OPEN ORIENTED

凹 凸 实 验 室

ES6 Proxy介绍

Simba

Proxy

Proxy 可以理解成，在目标对象之前架设一层“拦截”，外界对该对象的访问，都必须先通过这层拦截，因此提供了一种机制，可以对外界的访问进行过滤和改写。

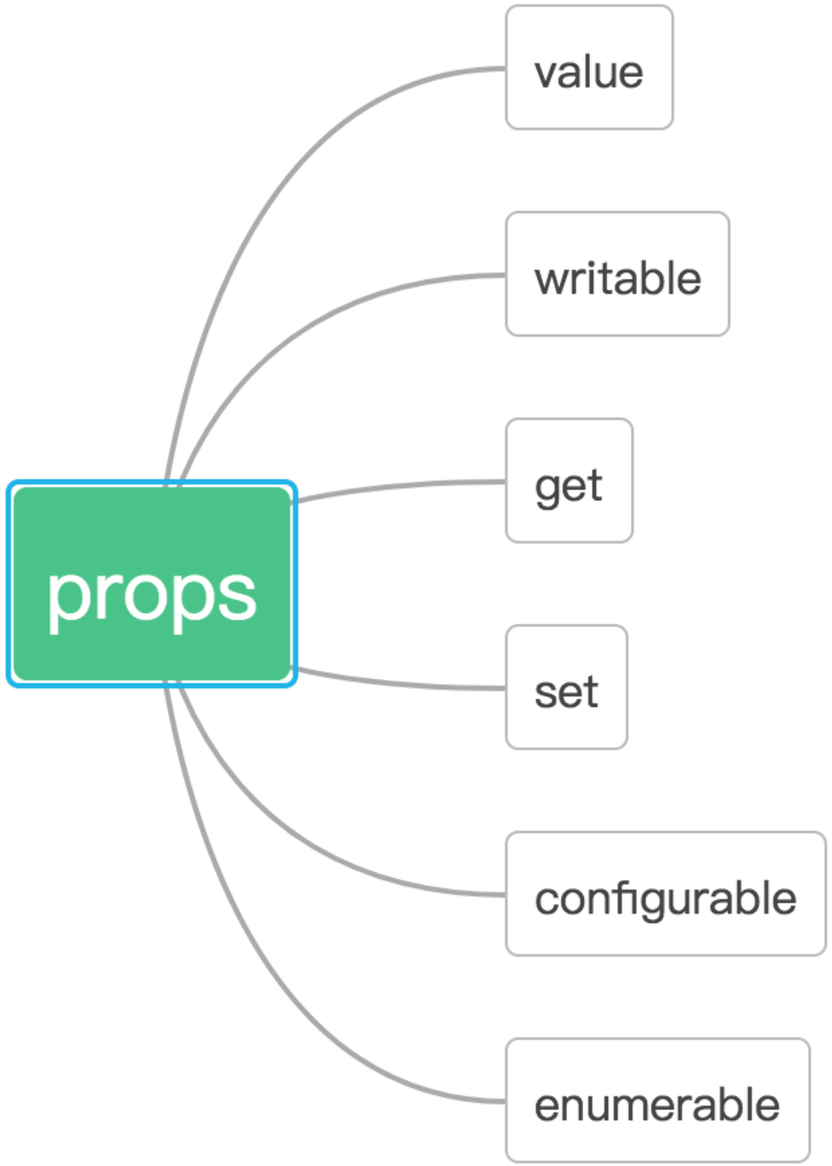
Proxy之前

- Object.defineProperty()
- Object.defineProperties()

```
2
3  var obj = {
4    a: 1
5  }
6
7  Object.defineProperty(obj, 'a', {
8    set: function(o){
9      console.log('set', o)
10   },
11   get: function(){
12     console.log('get')
13   }
14 })
15
```

Object.defineProperty

- 功能有限
- 不够优雅



Proxy

```
21
22  var obj = {
23    a: 1
24  }
25
26  var proxy = new Proxy(obj, {
27    get(tar, key, proxy) {
28      console.log('get')
29      return Reflect.get(tar, key, proxy);
30    },
31    set(tar, key, val) {
32      console.log('set',tar, key, val)
33    }
34  })
35
36  proxy.b = 2
37
```

Reflect-为操作对象而提供的新 API

- 让Objet操作都变成函数行为。例如：

  'assign' in Object // true

  Reflect.has(Object, 'assign') // true

- Reflect对象的方法与Proxy对象的方法一一对应

Reflect提供13个静态方法

Reflect.apply(target,thisArg,args)
Reflect.construct(target,args)
Reflect.get(target,name,receiver)
Reflect.set(target,name,value,receiver)
Reflect.defineProperty(target,name,desc)
Reflect.deleteProperty(target,name)
Reflect.has(target,name)
Reflect.ownKeys(target)
Reflect.isExtensible(target)
Reflect.preventExtensions(target)
Reflect.getOwnPropertyDescriptor(target, name)
Reflect.getPrototypeOf(target)
Reflect.setPrototypeOf(target, prototype)

使用场景一：访问拦截


```
41
42  var obj = {
43    _a: 1
44  }
45
46  var proxy = new Proxy(obj, {
47    get(tar, key, proxy) {
48      if(/^_/.test(key)){
49        return undefined
50      }
51      return Reflect.get(tar, key, proxy);
52    }
53  })
54
55  console.log(proxy._a)
56
```

私有属性

访问日志

非法操作警告

...

使用场景二：校验和过滤


```
58
59  var obj = {
60    a: 1
61  }
62
63  var proxy = new Proxy(obj, {
64    set(tar, key, val) {
65      if(typeof val !== 'number'){
66        throw Error('Invalid.')
67      }
68      return Reflect.set(tar, key, val);
69    }
70  })
71
72  proxy.a = 'a'
73
```

赋值校验

操作过滤

...

其它：修饰器Decorator


```
@testable
class MyTestableClass {
  // ...
}

function testable(target) {
  target.isTestable = true;
}

MyTestableClass.isTestable // true
```

THANKS

FOR YOUR WATCHING

02

OPEN ORIENTED

凹 凸 实 验 室