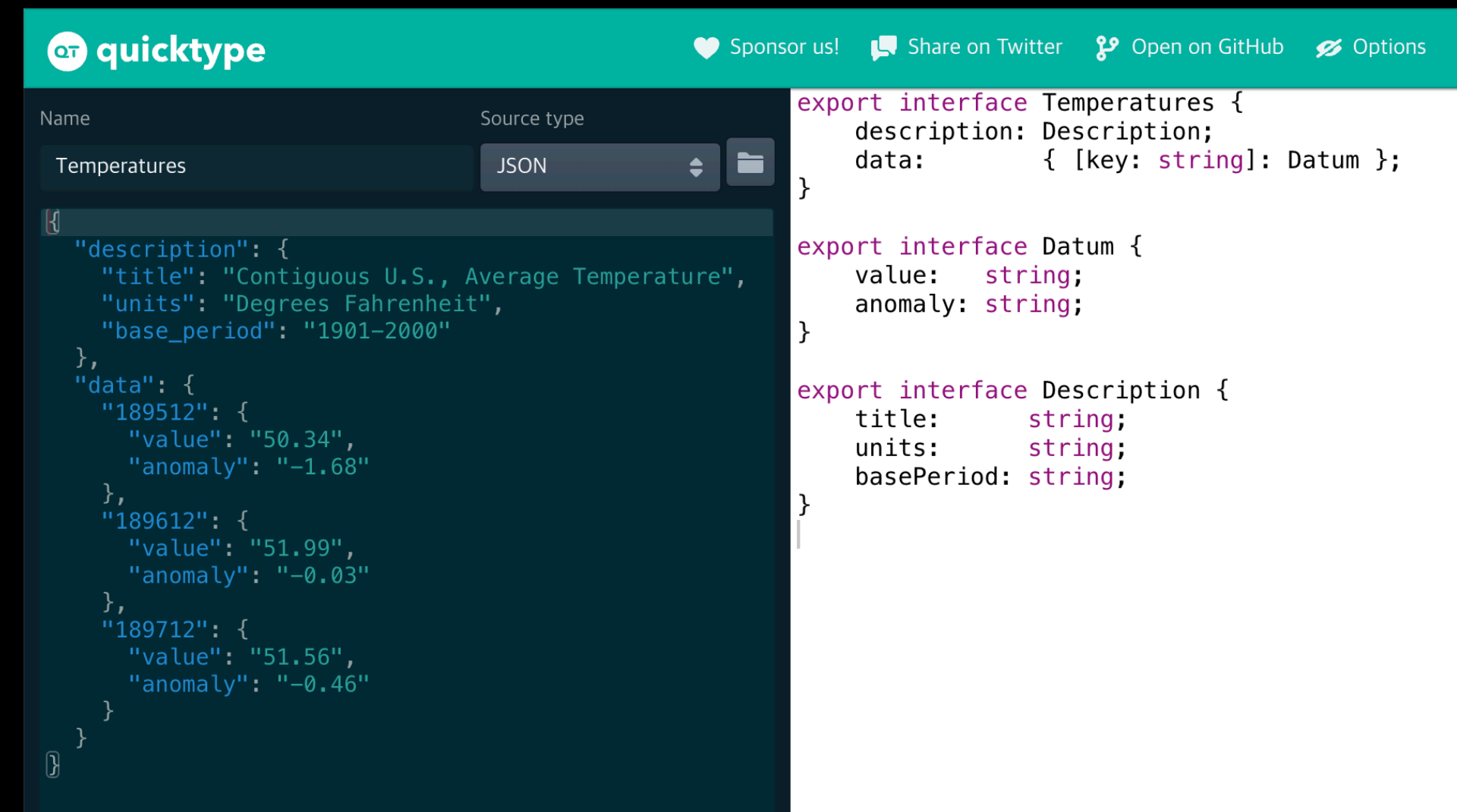
```
// GraphQL scheme
type Project {
  name: String
  tagline: String
  contributors: [User]
}
```

- Apollo와 같은 GraphQL 쿼리를 타입스크립트 타입으로 변환해주는 도구를 이용

⇒ 스키마로부터 특정 쿼리의 **타입스크립트 타입을 생성** 및 **API를 문서화**

3. 데이터로부터 타입을 생성하기

만약 명세 정보나 공식 스키마가 없다면? 데이터로부터 타입을 생성



- JSON 데이터로부터 자동으로 타입 선언을 생성해주는 도구
 - 개발자가 수동으로 타입을 작성하는 번거로움을 줄여줌
- ⇒ 하지만 생성된 타입이 실제 데이터와 일치하지 않을 수 있다.

4. 마무리

- 코드의 타입 안정성을 얻기 위해 **API 또는 데이터 형식에 대한 타입 생성**을 고려해야 합니다.
- 데이터에 드러나지 않는 예외적인 경우들이 문제가 될 수 있기 때문에 데이터보다는 **명세로부터 타입을 생성**하는 것이 좋습니다.

감사합니다 :))

Effective TypeScript

@Bori-github(이보리)