

ОСНОВЫ И ОСОБЕННОСТИ TypeScript

Хворостов В.А. 18.04.2022

Безопасность типов

```
> 3 + []  
'3'
```

```
> let obj = {}  
> obj.foo  
undefined
```

```
> function a(b) {  
  return b/2  
}  
> a("z")  
NaN
```

Выполнение Javascript

- Текст программы преобразуется в абстрактное синтаксическое дерево (АСД)
- АСД компилируется в байт-код
- Байт-код считывается средой выполнения

Компиляция Typescript

- Текст программы преобразуется в абстрактное синтаксическое дерево (АСД)
- АСД проверяется проверкой типов
- АСД компилируется в код на Javascript

Влияние типов на программу

- Типы участвуют только на первых двух шагах процесса компиляции
- Когда TSC компилирует код в Javascript он уже не смотрит на типы
- Это означает, что типы никогда не смогут повлиять на сгенерированный вывод и можно безопасно с ними экспериментировать - обновлять и улучшать их без риска сломать приложение

Установка и использование

```
npm install -g typescript
```

```
tsc hello.ts
```

tsconfig.json

Пример

```
{  
  "compilerOptions": {  
    "outDir": "./built",  
    "allowJs": true,  
    "target": "es5"  
  },  
  "include": ["./src/**/*.ts"]  
}
```

Еще инструменты

- ts-node
<https://www.npmjs.com/package/ts-node>
- webpack
<https://webpack.js.org/guides/typescript/>

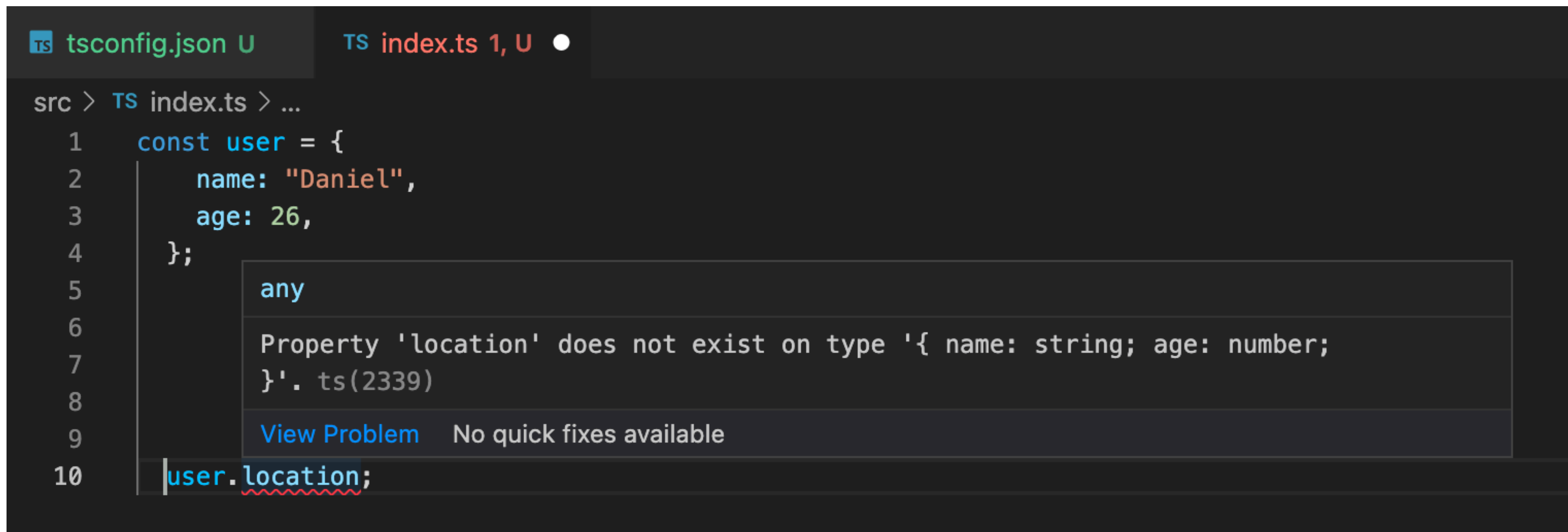
Работа в VSCode

```
src > TS index.ts > ...
1  const message = "Hello World!";
2
3  message.toLowerCase();
4
5  message();
```

```
src > TS const message: "Hello World!"
1  const message = "Hello World!";
2
3  This expression is not callable.
4  Type 'String' has no call signatures. ts(2349)
5  message();
```

[View Problem](#) No quick fixes available

Работа в VSCode



The screenshot shows the VS Code editor interface. The top bar displays two tabs: 'tsconfig.json U' and 'TS index.ts 1, U'. The active file is 'index.ts', which contains the following code:

```
1  const user = {  
2    name: "Daniel",  
3    age: 26,  
4  };  
5  
6  
7  
8  
9  
10 user.location;
```

A tooltip is visible over the `user.location` property access on line 10. The tooltip contains the following information:

- any
- Property 'location' does not exist on type '{ name: string; age: number; }'. ts(2339)
- [View Problem](#) No quick fixes available

The `user.location` property access is underlined with a red wavy line, indicating a TypeScript error.

Проверка аргументов функций

TS tsconfig.json U

TS index.ts 1, U

src > TS index.ts > ...

```
1 // This is an industrial-grade general-purpose greeter function:
2 function greet(person, date) {
3     console.log(`Hello ${person}, today is ${date}!`);
4 }
```

Expected 2 arguments, but got 1. ts(2554)

index.ts(2, 24): An argument for 'date' was not provided.

[View Problem](#) No quick fixes available

```
9 greet("Brendan");
```

Явное объявление типов

TS tsconfig.json U

TS index.ts 1, U ●

src > TS index.ts > ...

```
1  function greet(person: string, date: Date) {  
2    |  console.log(`Hello ${person}, today is ${date.toDateString()}!`);  
3  }  
4  
5  greet("Maddison", Date());
```

Возможные типы

- <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html>

Псевдонимы типов

```
type Point = {  
  x: number;  
  y: number;  
};  
  
// Exactly the same as the earlier example  
function printCoord(pt: Point) {  
  console.log("The coordinate's x value is " + pt.x);  
  console.log("The coordinate's y value is " + pt.y);  
}  
  
printCoord({ x: 100, y: 100 });
```

Еще возможности

- Полиморфизм и обобщенные типы
- Ограничения видимости методов и свойств классов: public, protected, private
- readonly-свойства
- Абстрактные классы, интерфейсы