

Why TypeScript

使用TypeScript高质量高效率的JavaScript前后端编程

刘冰心 2017.02 Bingxin.liu@zhaopin.com.cn

- . TypeScript 是个啥?
- . TypeScript 特性
- . 引入 TypeScript 的成本
- . DEMO
- . 总结

TypeScript 是介啥?

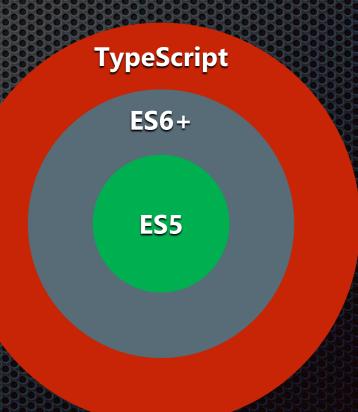
- · 产生的背景,即JavaScript中遇到的问题
 - · JavaScript 语言设计的缺失
 - · 大型JavaScript是困难的
 - 项目逐渐长大->难以维护(不易重构)->更多代码更多功能-> 难以增加新功能->重写
 - The Feature Gap

TypeScript 概述

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript.

Any browser. Any host. Any OS.

Open source.



谁在用TypeScript?

- VS Code
- Angular 2
- · Egret 白鹭引擎
- Ant Design
- . Ionic
- 华为、腾讯

```
縫 https://monacotest1.scm.ant-intapp.windo... 🔎 🔻 🔒 💍
                                                            todo.model.js - wwwroot - ...
Visual Studio Online "Monaco" | monacotest1 ▼ | ♦ master* ▼
                              todo.model.js /MvcApplication3/Scripts/app
<u>a</u>
                               1 (function (ko, datacontext) {
        ▶ .git
                                       datacontext.todoItem = todoItem;

■ MvcApplicatio...

                                3
                                       datacontext.todoList = todoList;

■ Content
                               4
                                5
                                       function todoItem(data) {
            themes
                                           var self = this;
                                6
             Site.css
                               7
                                           data = data || {};
             TodoList.css
          Images
                                           // Persisted properties
                               9
oldsymbol{f (}
                                            self.id = data.id;
          Properties
                              10
                                           self.title = ko.observable(data.title)
                              11
          Scripts
                                           self.isDone = ko.observable(data.isDon
                              12
Ð
            ■ app
                                            self.todoListId = data.todoListId;
                              13
               todo.bin...
                              14
(D)
               todo.dat...
                              15
                                           // Non-persisted properties
                                           self.errorMessage = ko.observable();
                              16
               todo.mo...
                              17
@
               todo.vie...
                              18
                                           var saveChanges = function () {
             reference...
                                                return datacontext.saveChangedTodo
                              19
                               20
                                           };
             jquery-1.8....
                               21
             jquery-1.8....
                                           // Auto-save when these properties cha
                              22
             jquery-1.8....
                                           self.isDone.subscribe(saveChanges);
                               23
                                           self.title.subscribe(saveChanges);
             jquery-ui-...
                               24
                               25
             iquery-ui-...
                                           //self.toJson = function () { return |
                               26
             iauerv.uno...
```

- · Type 类型 已经证明能够提高代码质量和可理解性
 - · 编译期发现错误 运行时出现错误的成本太高
 - 类型系统实际上是最好的文档
 - 增强了编辑器和 IDE 的功能,包括代码补全、接口提示、 跳转到定义、重构等等

- . TypeScript 是 JavaScript 的超集 , .js 文件可以直接重命名 为 .ts 即可
- · 即使 TypeScript 编译报错,也可以生成 JavaScript 文件
- · 兼容第三方库,即使第三方库不是用 TypeScript 写的

One Input, Targets All ES6 ES6 TS TypeScript Compiler ES5 ES3 ES5

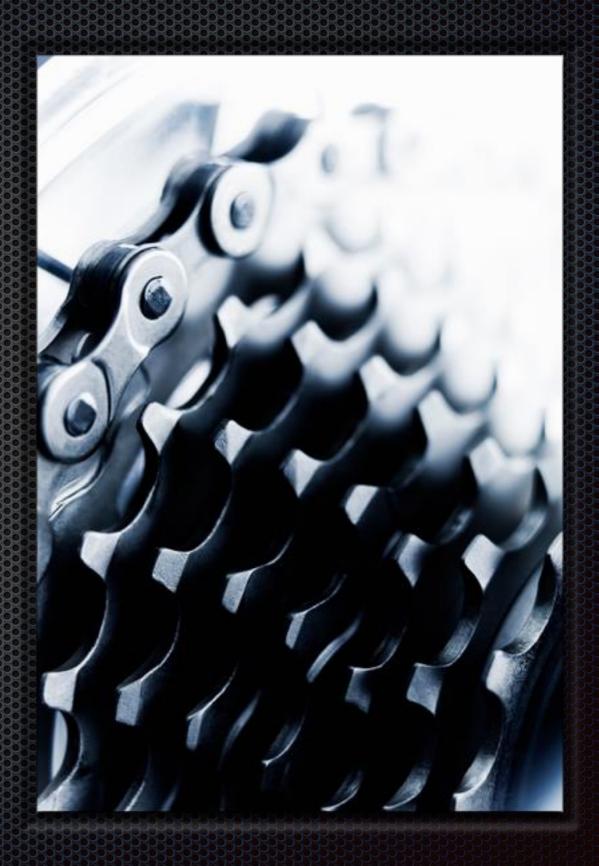
- . Class
- Interface
- Generic
- **.** Enum
- · Decorator 装饰器
 - · 类似于 C# 的attribute(特性), P Java的annotation(注解)

```
class Person {
    @log()
    say() {
       console.log('hello')
    }
}
function log() {
    // todo
}
```

引入TypeScript的成本

- · 开发效率问题
 - 不仅仅是编码效率,还有debug/fixbug效率、运维效率等等
- · 开发成本问题
 - . 学习成本 ≈ 0
 - 安装成本 = 0
 - 编码成本 = 0
 - ▶ 部署运维成本 = 0

Demo



总结





Next: Node.js 单元测试