### todos接口案例

#### 1 请求后台接口list 获取 数据展示到页面上

* 发送ajax请求
* 在mounted 钩子中将数据渲染到页面上

axios.defaults.baseURL = 'http://localhost:3001/';  
​  
new Vue({  
    data: {  
               currentEditing: null,  
               todos: []  
    },   
 mounted() {  
        # 1.2 在mounted 钩子中将数据渲染到页面上  
               this.getList()  
          },  
     methods: {  
        # 1.1 发送ajax请求  
               getList() {  
                   axios.get('list').then(res => {  
                       console.log(res.data)  
                       this.todos = res.data  
                  })  
              }  
      }            
​  
})

#### 2 增加内容

* 在回车事件中发送ajax请求
  + 我们已经封装好的 addTodo 方法中发送ajax请求
* 将用户输入的内容添加到后台
  + 把用户输入的内容传递给后台
* 因为我们所有的数据增删改查都是在父组件中修改 所有ajax也在父组件的 方法中发送

addTodo(todoText) {  
                   axios.post('add', {  
                           title: todoText,  
                      }).then(res => {  
                      # 在请求成功之后重新渲染页面  
                           this.getList()  
                      })  
                   // 清空文本框  
                   event.target.value = ''  
},

#### 3 双击修改列表信息

* 按回车键或者失去焦点的时候获取到用户信息
  + 这里需要把当前列表信息的id获取到
* 把收集到的用户信息发送到后台

## 子组件中   需要把当前id 传递过去  
<input class="edit"   
   @keyup.enter="saveEdit(item, $event)"   
   @blur="saveEdit(item, $event)"   
    :value="item.title"   
   @keyup.esc="currentediting = null">     
          
saveEdit(item, event) {  
                   var editText = event.target.value.trim()  
                   console.log(item, '--------')  
                   this.$emit("save-edit", editText, item)  
},  
​  
### 父组件方法  
### 因为我们所有的数据增删改查都是在父组件中修改 所有ajax也在父组件的 方法中发送  
// 保存编辑项  
        saveEdit(editText, item) {  
axios.put(`list/${item.id}`, {  
title: editText,  
                  }).then(res => {  
                       this.getList()  
                        // 3. 去除 editing 样式  
                         this.currentEditing = null  
                  })  
​  
                    
              }

####4 删除列表信息

* 点击按钮 删除当前列表信息
  + 需要把当前列表的id传递过去
* 发送ajax请求 根据id 删除当前信息

# 4.1 需要把删除按钮的id 传递过去   
<button class="destroy" @click="removeTodoParent(item.id, $event)"></button>  
​  
​  
​  
​  
# 父组件  
### 因为我们所有的数据增删改查都是在父组件中修改 所有ajax也在父组件的 方法中发送  
​  
 // 删除任务项  
 removeTodo(delIndex) {  
     axios.delete(`list/${delIndex}`).then(res => {  
         this.getList()  
    })  
},  
​  
​

#### 5 删除全部完成

* 在fotter 的Clear completed 的事件处理函数中发送ajax 请求删除所有complete 为true 的

# 父组件  
### 因为我们所有的数据增删改查都是在父组件中修改 所有ajax也在父组件的 方法中发送  
  removeAllDone() {  
      axios.delete('deleteall').then(res=>  
          this.getList()  
      })

#### 6 实现全选功能

* 在toggleStat1 中发送ajax请求 根据当前复选框选中状态 修改后台数据中对应的 complete 值

 #2 把数据更新放在父组件中   
<mylist @get-list="getList()" ></mylist>  
​  
computed: {  
               toggleStat1: {  
                   get() {  
                       return this.todos.every(item => item.completed)  
                  },  
                   set(val) {  
                       axios.get(`all/${Number(val)}`).then(res => {  
                           #1 子组件通知父组件更新数据  
                           this.$emit('get-list')  
                      })  
​  
                  }  
              }  
}