

VUE源码“食用”心得

- 邓磊

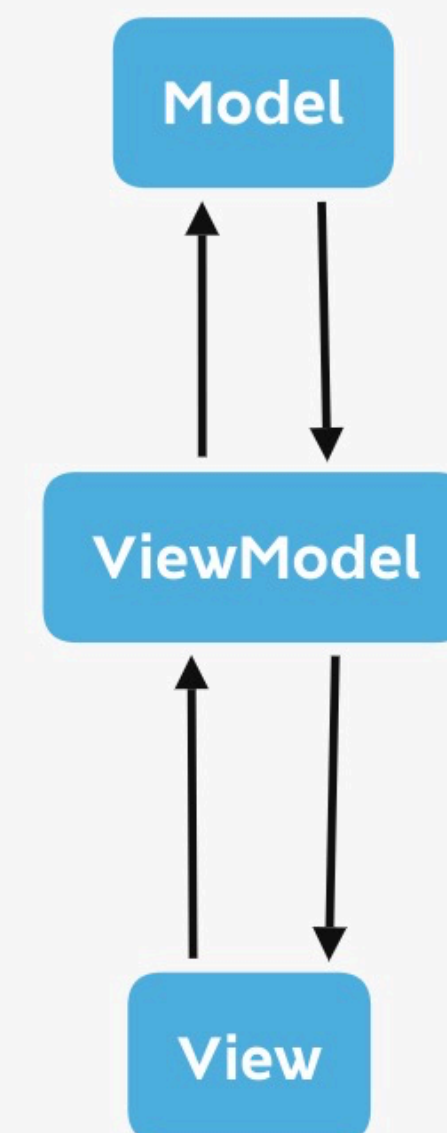
目录

- MVVM设计模式
- 双向绑定原理
- 一个 compute 属性的 “BUG”
- 深入核心源码
- 对比 React 差别



MVVM设计模式

- **Model 模型**
- **ViewModel 视图模型层（胶水）**
- **View 视图层**



“ 在VUE文件里，M、V、VM分别代表那些呢？ ”

```
<template>
  <view class="content">
    <view class="search-box">
      <input type="text" v-model="keyword">
      <text class="button" v-if="keyword" @click="submit">搜索</text>
      <text class="button" v-else @click="clear">取消</text>
    </view>
    <view class="list" v-for="(item, index) in lists" :key="index">
      <text class="name">{{ item.name }}</text>
      <text class="age">{{ item.age }}</text>
    </view>
    <view class="footer" v-if="isShowFooter">{{ footerText }}</view>
  </view>
</template>
```

View

```
<script>
  import Vue from 'vue';

  let hasShow = true;
  export default Vue.extend({
    data: () => ({ ...
  }},
    computed: {
      isShowFooter() {
        console.log('isShowFooter function');
        if (!hasShow) return false;

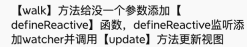
        return this.keyword === '';
      }
    },
    methods: { ...
  }
});
</script>
```

Model

```
<template>
  <view class="content">
    <view class="search-box">
      <input type="text" v-model="keyword">
      <text class="button" v-if="keyword" @click="submit">搜索</text>
      <text class="button" v-else @click="clear">取消</text>
    </view>
    <view class="list" v-for="(item, index) in lists" :key="index">
      <text class="name">{{ item.name }}</text>
      <text class="age">{{ item.age }}</text>
    </view>
    <view class="footer" v-if="isShowFooter">{{ footerText }}</view>
  </view>
</template>
```

ViewModel

VUE代码结构



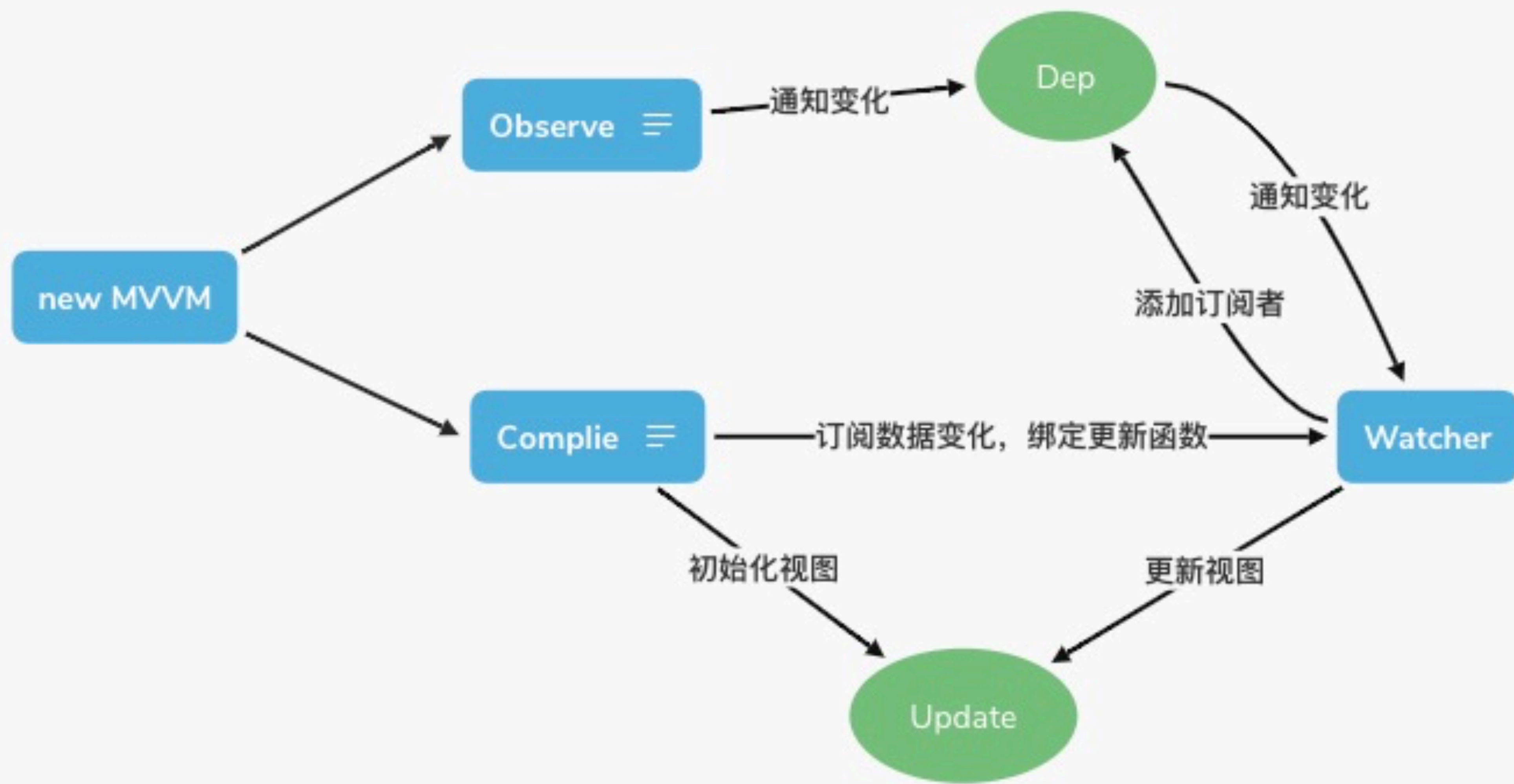
VUE双向绑定原理

简单版 “ Object.defineProperty ”

复杂版！！！！

Observe、Dep、Watcher

三者关系



代码栗子🌰

computed计算属性

例子：computed 属性的一个“BUG”

创建Dep类有哪些? inject、provide、props、\$attrs、\$listeners

1. data 里面的属性会 new Dep 对象

2. props 属性也会 new Dep 对象

3. Observe对象都会 new Dep 对象

创建Watcher类有哪些? computed、watch

代码  案例

```
<template>
  <div>
    <span>{{ person }}</span>
    <span>{{ a }}</span>
    <span>{{ b }}</span>
  </div>
</template>
```

```
<script>
let isLock = false;
export default {
  data: {
    a: 1,
    b: 1
  },
  computed: {
    person() {
      return isLock && this.b === 1;
    }
  }
}
</script>
```

去解决“BUG”



最后几个问题🤔

Vue.\$nextTick 源码解析

Vue Microtask差异, 里面会有一个 **pending** 状态来锁定, 只能同时处理一个函数

1. **window.setImmediate** 默认, 代替 **setTimeout** 做繁重任务
2. **window.MessageChannel** PhantomJS兼容
3. **Promise.resolve** Promise微队列, 会返回一个Promise对象
4. **window.setTimeout** 宏任务队列

Computed 计算属性缓存是什么？

Observer 中的 getter

```
Object.defineProperty(vm, 'sum', {
  get() {
    const watcher = this._computedWatchers && this._computedWatchers[key]
    if (watcher) {
      // ✨上一步中 dirty 已经置为 true, 所以会重新求值
      if (watcher.dirty) {
        watcher.evaluate()
      }
      if (Dep.target) {
        watcher.depend()
      }
      // 最后返回计算出来的值
      return watcher.value
    }
  }
})
```

Dep 中的 notify

```
class Dep {  
  subs = []  
  
  notify () {  
    for (let i = 0, l = subs.length; i < l; i++) {  
      subs[i].update()  
    }  
  }  
}
```

Watcher 中的 update 方法

```
update () {  
  if (this.lazy) {  
    this.dirty = true  
  }  
}
```

beforeCreate 里面可以使用 this 调用 methods 的方法吗？

不能, **beforeCreate** 还没有挂载 **methods** 里面的方法, 也没有对`data`里面的数据双向绑定

对比React差别

- 单向数据流 vs 双向数据流
- 库 vs 框架
- 声明式 vs 渐进式

摘要

- <https://juejin.im/post/5e8fd7a3f265da47c35d7d29> [深度剖析VUE计算属性]
- <https://github.com/vuejs/vue> [VUE仓库]
- <https://cn.vuejs.org/v2/api/> [VUE官网API]

Thanks! ❤️