

keep-alive

缓存组件，提升性能 被缓存的组件的created只会在首次进入页面的时候执行

Actived()钩子会在每次进入都执行

场景： 从首页进入详情页，每次都是同一个ID 就不用发送请求了。

Scoped()

实现原理：给元素节点新增data-私有属性，从而实现CSS层面的样式隔离

样式穿透

SCSS 父元素 /deep/ 子元素

Stylus 父元素 /deep/ 子元素; >>>

Computed

特点： 基于响应式依赖进行缓存 只在属性发生改变时候 才会重新求值

Vuex的持久化存储

1： localstorage

2： 插件

Vue-router的实现原理

1： Vue.router 是 Vue的一个插件 遵守插件的注册方法 内部实现了install方法；

2： 两种模式 hash history 对应两种实现方式

Hash： HashHistory Hash不会包含在HTTP请求中 不会重新加载页面

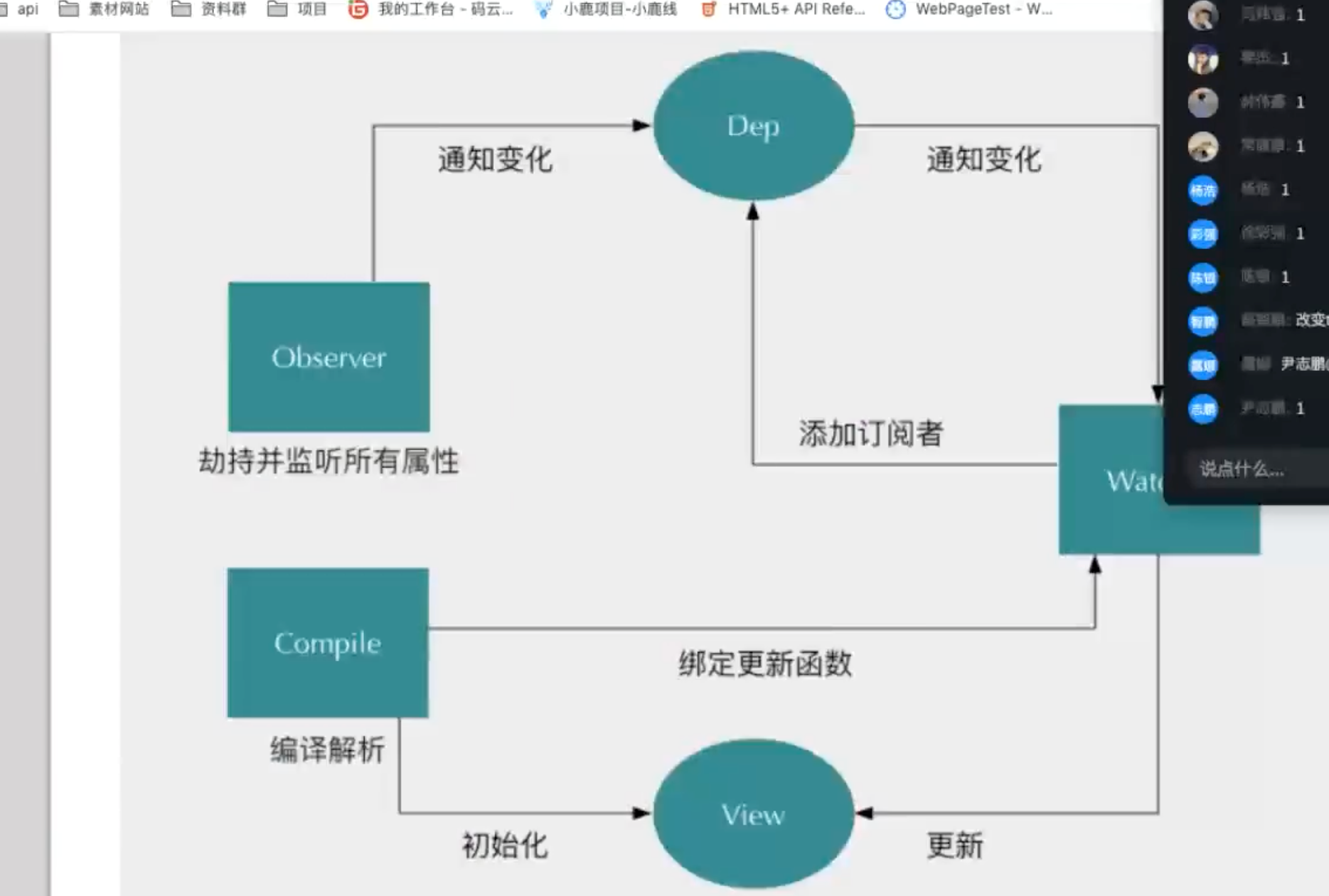
History：HTML5History pushState() replaceState() 修改浏览器的历史记录栈 URL改变 但是不会刷新页面

3： 监听地址栏的改变 监听 “hashchange事件”

4： 注册了两个全局组件 router-view router-link



MVVM



$el 在有值的时候 和$mounted()那个优先级高 ？？

优先级排序 ： el > $mounted(‘dom’)

原因： el是直接在new vue的option里面 ； $mounted（）是在实例化vue后调用的

两者只需要写一个即可

vue会在挂载时，将el /mounted(‘dom’) 传入的DOM的innerHTML全部替换掉

new Vue({

el: "#app1",

render: (h) => h(App),

}).$mount("#app");

Template and Render 同时存在那个优先级高

Render: 的优先级

路由守卫的三种方式

beforeEach() afterEach()

路由独享守卫 beforeEnter() 只有在从一个不同的路由导航时才触发

组件守卫 beforeRouterEnter() 不能访问this ；

在渲染该组件对应的路由被验证前调用

beforeEnterUpdate()

beforeRouteLeave()

$route的实现方式

$store 的实现方式