types 和 @types 是什么？

2020-08-21

TypeScript 的学习资料非常多，其中也不乏很多优秀的文章和教程。但是目前为止没有一个我特别满意的。原因有：

* 它们大多数没有一个清晰的主线，而是按照 API 组织章节的，内容在***逻辑上***比较零散。
* 大多是“讲是什么，怎么用“，而不是”讲为什么，讲原理“。
* 大多数内容比较枯燥，趣味性比较低。都是干巴巴的文字，没有图片，缺乏能够引起强烈共鸣的例子。

因此我的想法是做一套不同市面上大多数的 TypeScript 学习教程。以人类认知的角度思考问题，学习 TypeScript，通过通俗易懂的例子和图片来帮助大家建立 TypeScript 世界观。

系列安排：

* [上帝视角看 TypeScript（已发布）](https://lucifer.ren/blog/2020/08/04/ts-internal/)
* [TypeScript 类型系统（已发布）](https://lucifer.ren/blog/2020/08/15/ts-type-system/)
* types 和 @types 是什么？（就是本文）
* [你不知道的 TypeScript 泛型（万字长文，建议收藏）](https://lucifer.ren/blog/2020/06/16/ts-generics/)（已发布）
* TypeScript 配置文件该怎么写？
* TypeScript 是如何与 React，Vue，Webpack 集成的？
* TypeScript 练习题

目录将来可能会有所调整。

注意，我的系列文章基本不会讲 API，因此需要你有一定的 TypeScript 使用基础，推荐两个学习资料。

* [深入理解 TypeScript](https://jkchao.github.io/typescript-book-chinese/)
* [官方文档](https://www.typescriptlang.org/docs/home)

结合这两个资料和我的系列教程，掌握 TypeScript 指日可待。

接下来，我们通过几个方面来从宏观的角度来看一下 TypeScript。

## 前言

* 作者：feiker & Lucifer

TypeScript 中有几个概念和名字很像，会让初学者傻傻分不清楚。比如配置文件中的 ***types 和 typeRoots***，并且还有一个 @types。接触过 TypeScript 的人一定接触过它们， 这几个有什么区别和联系呢？今天就带你来重新认识下它们。

## 一个例子

这里我通过一个例子来说明一下什么是 @types，这样大家理解起来更深刻一点。

当我们用 npm 等包管理工具安装第三方包的时候，有些包并不是 TypeScript 编写的，自然也不会导出 TypeScript 声明文件。这种情况下，如果我们在 TypeScript 项目中引入了这种包，则会编译报错(没有设置 allowJS)。举个例子，当我们通过npm install jquery --save 安装 jquery 包并引用的时候，TypeScript 会报错。

allowJS 是 TypeScript 1.8 引进的一个编译项。

报错内容如下：

Could not find a declaration file for module ‘jquery’. Try npm install @types/jquery if it exists or add a new declaration (.d.ts) file containing declare module 'jquery';

这里的意思是 TypeScript 没有找到 jquery 这个包的定义，你可以通过npm install @types/jquery安装相关声明，或者自己定义一份.d.ts 文件，并将 jquery 声明为 module。

全世界不是 TypeScript 编写的包多了去了。即使你的包是 TypeScript 编写的，如果你没有导出声明文件，也是没用的。（TypeScript 默认不会导出声明文件，只会编译输出 JavaScript 文件）。因此 TypeScript 必须对这种情况提供解决方案，而上面的两种方案（安装 @types 和 自己 declare module）就是 TypeScript 官方提出的， 你可以选择适合你的方案。我的推荐是尽量使用 @types 下的声明，实在没有，再使用第二种方法。

值得一提的是，并不是所有的包都可以通过这种方式解决的， 能解决的是 DefinitelyTyped 组织已经写好定义的包， 好消息是比较流行的包基本都有。 如果你想查一个包是否在 @type 下，可以访问 https://microsoft.github.io/TypeSearch/

那么 TypeScript 是怎么找定义的，什么情况会找不到定义而报类似上面举的例子的错误，这里简单介绍下原理。

## 包类型定义的查找

就好像 node 的包查找是先在当前文件夹找 node\_modules，在它下找递归找，如果找不到则往上层目录继续找，直到顶部一样， TypeScript 类型查找也是类似的方式。

具体来说就是：

* TypeScript 编译器先在当前编译上下文找 jquery 的定义。
* 如果找不到，则会去 node\_modules 中的@types （默认情况，目录可以修改，后面会提到）目录下去寻找对应包名的模块声明文件。

@types/\*模块声明文件由社区维护，通过发布到@types 空间下。 [GitHub - DefinitelyTyped/DefinitelyTyped: The repository for high quality TypeScript type definitions.](https://github.com/DefinitelyTyped/DefinitelyTyped)

## 变量类型定义的查找

和包查找类似，默认情况下变量类型定义的查找也会去 @types 下去寻找。只不过并不是直接去 @types 找，而是有一定的优先级， 这个过程类似原型链或者作用域链。

比如如下代码：

const user: User = { name: "lucifer" };

* Typescript 则会先在本模块查找 User 的定义。
* 如果找到，则直接返回。 如果找不到， 则会到全局作用域找，而这个全局默认就是指的就是 @types 下的所有类型定义。（注意目录页是可以配的）

也就是说 @types 下的定义都是全局的。当然你可以导入 @types 下导出的定义，使得它们的作用域变成你的模块内部。

## typeRoots 与 types

前面说了 TypeScript 会默认引入node\_modules下的所有@types声明，但是开发者也可以通过修改tsconfig.json的配置来修改默认的行为.

tsconfig.json 中有两个配置和类型引入有关。

1. typeRoots: 用来指定默认的类型声明文件查找路径，默认为node\_modules/@types, 指定typeRoots后，TypeScript 编译器会从指定的路径去引入声明文件，而不是node\_modules/@types, 比如以下配置会从typings路径下去搜索声明

{  
 "compilerOptions": {  
 "typeRoots": ["./typings"]  
 }  
}

1. types: TypeScript 编译器会默认引入typeRoot下所有的声明文件，但是有时候我们并***不希望全局引入所有定义***，而是仅引入部分模块。这种情景下可以通过types指定模块名只引入我们想要的模块，比如以下只会引入 jquery 的声明文件

{  
 "compilerOptions": {  
 "types": ["jquery"]  
 }  
}

## 总结

1. typeRoots 是 tsconfig 中 compilerOptions 的一个配置项，typeRoots 下面的包会被 ts 编译器自动包含进来，typeRoots 默认指向 node\_modules/@types。
2. @types 是 npm 的 scope 命名空间，和@babel 类似，@types 下的所有包会默认被引入，你可以通过修改 compilerOptions 来修改默认策略。
3. types 和 typeRoots 一样也是 compilerOptions 的配置，指定 types 后，typeRoots 下只有被指定的包才会被引入。

## 参考

* [GitHub - DefinitelyTyped/DefinitelyTyped: The repository for high quality TypeScript type definitions.](https://github.com/DefinitelyTyped/DefinitelyTyped)
* [@types | 深入理解 TypeScript](https://jkchao.github.io/typescript-book-chinese/typings/types.html)
* [tsconfig.json · TypeScript 中文网 · TypeScript——JavaScript 的超集](https://www.tslang.cn/docs/handbook/tsconfig-json.html)
* [理解 Typescript 配置文件 - 个人文章 - SegmentFault 思否](https://segmentfault.com/a/1190000013514680)

## 关注我

大家也可以关注我的公众号《脑洞前端》获取更多更新鲜的前端硬核文章，带你认识你不知道的前端。



公众号【 [力扣加加](https://tva1.sinaimg.cn/large/007S8ZIlly1gfcuzagjalj30p00dwabs.jpg)】 知乎专栏【 [Lucifer - 知乎](https://www.zhihu.com/people/lu-xiao-13-70)】

点关注，不迷路！