

item 46

타입 선언과 관련된 세 가지 버전 이해하기

채희수 / 23.07.23

1. 라이브러리의 버전
2. 타입 선언(@types)의 버전
3. 타입스크립트의 버전

```
"dependencies": {
  "@emotion/react": "^11.11.0",
  "@emotion/styled": "^11.11.0",
  "@toss/utils": "^1.4.2",
  "axios": "^1.4.0",
  "classnames": "^2.3.2",
  "crypto-js": "^4.1.1",
  "emotion-normalize": "^11.0.1",
  "react": "^18.2.0",
  "react-dom": "^18.2.0",
  "uuid": "^9.0.0",
  "zod": "^3.21.4"
},
"devDependencies": {
  "@babel/plugin-proposal-nullish-coalescing-operator": "^7.18.6",
  "@babel/plugin-proposal-optional-chaining": "^7.21.0",
  "@babel/preset-env": "^7.21.5",
  "@babel/preset-react": "^7.18.6",
  "@babel/preset-typescript": "^7.21.5",
  "@emotion/babel-plugin": "^11.11.0",
  "@emotion/eslint-plugin": "^11.11.0",
  "@testing-library/dom": "^9.3.0",
  "@testing-library/jest-dom": "^5.16.5",
  "@testing-library/react": "^13.4.0",
  "@testing-library/user-event": "^13.5.0",
  "@types/crypto-js": "^4.1.1",
  "@types/jest": "^29.5.1",
  "@types/react": "^18.2.6",
  "@types/react-dom": "^18.2.4",
  "@types/testing-library__jest-dom": "^5.14.5",
  "@types/uuid": "^9.0.1",
  "@typescript-eslint/eslint-plugin": "^5.59.6",
  "@typescript-eslint/parser": "^5.59.6",
  "customize-cra": "^1.0.0",
  "eslint": "^8.41.0",
  "eslint-config-prettier": "^8.8.0",
  "eslint-plugin-import": "^2.27.5",
  "eslint-plugin-prettier": "^4.2.1",
  "eslint-plugin-react": "^7.32.2",
  "eslint-plugin-react-hooks": "^4.6.0",
  "eslint-plugin-simple-import-sort": "^10.0.0",
  "identity-obj-proxy": "^3.0.0",
  "jest": "^29.5.0",
  "jest-environment-jsdom": "^29.5.0",
  "msw": "1.2.1",
  "prettier": "^2.8.8",
  "react-app-rewired": "^2.2.1",
  "react-scripts": "^5.0.1",
  "typescript": "^5.0.4"
},
```

**“ 세 가지 버전을 맞춰야 하기 때문에,
오히려 타입스크립트를 사용하면 의존성 관리가 어렵다. ”**

Minor = 기능 추가

1. 2. 3

Major
= 기능 대거 수정

Patch
= 에러 수정

* 사용하는 방법이 달라졌을 수 있기 때문에 무조건 업데이트 하면 안됨

Q. 라이브러리 버전과 타입 선언의 버전이 다릅니다. 괜찮나요?

A. 네 괜찮습니다.

타입 선언의 패치 버전이 오히려 더 높는데,
이는 버그나 누락으로 인한 수정과 추가에 따른 것입니다.
라이브러리 자체보다 타입 선언에 더 많은 업데이트가 있었음을 의미합니다.

```
"react": "^18.2.0",
```

```
"@types/react": "^18.2.6",
```

그러나 실제 라이브러리와 타입 정보의 버전이 별도로 관리되는 방식은 4가지 문제점이 있습니다.

첫 번째, 라이브러리를 업데이트했지만 타입 선언은 업데이트 하지 않는 경우

가능성) 업데이트 된 새로운 기능을 사용할 때 타입 오류 발생

해결책) 타입 선언도 업데이트해 라이브러리와 버전을 맞춰준다.

그러나 만약 업데이트해야 할 타입 선언 버전이 없는 경우 ? 업데이트를 직접 작성한다.

두 번째, 라이브러리보다 타입 선언의 버전이 최신인 경우

= 라이브러리를 버전 정보없이 사용하다가 뒤늦게 타입 선언을 설치하는 경우, 라이브러리와 타입 선언의 새 버전이 릴리즈되었다면 새 버전이 다운로드 된다.

해결책) 타입 선언도 업데이트해 라이브러리와 버전을 맞춰준다.

라이브러리 버전을 올리거나 타입 선언의 버전을 내린다.

그러나 실제 라이브러리와 타입 정보의 버전이 별도로 관리되는 방식은 4가지 문제점이 있습니다.

세 번째, 프로젝트에서 사용하는 타입스크립트 버전보다

라이브러리에서 필요로 하는 타입스크립트 버전이 최신인 경우

= 유명 자바스크립트 라이브러리는 타입 정보를 더 정확하게 표현하기 위해 타입 시스템을 개선하고 버전을 올린다.

가능성) @types 선언 자체에서 타입 오류가 발생하게 된다.

해결책) 프로젝트의 타입스크립트 버전을 올리거나,

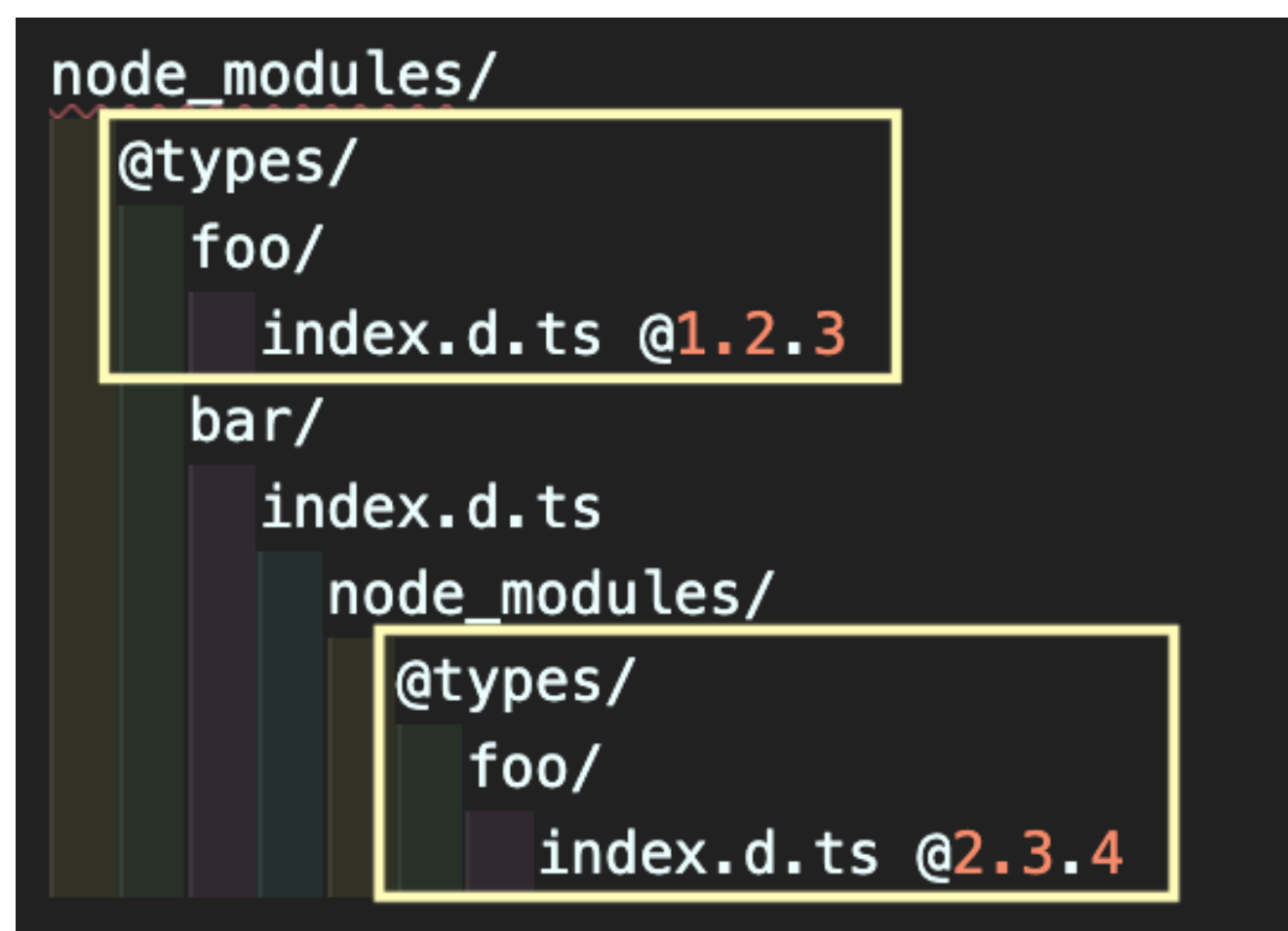
라이브러리 타입 선언의 버전을 원래대로 내리거나,

declare module 선언으로 라이브러리 타입 정보를 없애 버린다.

그리고 라이브러리에 따라 버전별 타입 선언을 제공할 수도 있으니 확인해보자.

그러나 실제 라이브러리와 타입 정보의 버전이 별도로 관리되는 방식은 4가지 문제점이 있습니다.

네 번째, @types 의존성이 중복되는 경우



= @types/bar가 현재 프로젝트와 호환되지 않는 버전의 @types/foo에 의존한다면 npm은 중첩된 폴더에 별도로 해당 버전을 설치해 문제를 해결하려 한다.

가능성) 전역 네임스페이스에 타입 선언이 존재하면 중복된 선언, 또는 선언이 병합될 수 없다는 오류가 난다.

해결책) 'npm ls @types/foo'를 실행해 어디서 타입 선언 중복이 발생했는지 추적
@types/foo를 업데이트하거나
@types/bar를 업데이트해서 서로 버전이 호환되도록 한다.

- * @types이 전이 의존성을 가지도록 만드는 것은 종종 문제를 일으킨다.
- * 만약 타입 선언을 작성하고 공개하려고 한다면, item51 참고


```
"types": "./types/runtime/index.d.ts",
```

일부 라이브러리, 특히 타입스크립트로 작성된 라이브러리들은
자체적으로 타입 선언을 포함하게 된다.

이는 package.json 의 'types' 필드에서 확인할 수 있다.

라이브러리 자체적으로 타입 선언을 포함하는 번들링 방식은 4가지 문제점이 있습니다.

**첫 번째, 번들된 타입 선언에 보강 기법으로 해결할 수 없는 오류가 있는 경우
또는 공개 시점에는 잘 동작했지만 타입스크립트 버전이 올라가면서
오류가 발생하는 경우 문제가 된다.**

가능성) 단 하나의 잘못된 타입 선언으로 인해 타입스크립트의 버전을 올리지 못할 수 있다.

두 번째, 프로젝트 내의 타입선언이 다른 라이브러리 타입 선언에 의존하면서 문제가 된다.

**세 번째, 프로젝트의 과거 버전에 타입 선언에 문제가 있는 경우에는
과거 버전으로 돌아가서 패치 업데이트를 해야한다.**

네 번째, 타입 선언의 패치 업데이트를 자주 하기 어렵다는 문제가 있다.

요약

- **@types 의존성과 관련된 세 가지 버전이 있다.**
라이브러리 버전, @types 버전, 타입스크립트 버전.
- 라이브러리를 업데이트하는 경우, 해당 @types 역시 업데이트 해야 한다.
- 타입 선언을 라이브러리에 포함하는 것과 DefinitelyTyped에 공개하는 것 사이의 장단점을 이해해야 한다.
타입스크립트로 작성된 라이브러리라면 타입 선언을 자체적으로 포함하고,
자바스크립트로 작성된 라이브러리라면 타입 선언을 DefinitelyTyped에 공개하는 것이 좋다.