

Vue.js 响应式原理

作者: [answershuto](#)

原文链接: <https://ld246.com/article/1503297062496>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

写在前面

因为对Vue.js很感兴趣，而且平时工作的技术栈也是Vue.js，这几个月花了些时间研究学习了一下Vue.js源码，并做了总结与输出。

文章的原地址：[answershuto/learnVue](https://answershuto.com/learnVue/)。

在学习过程中，为Vue加上了中文的注释[answershuto/learnVue](https://answershuto.com/learnVue/)，希望对其他想学习Vue源码小伙伴有所帮助。

可能会有理解存在偏差的地方，欢迎提issue指出，共同学习，共同进步。

关于Vue.js

Vue.js是一款MVVM框架，上手快速简单易用，通过响应式在修改数据的时候更新视图。Vue.js的响应式原理依赖于[Object.defineProperty](#)，尤大大在[Vue.js文档](#)中就已经提到过，这也是Vue.js不支持ES8以及更低版本浏览器的原因。Vue通过设定对象属性的 setter/getter 方法来监听数据的变化，通过getter进行依赖收集，而每个setter方法就是一个观察者，在数据变更的时候通知订阅者更新视图。

将数据data变成可观察（observable）的

那么Vue是如何将所有data下面的所有属性变成可观察的（observable）呢？

```
function observer(value) {
  Object.keys(value).forEach((key) => defineReactive(value, key, value[key], cb))
}
```

```
function defineReactive (obj, key, val, cb) {
  Object.defineProperty(obj, key, {
    enumerable: true,
    configurable: true,
    get: ()=>{
      /* ....依赖收集等.... */
    },
    set:newVal=> {
      cb();/*订阅者收到消息的回调*/
    }
  })
}
```

```
class Vue {
  constructor(options) {
    this._data = options.data;
    observer(this._data, options.render)
  }
}
```

```
let app = new Vue({
  el: '#app',
  data: {
    text: 'text',
    text2: 'text2'
  }
})
```

```
    },  
    render(){  
      console.log("render");  
    }  
  })  
})
```

为了便于理解，首先考虑一种最简单的情况，不考虑数组等情况，代码如下所示。在`initData`中会调用`bserve`这个函数将Vue的数据设置成observable的。当`_data`数据发生改变的时候就会触发`set`，对订者进行回调（在这里是`render`）。

那么问题来了，需要对`app. date.text`操作才会触发`set`。为了偷懒，我们需要一种方便的方法通过`app text`直接设置就能触发`set`对视图进行重绘。那么就需要用到代理。

代理

我们可以在Vue的构造函数`constructor`中为`data`执行一个代理`proxy`。这样我们就把`data`上面的属性理到了`vm`实例上。

```
_proxy(options.data);/*构造函数中*/  
  
/*代理*/  
function _proxy (data) {  
  const that = this;  
  Object.keys(data).forEach(key => {  
    Object.defineProperty(that, key, {  
      configurable: true,  
      enumerable: true,  
      get: function proxyGetter () {  
        return that._data[key];  
      },  
      set: function proxySetter (val) {  
        that._data[key] = val;  
      }  
    })  
  })  
};  
}
```

我们就可以用`app.text`代替`app._data.text`了。