title: ES6 Set、Map 在 Typescript 中的使用
date: 2019-09-09
tag: ES6, Typescript

description: ES6 中 Set 和 Map 介绍以及在 Typescript 的应用

Typescript 不支持 ES6 中的 Set 和 Map 方法，Github 上有相关的讨论。建议使用第三方库支持，
比如 es6Collection, es6-shims, core-js 等。

Map

对比

- 对象只能用 `字符串` 作为键值，即 `字符串-值`，导致其使用有局限。

```
const obj = { a: 1 };
const data = {};
data[obj] = test;
Object.keys(data) // => ["[object Object]"]
```

- ES6 的 Map 做了扩充，支持 `值-值`

```
const obj = { a: 1 };
const data = {};
const m = new Map();
m.set(obj, 'test')
m.has(obj) // => true
```

Typscript 中使用

定义 Map 类型

```
interface Map<K, V> {
  clear(): void;
  delete(key: K): boolean;
  entries(): IterableIterator<[K, V]>;
  forEach(callbackfn: (value: V, index: K, map: Map<K, V>) => void, thisArg?: any): void;
  get(key: K): V;
  has(key: K): boolean;
  keys(): IterableIterator<K>;
  set(key: K, value?: V): Map<K, V>;
  size: number;
  values(): IterableIterator<V>;
  [Symbol.iterator](): IterableIterator<[K, V]>;
  [Symbol.toStringTag]: string;
}
```

Set

Typscript 中使用

定义 Set 类型

```
interface Set<T> {
  add(value: T): Set<T>;
  clear(): void;
```

```
  delete(value: T): boolean;
  entries(): IterableIterator<[T, T]>;
  forEach(callbackfn: (value: T, index: T, set: Set<T>) => void, thisArg?: any): void;
  has(value: T): boolean;
  keys(): IterableIterator<T>;
  size: number;
  values(): IterableIterator<T>;
  [Symbol.iterator]():IterableIterator<T>;
  [Symbol.toStringTag]: string;
}

interface SetConstructor {
  new <T>(): Set<T>;
  new <T>(iterable: Iterable<T>): Set<T>;
  prototype: Set<any>;
}
declare var Set: SetConstructor;
```