

## vuecli 2.x

笔记本： vue

创建时间： 2021/2/25 15:51

更新时间： 2021/3/13 13:58

作者： 156atulp860

URL： <https://blog.csdn.net/zhouluo/article/details/84752280>

### vuecli 2.x

安装方法: npm install -g vue-cli ,

卸载方法： npm uninstall -g vue-cli

创建项目： vue init webpack my-project

### vuecli>=3

安装方法： npm install -g @vue/cli

卸载方法： npm uninstall -g @vue/cli

创建项目： vue create my-project

注：

两种不能同时存在 当卸载其中一个安装另一个时可能会有如下报错：

```
npm ERR! code EEXIST
npm ERR! path C:\Users\Administrator\AppData\Roaming\npm\node_modules\vue-cli\bin\vue
npm ERR! dest C:\Users\Administrator\AppData\Roaming\npm\vue
npm ERR! EEXIST: file already exists, cmd shim 'C:\Users\Administrator\AppData\Roaming\npm\node_modules\vue-cli\bin\vue' -> 'C:\Users\Administrator\AppData\Roaming\npm\vue'
npm ERR! File exists: C:\Users\Administrator\AppData\Roaming\npm\vue
npm ERR! Remove the existing file and try again, or run npm
npm ERR! with --force to overwrite files recklessly.

npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\Administrator\AppData\Roaming\npm-cache\_logs\2021-02-24T07_33_26_286Z-debug.log
```

上图错误属于原来的版本没有卸载干净 需按照上面路径手动删除关于vue的所有文件

### git 使用：

- 1.本地安装好git；
- 2.桌面右键 Git Bash Here 打开git命令行；
- 3.ssh-keygen -t rsa -C "nideyouxiang@xxx.com" （全部按enter）；
- 4.cd ~/.ssh （如果没有执行第三步，则不会有这个文件夹）；
- 5.cat id\_rsa.pub 在命令行打开这个文件，会直接输出密钥；
- 6.复制，打开github，点自己头像 >> settings >> SSH and GPG keys >> New SSH key
- 7.随便写。key里 粘贴第六步的内容；完成。

### 命令：

```
{git config --global user.email "自己的git账号"
git config --global user.name "给自己起个名字"}让Git知道你是谁 以便提交项目
```

```
git clone + 地址 （拉取项目）
```

#### 提交项目：

- 1.git add .（推到本地git缓存区）
- 2.git commit -m “注释”（推送修改的内容到本地git库中）
- 3.git pull origin 远程分支名（如master）（下拉代码 将远程主干代码与本地代码合并）
- 4.git push origin 远程分支名（如master）（将代码推送到远程）

### vue路由：

在建路由时自己建一个根路由，尽量不要在App.vue中放其他组件，因为在App.vue放组件你无论跳转哪个路由都会使整个vue实例和其他组件重新加载，影响性能。在创建的根组件中插入子组件。

routes 是一个数组 里面可以存在无数个路由对象 每一个单独的路由对象都是打开新的地址和组件（页面），如果在同一个页面那在根路由中配置子路由

请求数据：

1.在vue中请求出来的表格数据标题一般不用v-for 遍历 要写死

2.请求出来的数据id 排序：

```
this.data.sort((a,b)=>{return a.id-b.id}) (从小到大)
this.data.sort((a,b)=>{return b.id-a.id}) (从大到小)
```

3.让请求出来的数据表格某一行或某一列变色：

**ref**被用来给某个元素或者子组件注册引用信息，引用信息将会注册在父组件的\$refs对象上，如果是普通的dom元素上使用 引用指向的是dom元素 如果是子组件中使用 引用指向的是组件实例

```
<div ref='refName'>hello</div> this.$refs.refName.style = "color:red"
```

注意：**ref** 作为渲染结果被创建 在初始渲染时不能访问他们 他们还不存在 所以如果元素或子组件标签上带有 **v-for v-if v-show** 等带有判断性质的语句 不能用**ref**，**\$refs** 也不是响应式的，因此你不应该试图用它在模板中做数据绑定。

**Vue.nextTick()** 在数据更新后和 dom渲染后执行回调 当需要在created ( ) 中操作dom元素时 使用 或者 元素或子组件标签上带有 v-for v-if v-show 等带有判断性质的语句可以操作dom元素 或 在要等到一个dom更新渲染后在操作此dom

```
<div ref='refName' v-if = 'xxx'>hello</div>
this.$nextTick(()=>{
  this.$refs.refName.style = "color:red"
})
```

vue增删改查：

增：

创建一个新对象：**megBoxData**:{} 用来放v-model 所绑定需要增加的各个字段和数据 此时要在新对象中加入id字段 = 原对象的id 设置新增数据id方法  
然后再用原始数据通过push方法把这些字段都赋值到原始数据对应的字段中

```
addBtn(){
    let _id = Math.max(...this.personData.map((item)=>{return item.per_id}))+1
    this.personData.push({
      name:this.megBoxData.name,
      sex:this.megBoxData.sex,
      address:this.megBoxData.address,
      time:this.megBoxData.time,
      email:this.megBoxData.email,
      phone:this.megBoxData.phone,
      per_id:_id
    })
    this.megBoxData={} //清空input中的值
    this.isMessageBox = false //关闭弹层
  },
```

删：传入索引 index 然后用原始数据通过**splice ( index , 1 )** 来删除

改：创建一个新对象 用来放v-model 所绑定需要增加的各个字段和数据 **修改**：把原始对象传到方法里 然后再把原始对象赋值到新对象上 此时要在新对象中加入id字段 = 原对象的id

```
edit(d){
    this.isMegEditBox = true

    // this.megEditBox=d
    this.megEditBox={
      name:d.name,
      sex:d.sex,
      address:d.address,
      time:d.time,
      email:d.email,
      phone:d.phone,
```

```

        per_id:d.per_id
      }
    },
  },
},

```

**更新：**遍历原始数组 判断原始对象中的id == 新对象id 把新对象赋值到原对象上 此时可能会遇到数据不立即渲染的情况用Vue.set()即可。

```

editBtn(){
  this.personData.forEach((item,index)=>{
    if (item.per_id == this.megEditBox.per_id) {
      this.personData[index] = this.megEditBox
      this.$set( this.personData, index, this.megEditBox )
    }
  })
  this.isMegEditBox = false
},

```

**查询：**新建一个新数组 然后遍历原数组 用v-model来绑定输入框的内容 判断要搜索的字段是否在原数组对象中 如果在用新数据的push ( ) 方法加入符合条件的数组 返回新数组 最后把新数组赋值到原数组上即可

```

search(){
  let arrNews = []
  this.personData.map((items)=>{
    if (items.name.search(this.searchDat) != -1 ) {
      arrNews.push(items)
    }
  })
  return arrNews
})
this.personData = arrNews
this.searchDat = ""
},

```