

Strictness - Tìm hiểu về strict flag

AGENDA

1. strict flag là gì?
2. Option: noImplicitAny
3. Option: strictNullChecks
4. Option: alwaysStrict
5. Other options
6. Summary

```
26
27
28 /* Strict Type-Checking Options */
29 "strict": true,           /* Enable all strict type-checking options. */
30 // "noImplicitAny": true, /* Raise error on expressions and declarations with an implied 'any' type. */
31 // "strictNullChecks": true, /* Enable strict null checks. */
32 // "strictFunctionTypes": true, /* Enable strict checking of function types. */
33 // "strictBindCallApply": true, /* Enable strict 'bind', 'call', and 'apply' methods on functions. */
34 // "strictPropertyInitialization": true, /* Enable strict checking of property initialization in classes. */
35 // "noImplicitThis": true, /* Raise error on 'this' expressions with an implied 'any' type. */
36 // "alwaysStrict": true, /* Parse in strict mode and emit "use strict" for each source file. */
```

1. strict flag là gì?

- Bật `strict=true` sẽ bật tất cả options liên quan tới strict lên `true`.
- Mặc định các options liên quan tới strict là `false`, nhưng `recommend` sử dụng `true`
- Config của từng option sẽ ưu tiên cao hơn so với "strict". Tức `strict=true`, mà `noImplicitAny=false` thì giá trị cuối cùng vẫn là `noImplicitAny=false`.

```
// tsconfig.json
{
  "compilerOptions": {
    // ...
    "strict": true,

    /*
     All options below are FALSE by default
     BUT recommend to use TRUE 😊
    */
    // "noImplicitAny": true,
    // "strictNullChecks": true,
    // "strictFunctionTypes": true,
    // "strictBindCallApply": true,
    // "strictPropertyInitialization": true,
    // "noImplicitThis": true,
    // "alwaysStrict": true,

    // typescript v4.4
    // "useUnknownInCatchVariables": true,
  }
}
```

2. Option: noImplicitAny

- Khi typescript không thể xác định được kiểu dữ liệu của một biến nào đó, thì biến đó sẽ được mang kiểu dữ liệu **any**
- Khi bật lên **true**, nó sẽ báo lỗi nếu trường hợp này xảy ra, ngược lại bỏ qua lỗi này.

```
// tsconfig.json
{
  "compilerOptions": {
    // ...
    "noImplicitAny": true,
  }
}
```

```
function isEmpty(numberList) {
  return numberList.length === 0;
}

console.log(isEmpty(123));
// js: false
// ts: error Parameter 'numberList' implicitly has an 'any' type.
```

```
function isEmpty(numberList) {
  return numberList.length === 0;
}
```

Parameter 'numberList' implicitly has an 'any' type. ts(7006)

(parameter) numberList: any

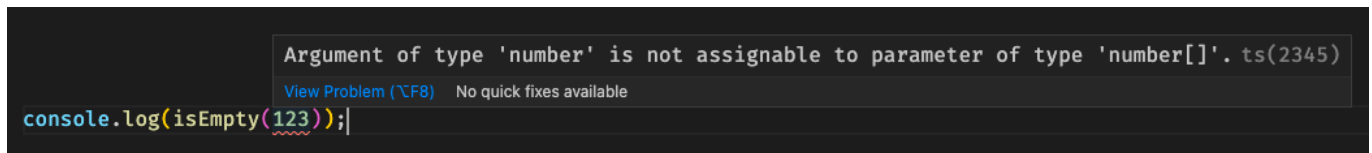
[View Problem](#) (^F8) [Quick Fix...](#) (⌘.)

FIX: khai báo kiểu dữ liệu vào nhé hehe

```
function isEmpty(numberList: number[]) {
  return numberList.length === 0;
}

console.log(isEmpty(123)); // wrong data type
// js: false
// ts: Argument of type 'number' is not assignable to parameter of type 'number[]'.ts(2345)

console.log(isEmpty([])); // it works and return true
```



3. Option: strictNullChecks

- Khi bật lên **true** thì sẽ báo lỗi khi **null/undefined** được sử dụng không đúng. Ví dụ như gọi những hàm của number, string, array, ... chẳng hạn.
- Khi config **false** thì sẽ bỏ qua, nhưng có rủi ro xảy ra lỗi lúc runtime.

```
// tsconfig.json
{
  "compilerOptions": {
    // ...
    "strictNullChecks": true,
  }
}
```

```
let str;

if (Math.random() > 0.5) {
  str = '';
}

str.toString();
// js: runtime error "Cannot read property 'toString' of undefined"
// ts: type error "Object is possibly 'undefined'"
```

```
interface Student {
  id: number;
  name: string;
  gender: 'male' | 'female';
  age: number;
}

function findStudentById(
  studentList: Student[],
  studentId: number
): Student | undefined {
  if (studentList.length === 0) return undefined;

  return studentList.find((x) => x.id === studentId);
}

const studentList: Student[] = [
  { id: 1, name: 'Alice', gender: 'female', age: 20 },
  { id: 2, name: 'Bob', gender: 'male', age: 20 },
];
const bob = findStudentById(studentList, 3);

console.log(bob.name); // error here
// js: runtime error "Cannot read property 'name' of undefined"
// ts: type error "Object is possibly 'undefined'"

// FIX: do check before using
if (bob) {
  console.log(bob.name);
}
```

4. Option: alwaysStrict

- Khi bật lên true thì typescript sẽ luôn parse files ở chế độ strict mode và sẽ có "use strict" ở đầu file javascript được tạo ra.

```
// tsconfig.json
{
  "compilerOptions": {
    // ...
    "alwaysStrict": true,
  }
}
```

The screenshot shows the TypeScript Playground interface. On the left, the source code defines an interface `Student` and a function `findStudentById`. On the right, the generated JavaScript code is shown, starting with `"use strict";` and containing the transpiled code for the function and the `studentList` array.

5. Other options

#	Option	Desc
1	<code>strictBindCallApply</code>	đảm bảo bind, call và apply được gọi đúng args
2	<code>strictFunctionTypes</code>	đảm bảo đúng dữ liệu cho parameters của function (ko áp dụng cho method)
3	<code>strictPropertyInitializer</code>	đảm bảo thuộc tính của class phải được khởi tạo
4	<code>noImplicitThis</code>	đảm bảo là this không mang kiểu dữ liệu any
5	<code>useUnknownInCatchVariables</code>	đảm bảo là biến trong catch là kiểu unknown (v4.4)

Tham khảo thêm: <https://www.typescriptlang.org/tsconfig#strict>

6. 📝 Notes

- Migrate JS sang TS bằng cách tăng dần độ strict.
- Dùng `strict` option để bật cho tất cả options.
- Khi cần tắt một option nào đó thì vào override lại.
- Không nhất thiết phải nhớ hết tất cả options, chỉ cần biết nó có tồn tại, khi cần thì đi tra cứu lại.

Tham khảo

- <https://www.typescriptlang.org/tsconfig#strict>
 - <https://basarat.gitbook.io/typescript/intro/noimplicitany>
-

Series - Typescript cơ bản 🎨

- Tác giả: **Hậu Nguyễn**
- Được phát hành trên kênh youtube **Easy Frontend**.
- Tài liệu pdf và videos đều có bản quyền thuộc về Easy Frontend.
- Videos được phát hành cho fan cứng trước, public sau.
- [Đăng ký fan cứng](#) để xem series này đầy đủ và sớm nhất nhé.



Kết nối với mình

- ☒ Follow Facebook: <https://www.facebook.com/nvhauesmn/>
- ☒ Like Fanpage: <https://www.facebook.com/learn.easyfrontend>
- ☒ Youtube Channel: <https://www.youtube.com/easyfrontend>