# 对vue-chat的理解

## 前端代码分布结构

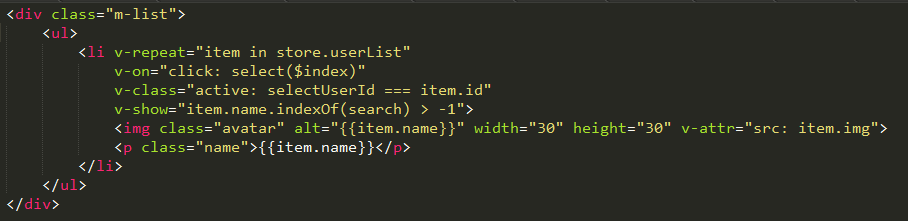
1. 每个组件都有三个文件：
   1. 后缀为.less的文件中定义了一些样式，可以理解为css；
   2. 后缀为.tpl的文件中定义的是模板（注意组件的模板替换了自定义元素，自定义元素的作用只是作为一个挂载点。），在.js文件中，定义某个module的template属性时，通过require语句直接引用；
   3. 后缀为.js的文件是该组件的定义（引用.less,.tpl），通过module.import和require，在整体的chat中使用
2. 在整体的chat部分，同样.less文件中定义了一些样式。在.tpl文件中定义模板，在模板中，把之前定义的组件以tag的形式直接使用。在.js文件中定义了chat。
3. 写好了整体的chat，组件card，message，list，text，之后运用webpack进行编译，编译之后生成了script.js文件
4. 在index.html中，添加script.js，vue.js的引用，定义一个名为chat的块，再定义一些样式。一个美丽网站就大功告成啦~

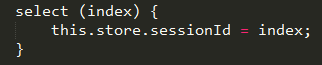
## 渲染过程

1. Card

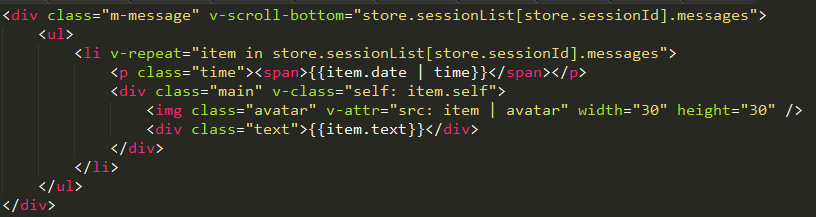
* 1. Input的v-model=”search”存输入
  2. 每当keyup时，调用inputing方法，将search，上传到父组件的方法中
  3. {{store.user.name}} store是通过 props: ['store']父组件传下来的

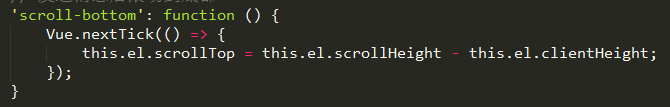
1. List



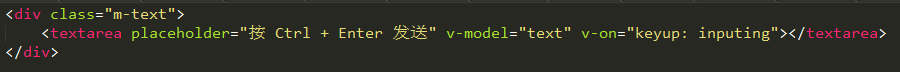
* 1. 只要item in store.userList就输出到列表中。props: ['store'] store是通过props传下来的
  2. 当点击它时，调用select($index)方法，将当前的index存入store.sessionId中
  3. 当当前item.name.indexOf(search)大于-1时，该item才显示，
  4. V-class这句话没太明白>.<????????

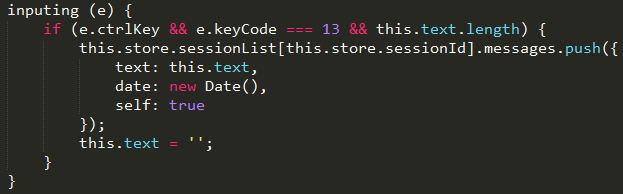
1. Message



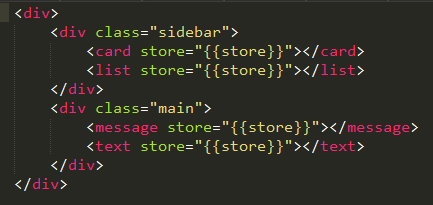
* 1. v-scroll-bottom是自定义指令目的是发送消息后滚动到底部
  2. 如果item在存储器中的消息中，就输出列表
  3. 先输出时间{{item.date | time}}，再输出图片内容等

1. Text



* 1. 输入的内容存入text中
  2. 当keyup时调用inputing方法当按住ctrl和enter键并且text长度大于0，将信息push到messages列表中

1. Chat

把这四个组件组合在chat中