

link: null  
title: 珠峰架构师成长计划  
description: 要考虑数据驱动视图，多次点击显示多个弹层,给每个层增加 id号,延迟时间和内容  
keywords: null  
author: null  
date: null  
publisher: 珠峰架构师成长计划  
stats paragraph=24 sentences=51, words=477

Message组件的调用方式 #

```
<template>
  <button @click="showMessage">珠峰架构师成长计划</button>
</template>
<script>
export default {
  methods: {
    showMessage() {
      Message.info({ // 直接调用
        content: 'hello 我不帅',
        duration: 3000
      });
      this.$message.info({ // 通过实例调用
        content: 'hello 你很帅',
        duration: 3000
      })
    }
  }
}
</script>
```

编写Message组件 #

要考虑数据驱动视图，多次点击显示多个弹层,给每个层增加 id号,延迟时间和内容

```
<template>
  <div class="messages">
    <!-- 显示在数组中的层 -->
    <div v-for="message in messages" :key="message.id">
      {{message.content}}
    </div>
  </div>
</template>
<script>
export default {
  data() {
    return {
      messages: ['珠峰架构师成长计划', '我很帅'],
      id: 0
    }
  },
  methods: {
    add() {
      let id = this.id++ // id号
      let layers = [...this.messages, id]
      this.$nextTick(() => {
        layers.forEach(layer => {
          this.$message(layer)
        })
      })
    },
    remove() {
      this.$message.closeAll()
      this.messages = this.messages.filter(message => message.id !== this.id)
    }
  }
}
</script>
```

通过js文件控制显示Message弹层 #

- 步骤一:暴露调用方法

```
// 珠峰架构师成长计划
const Message = {
  info(options) {
    // 珠峰架构师成长计划
    getInstance(options).add(options)
  },
  warn() {},
  danger() {},
  success() {}
}
export {
  Message
}
```

- 步骤二:生产实例

```
import Vue from 'vue';
import MessageComponent from './Message.vue';
let getInstance = () =>{
  let vm = new Vue({
    render:h=>h(MessageComponent)
  }).$mount();
  let messageComponent = vm.$children[0];
  // 66B4;66B7;66D6;671F;68B9E;dom65143;67D20;6FF0C;65C06;65176;66302;65728;69875;69762;64E0A;
  document.body.appendChild(vm.$el);
  return {
    add(options){ // 68C03;67528;67EC4;64EF6;67684;add665B9;66CD5;
      messageComponent.add(options);
    }
  }
}
const Message = {
  info(options){
    // 68C03;67528;6589E;652A0;65F39;65C42;665B9;66CD5;
    getInstance().add(options);
  },
  warn() {},
  danger() {},
  success() {}
}
// 66B4;69732; 64E0D;6540C;67C7B;6578B;67684;65F39;65C42;665B9;66CD5;
export {
  Message
}
```

- 步骤三: 实现单例

```
let instance;
let inst = () => {
  instance = instance || getInstance();
  return instance;
}
const Message = {
  info(options){
    // 68C03;67528;6589E;652A0;65F39;65C42;665B9;66CD5;
    inst().add(options);
  },
  warn() {},
  danger() {},
  success() {}
}
// 66B4;69732; 64E0D;6540C;67C7B;6578B;67684;65F39;65C42;665B9;66CD5;
export {
  Message
}
```

## 封装插件 #

install方法和use方法的应用

```
let _Vue;
export default {
  install(Vue,options) {
    if(!_Vue){
      _Vue = Vue;
      let $message = {}
      Object.keys(Message).forEach(type => {
        $message[type] = Message[type]
      });
      Vue.prototype.$message = $message
    }
  }
}
// 67EC4;64EF6;64E2D;64F7F;67528;663D2;64EF6;
import {Message} from './components/Message';
this.$message.info({
  content:'hello 64F60;65F88;65E05;',
  duration:3000
})
```

## Minx方法可以注入数据 #

给每个组件增加公用属性

```
Vue.mixin({
  beforeCreate() {
    if(this.$options.info){
      // 6628A;66570;6636E;66302;65728;681EA;65DF1;67684;_info664E0A;
      this._info = this.$options.info;
    }else{
      this._info = this.$parent && this.$parent._info;
    }
  },
})
```