Vuejs技巧篇

## （1）在v-for中，利用index来对第一项添加class

<**a** class="list-group-item" :class="{'active':!index}" v-for="(i,index) in announcement">

index来源于v-for，i表示内容，index表示索引。

由于index从0开始，因此如果要指定第一项有active这个类，那么就用v-bind:class="{'active':!index}"

第一项原本是false（0），第二项和之后是true（>0），通过逻辑非操作符，让其值反转。

因此，第一项有active这个类，而后面的没有。

如果除了第一项之外有某个类，也可以通过不加逻辑非操作符来达到。

类似的，可以用index==2这样的表达式来让第三项获得这个类。

注意，我的版本是Vuejs2.0，因此貌似不能用$index来替换。

## （2）传递vue组件的实例

①在Vue2.0中，取消了向上和向下传播，建议使用global event bus这种方法（简单来说，这个方法就是创建一个新的vue的实例，然后通过这个方法的实例，来使用$emit和$on来达成组件间的信息传播。

②但有的时候，我们需要使用vue的某个组件。

在使用组件时，我们可以使用ref这个功能来将组件添加到this.$refs.xxxx中（注意，在vue2.0中，具体使用方法改变了，参照<https://github.com/vuejs/vue/issues/2873>搜索refs）。

但是当我使用如何获取[browserify](https://github.com/substack/node-browserify)这个工具时，我发现一件很尴尬的事情，无法获取顶层的组件（准确的说，是\_uid: 3这个vue组件，而$root这个组件的\_uid值为2）。

具体而言，请参照我的github项目<https://github.com/qq20004604/govsite>

无法直接获取到app.vue这个组件。（如果你有方法通过$refs来获取，欢迎评论留言）；

因此，我使用了另外一个办法，修改用于$emit和$on的event bus实例，将app.vue这个组件赋值给这个实例，然后需要使用这个组件时，再通过event bus来获取这个app组件。（如果你看到我的github上面的使用的global-settings，那说明我还没有修改）。

③使用②中方法需要注意的是，我推荐你将app组件传递给需要使用app的组件中的某个变量，然后如果你需要调用app组件的方法或变量时，直接通过这个变量来调用。

特别是当你需要在模板中使用v-if时，如果你需要使用app中的A变量（在app的data属性中），你用另一个组件的B变量来获取A变量，你只能这么写：

**haveLogined**:globalSetting.getAppComponent().**haveLogined**

注：globalSetting.getAppComponent()是获取app组件的方法，返回值是app组件。

而不能这么写：

**app**: globalSetting.getAppComponent(),  
**haveLogined**: this.**app**.**haveLogined**,

这种写法是会报错的

事实上，我推荐省略掉第二个，如果需要使用这个变量名，那么就用app. haveLogined来使用。