

<template>

    <div>

        <label>编号</label><input type="text" max="10" v-model.trim="todoId" />

        <br/>

        <label>任务名称</label><input type="text" maxlength="50" v-model.trim="todoElement" />

        <br/>

        <button @click="submintodo">保存 </button>

    </div>

</template>

<script setup>

import { ref } from 'vue'

const todoElement=ref('')

const todoId=ref('')

//注册点击事件

const emit=defineEmits(['add-todo'])

//保存函数的编写

const submintodo=()=>{

    if (todoElement.value){

        console.log(todoElement.value)

        emit('add-todo',todoElement.value,todoId.value)

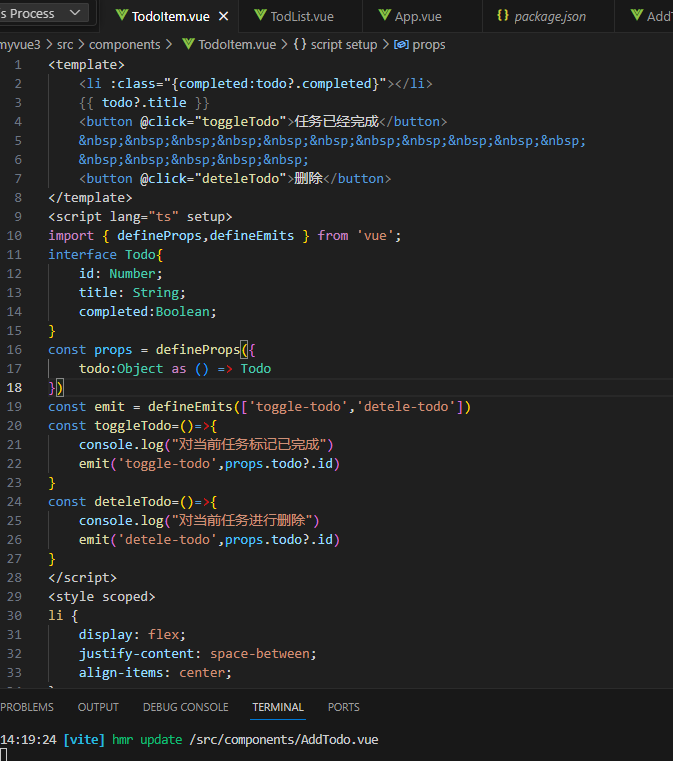
        todoElement.value=' '

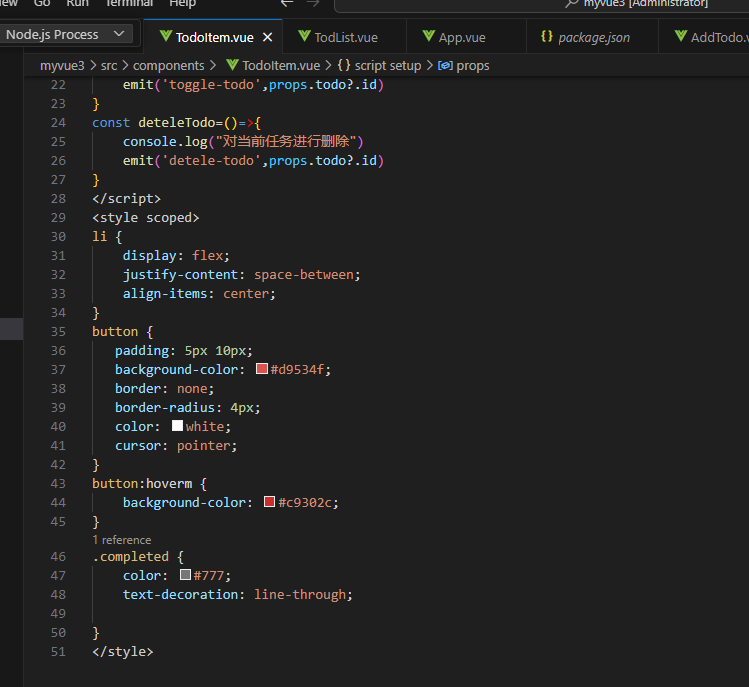
        todoId.value=' '

    }

}

</script>





<template>

    <li :class="{completed:todo?.completed}"></li>

    {{ todo?.title }}

    <button @click="toggleTodo">任务已经完成</button>

    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

    <button @click="deteleTodo">删除</button>

</template>

<script lang="ts" setup>

import { defineProps,defineEmits } from 'vue';

interface Todo{

    id: Number;

    title: String;

    completed:Boolean;

}

const props = defineProps({

    todo:Object as () => Todo

})

const emit = defineEmits(['toggle-todo','detele-todo'])

const toggleTodo=()=>{

    console.log("对当前任务标记已完成")

    emit('toggle-todo',props.todo?.id)

}

const deteleTodo=()=>{

    console.log("对当前任务进行删除")

    emit('detele-todo',props.todo?.id)

}

</script>

<style scoped>

li {

    display: flex;

    justify-content: space-between;

    align-items: center;

}

button {

   padding: 5px 10px;

   background-color: #d9534f;

   border: none;

   border-radius: 4px;

   color: white;

   cursor: pointer;

}

button:hoverm {

    background-color: #c9302c;

}

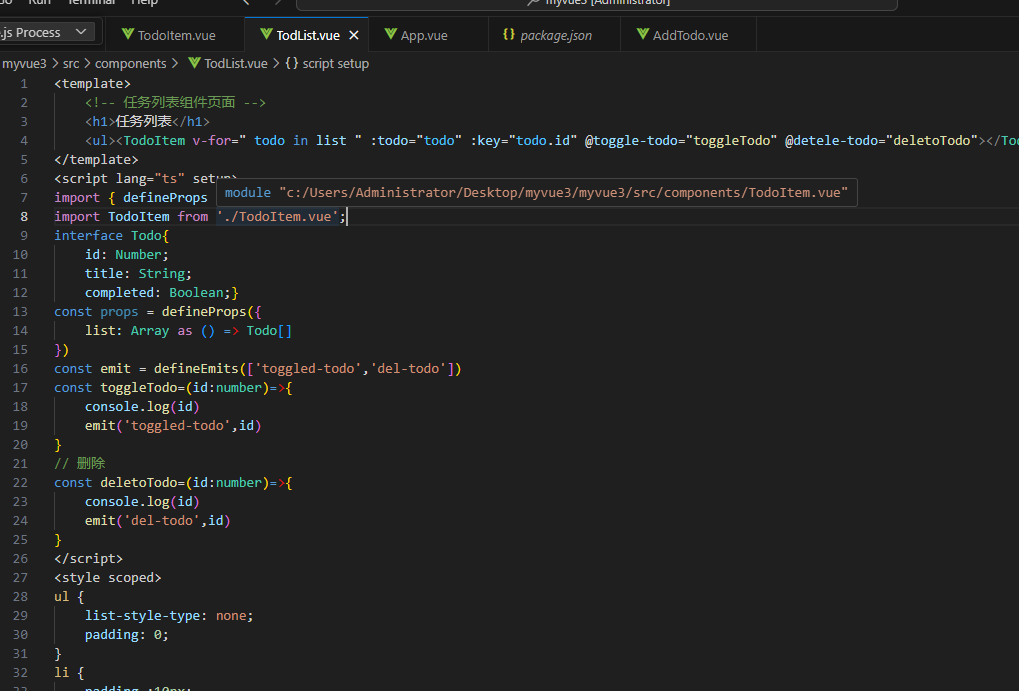
.completed {

    color: #777;

    text-decoration: line-through;

}

</style>



<template>

    <!-- 任务列表组件页面 -->

    <h1>任务列表</h1>

    <ul><TodoItem v-for=" todo in list " :todo="todo" :key="todo.id" @toggle-todo="toggleTodo" @detele-todo="deletoTodo"></TodoItem></ul>

</template>

<script lang="ts" setup>

import { defineProps } from 'vue';

import TodoItem from './TodoItem.vue';

interface Todo{

    id: Number;

    title: String;

    completed: Boolean;}

const props = defineProps({

    list: Array as () => Todo[]

})

const emit = defineEmits(['toggled-todo','del-todo'])

const toggleTodo=(id:number)=>{

    console.log(id)

    emit('toggled-todo',id)

}

// 删除

const deletoTodo=(id:number)=>{

    console.log(id)

    emit('del-todo',id)

}

</script>

<style scoped>

ul {

    list-style-type: none;

    padding: 0;

}

li {

    padding :10px;

    border-bottom: 1px solid #ddd;

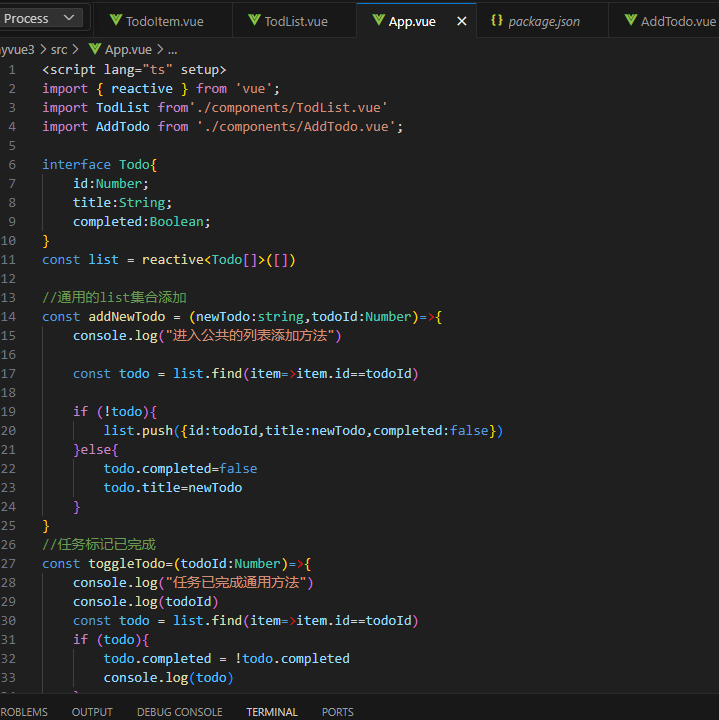
}

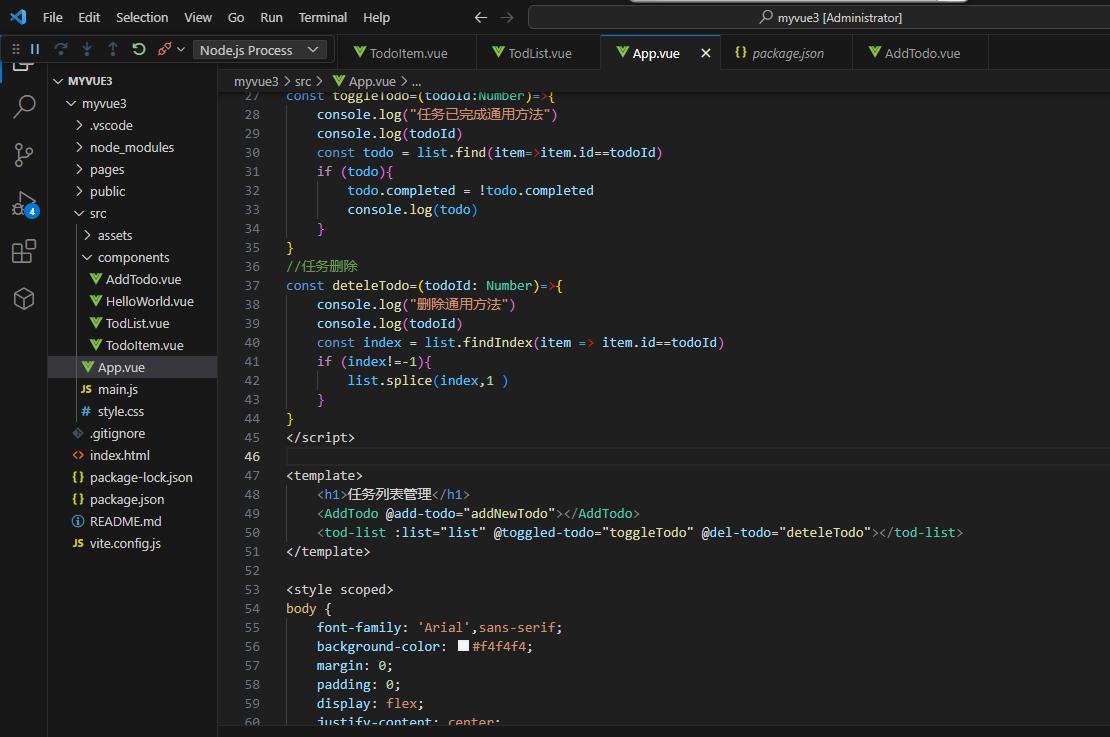
li:last-child {

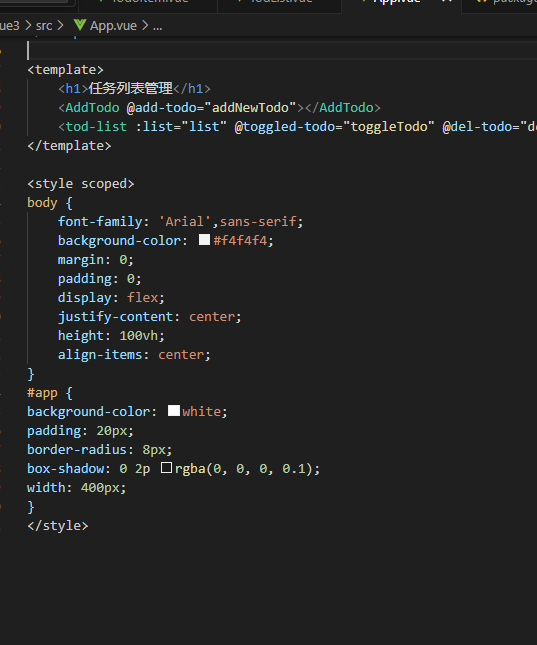
    border-bottom:none;

}

</style>







<script lang="ts" setup>

import { reactive } from 'vue';

import TodList from'./components/TodList.vue'

import AddTodo from './components/AddTodo.vue';

interface Todo{

    id:Number;

    title:String;

    completed:Boolean;

}

const list = reactive<Todo[]>([])

//通用的list集合添加

const addNewTodo = (newTodo:string,todoId:Number)=>{

    console.log("进入公共的列表添加方法")

    const todo = list.find(item=>item.id==todoId)

    if (!todo){

        list.push({id:todoId,title:newTodo,completed:false})

    }else{

        todo.completed=false

        todo.title=newTodo

    }

}

//任务标记已完成

const toggleTodo=(todoId:Number)=>{

    console.log("任务已完成通用方法")

    console.log(todoId)

    const todo = list.find(item=>item.id==todoId)

    if (todo){

        todo.completed = !todo.completed

        console.log(todo)

    }

}

//任务删除

const deteleTodo=(todoId: Number)=>{

    console.log("删除通用方法")

    console.log(todoId)

    const index = list.findIndex(item => item.id==todoId)

    if (index!=-1){

        list.splice(index,1 )

    }

}

</script>

<template>

    <h1>任务列表管理</h1>

    <AddTodo @add-todo="addNewTodo"></AddTodo>

    <tod-list :list="list" @toggled-todo="toggleTodo" @del-todo="deteleTodo"></tod-list>

</template>

<style scoped>

body {

    font-family: 'Arial',sans-serif;

    background-color: #f4f4f4;

    margin: 0;

    padding: 0;

    display: flex;

    justify-content: center;

    height: 100vh;

    align-items: center;

}

#app {

background-color: white;

padding: 20px;

border-radius: 8px;

box-shadow: 0 2p rgba(0, 0, 0, 0.1);

width: 400px;

}

</style>

