

一、什么是MVVM？

MVVM是Model-View-ViewModel的缩写。MVVM是一种设计思想。Model 层代表数据模型，也可以在Model中定义数据修改和操作的业务逻辑；View 代表UI 组件，它负责将数据模型转化成UI 展现出来，ViewModel 是一个同步View 和Model的对象（桥梁）。

在MVVM架构下，View 和 Model 之间并没有直接的联系，而是通过ViewModel进行交互，Model 和 ViewModel 之间的交互是双向的，因此View 数据的变化会同步到Model中，而Model 数据的变化也会立即反应到View 上。

ViewModel 通过双向数据绑定把 View 层和 Model 层连接了起来，而View 和 Model 之间的同步工作完全是自动的，无需人为干涉，因此开发者只需关注业务逻辑，不需要手动操作DOM，不需要关注数据状态的同步问题，复杂的数据状态维护完全由 MVVM 来统一管理。

二、mvvm和mvc区别？它和其它框架（jquery）的区别是什么？哪些场景适合？

mvc和mvvm其实区别并不大。都是一种设计思想。主要就是mvc中Controller演变成mvvm中的viewModel。mvvm主要解决了mvc中大量的DOM 操作使页面渲染性能降低，加载速度变慢，影响用户体验。

区别：vue数据驱动，通过数据来显示视图层而不是节点操作。

场景：数据操作比较多的场景，更加便捷

三、vue的优点是什么？

- 低耦合。视图（View）可以独立于Model变化和修改，一个ViewModel可以绑定到不同的"View"上，当View变化的时候Model可以不变，当Model变化的时候View也可以不变。
- 可重用性。你可以把一些视图逻辑放在一个ViewModel里面，让很多view重用这段视图逻辑。
- 独立开发。开发人员可以专注于业务逻辑和数据的开发（ViewModel），设计人员可以专注于页面设计。
- 可测试。界面素来是比较难于测试的，而现在测试可以针对ViewModel来写。

四、组件之间的传值？

· 父组件与子组件传值

父组件通过标签上面定义传值 `:eg='data'` 父组件中 `data(){return {data:'egdata'}}`

子组件通过props方法接受数据 `props:['eg']` 在 `props` 中添加了元素之后，就不需要在 `data` 中再添加变量了

· 子组件向父组件传递数据

子组件通过`$emit`方法传递参数

子组件部分：

父组件中：

子组件向子组件传递数据

Vue 没有直接子对子传参的方法，建议将需要传递数据的子组件，都合并为一个组件。如果一定需要子对子传参，可以先从传到父组件，再传到子组件。

为了便于开发，Vue 推出了一个状态管理工具 Vuex，可以很方便实现组件之间的参数传递

五、路由之间跳转

声明式（标签跳转） 编程式（js跳转）

1. 直接修改地址栏中的路由地址 ②通过router-link实现跳转

③通过js的编程的方式

六、vue.cli中怎样使用自定义的组件？有遇到过哪些问题吗？

- 第一步：在components目录新建你的组件文件（如：indexPage.vue），script一定要export default {}
- 第二步：在需要用的页面（组件）中导入：import indexPage from '@/components/indexPage.vue'
- 第三步：注入到vue的子组件的components属性上面，components:{indexPage}
- 第四步：在template视图view中使用，
例如有indexPage命名，使用的时候则index-page

七、vue如何实现按需加载配合webpack设置

webpack中提供了require.ensure()来实现按需加载。以前引入路由是通过import 这样的方式引入，改为const定义的方式进行引入。

不进行页面按需加载引入方式：import home from '.././common/home.vue'

进行页面按需加载的引入方式：const home = r => require.ensure([], () => r
(require('.././common/home.vue')))

在音乐app中使用的路由懒加载方式为：

复制代码

```
const Recommend = (resolve) => {  
  
  import('components/recommend/recommend').then((module) => {
```

```

    resolve(module)
  })
}

const Singer = (resolve) => {
  import('components/singer/singer').then((module) => {
    resolve(module)
  })
}
复制代码

```

八、vuex面试相关

(1) vuex是什么？怎么使用？哪种功能场景使用它？

vue框架中状态管理。在main.js引入store，注入。新建一个目录store，.... export。场景有：单页应用中，组件之间的状态。音乐播放、登录状态、加入购物车

main.js:

复制代码

```
import store from './store'
```

```

new Vue({
  el: '#app',
  store
})
复制代码

```

(2) vuex有哪几种属性？

有五种，分别是 State、Getter、Mutation、Action、Module

- vuex的State特性

A、Vuex就是一个仓库，仓库里面放了很多对象。其中state就是数据源存放地，对应于一般Vue对象里面的data

B、state里面存放的数据是响应式的，Vue组件从store中读取数据，若是store中的数据发生改变，依赖这个数据的组件也会发生更新

C、它通过mapState把全局的 state 和 getters 映射到当前组件的 computed 计算属性中

- vuex的Getter特性

A、getters 可以对State进行计算操作，它就是Store的计算属性

B、虽然在组件内也可以做计算属性，但是getters 可以在多组件之间复用

C、如果一个状态只在一个组件内使用，是可以不用getters

- vuex的Mutation特性

Action 类似于 mutation，不同在于：Action 提交的是 mutation，而不是直接变更状态；Action 可以包含任意异

步操作。

(3) 不用Vuex会带来什么问题？

- 可维护性会下降，想修改数据要维护三个地方；
- 可读性会下降，因为一个组件里的数据，根本就看不出来是从哪来的；
- 增加耦合，大量的上传派发，会让耦合性大大增加，本来Vue用Component就是为了减少耦合，现在这么用，和组件化的初衷相背。

九、v-show和v-if指令的共同点和不同点

- v-show指令是通过修改元素的display的CSS属性让其显示或者隐藏
- v-if指令是直接销毁和重建DOM达到让元素显示和隐藏的效果

十、如何让CSS只在当前组件中起作用

将当前组件的<style>修改为<style scoped>

十一、<keep-alive></keep-alive>的作用是什么？

<keep-alive></keep-alive> 包裹动态组件时，会缓存不活动的组件实例，主要用于保留组件状态或避免重新渲染。

十二、Vue中引入组件的步骤？

1) 采用ES6的import ... from ...语法或CommonJS的require()方法引入组件

2) 对组件进行注册,代码如下

```
1. // 注册
2. Vue.component('my-component', {
3.   template: '<div>A custom component!</div>'
4. })
5.
```

3) 使用组件<my-component></my-component>

十三、指令v-el的作用是什么？

提供一个在页面上已存在的 DOM 元素作为 Vue 实例的挂载目标.可以是 CSS 选择器，也可以是一个 HTMLElement 实例

十四、在Vue中使用插件的步骤

- 采用ES6的import ... from ...语法或CommonJS的require()方法引入插件
- 使用全局方法Vue.use(plugin)使用插件,可以传入一个选项对象Vue.use(MyPlugin, { someOption:

```
true })
```

如使用懒加载插件：

```
Vue.use(VueLazyload, {  
  loading: require('common/image/default.png')  
})
```

十五、请列举出3个Vue中常用的生命周期钩子函数（见博客）

- `created`：实例已经创建完成之后调用，在这一步，实例已经完成数据观测，属性和方法的运算，`watch/event`事件回调。然而，挂载阶段还没有开始，`$el`属性目前还不可见
- `mounted`：`el`被新创建的 `vm.$el` 替换，并挂载到实例上去之后调用该钩子。如果 `root` 实例挂载了一个文档内元素，当 `mounted` 被调用时 `vm.$el` 也在文档内。
- `activated`：`keep-alive`组件激活时调用

十六、`active-class`是哪个组件的属性？

`vue-router`模块的`router-link`组件。

十七、怎么定义`vue-router`的动态路由以及如何获取传过来的动态参数？

在`router`目录下的`index.js`文件中，对`path`属性加上`/:id`。
使用`router`对象的`params.id`。

十八、`vue-router`有哪几种导航钩子？（见博客）

三种，一种是全局导航钩子：`router.beforeEach(to, from, next)`，作用：跳转前进行判断拦截。
第二种：组件内的钩子；
第三种：单独路由独享组件

十九、生命周期相关面试题

总共分为8个阶段创建前/后，载入前/后，更新前/后，销毁前/后。

- 创建前/后：在`beforeCreate`阶段，`vue`实例的挂载元素`el`和数据对象`data`都为`undefined`，还未初始化。在`created`阶段，`vue`实例的数据对象`data`有了，`el`还没有。
- 载入前/后：在`beforeMount`阶段，`vue`实例的`$el`和`data`都初始化了，但还是挂载之前为虚拟的`dom`节点，`data.message`还未替换。在`mounted`阶段，`vue`实例挂载完成，`data.message`成功渲染。
- 更新前/后：当`data`变化时，会触发`beforeUpdate`和`updated`方法。
- 销毁前/后：在执行`destroy`方法后，对`data`的改变不会再触发周期函数，说明此时`vue`实例已经解除了事件监听以及和`dom`的绑定，但是`dom`结构依然存在

（1）、什么是`vue`生命周期

答：Vue 实例从创建到销毁的过程，就是生命周期。也就是从开始创建、初始化数据、编译模板、挂载Dom→渲染、更新

→渲染、卸载等一系列过程，我们称这是 Vue 的生命周期。

(2)、vue生命周期的作用是什么

答：它的生命周期中有多个事件钩子，让我们在控制整个Vue实例的过程时更容易形成好的逻辑。

(3)、第一次页面加载会触发哪几个钩子

答：第一次页面加载时会触发 beforeCreate, created, beforeMount, mounted 这几个钩子

(4)、DOM 渲染在哪个周期中就已经完成

答：DOM 渲染在 mounted 中就已经完成了。

(5)、简单描述每个周期具体适合哪些场景

答：生命周期钩子的一些使用方法：

- beforecreate : 可以在这加个loading事件，在加载实例时触发
- created : 初始化完成时的事件写在这里，如在这结束loading事件，异步请求也适宜在这里调用
- mounted : 挂载元素，获取到DOM节点
- updated : 如果对数据统一处理，在这里写上相应函数
- beforeDestroy : 可以做一个确认停止事件的确认框
- nextTick : 更新数据后立即操作dom

二十、说出至少4种vue当中的指令和它的用法？

v-if : 判断是否隐藏 ; v-for : 数据循环 ; v-bind:class : 绑定一个属性 ; v-model : 实现双向绑定

二十一、vue-loader是什么？使用它的用途有哪些？

解析.vue文件的一个加载器。(深入理解见<https://www.jb51.net/article/115480.htm>)

用途：js可以写es6、style样式可以scss或less、template可以加jade等

根据官网的定义，vue-loader 是 webpack 的一个 loader，用于处理 .vue 文件。

其次，使用vue-cli脚手架，作者已经配置好了基本的配置，开箱即用，你需要做的就是npm install 安装下依赖，然后就可以开发业务代码了。当然，如果你想进阶，最好熟悉下vue-loader的具体配置，而不要依赖脚手架

二十二、scss是什么？在vue.cli中的安装使用步骤是？有哪几大特性？

答：css的预编译。(scss是sass的一个升级版本，完全兼容sass之前的功能，又有了些新增能力，最主要的就是sass是靠缩进表示嵌套关系，scss是花括号)

使用步骤：

第一步：先装css-loader、node-loader、sass-loader等加载器模块

第二步：在build目录找到webpack.base.config.js，在那个extends属性中加一个拓展.scss

第三步：在同一个文件，配置一个module属性

第四步：然后在组件的style标签加上lang属性，例如：lang="scss"

特性：

- 可以用变量，例如（\$变量名称=值）；
- 可以用混合器，混入@mixin 可以传变量
- 可以嵌套

继承@extend 不可以传变量，相同样式直接继承，不会造成代码冗余；基类未被继承时，也会被编译成css代码

二十三、为什么使用key？

当有相同标签名的元素切换时，需要通过 key 特性设置唯一的值来标记以让 Vue 区分它们，否则 Vue 为了效率只会替换相同标签内部的内容。

二十四、为什么避免 v-if 和 v-for 用在一起

当 Vue 处理指令时，v-for 比 v-if 具有更高的优先级，这意味着 v-if 将分别重复运行于每个 v-for 循环中。通过v-if 移动到容器元素，不会再重复遍历列表中的每个值。取而代之的是，我们只检查它一次，且不会在 v-if 为否的时候运算 v-for。

二十五、VNode是什么？虚拟 DOM是什么？ 详情见 <https://www.jb51.net/article/105221.htm>

Vue在 页面上渲染的节点，及其子节点称为“虚拟节点 (Virtual Node)”，简称为“VNode”。“虚拟 DOM”是由 Vue 组件树建立起来的整个 VNode 树的称呼。

2018 vue前端面试题

1、active-class是哪个组件的属性？嵌套路由怎么定义？

答：vue-router模块的router-link组件。

嵌套路由顾名思义就是路由的多层嵌套。 一级路由里面使用children数组配置子路由，就是嵌套路由。

2、怎么定义vue-router的动态路由？怎么获取传过来的动态参数？

答：在router目录下的index.js文件中，对path属性加上/:id。 使用router对象的params.id

4、scss是什么？安装使用的步骤是？有哪几大特性？

答：预处理css，把css当前函数编写，定义变量,嵌套。 先装css-loader、node-loader、sass-loader等加载器模块，在webpack-base.config.js配置文件中加多一个拓展:extenstion，再加多一个模块：module里面test、loader

5、mint-ui是什么？怎么使用？说出至少三个组件使用方法？

<https://www.cnblogs.com/stella1024/p/7771334.html>

答：基于vue的前端组件库。npm安装，然后import样式和js，vue.use (mintUi) 全局引入。在单个组件局部引入：import {Toast} from 'mint-ui'。组件一：Toast('登录成功')；组件二：mint-header；组件三：mint-swiper

6、v-model是什么？怎么使用？ vue中标签怎么绑定事件？

答：可以实现双向绑定，指令（v-class、v-for、v-if、v-show、v-on）。vue的model层的数据属性。绑定事件：

```
<input @click=doLog() />
```

7、axios是什么？怎么使用？描述使用它实现登录功能的流程？

答：请求后台资源的模块。npm install axios -S装好，然后发送的是跨域，需在配置文件中config/index.js进行设置。后台如果是Tp5则定义一个资源路由。js中使用import进来，然后.get或.post。返回在.then函数中如果成功，失败则是在.catch函数中

Vue.js 1.0 我们常使用 vue-resource (官方ajax库)，Vue 2.0 发布后作者宣告不再对 vue-resource 进行更新，推荐我们使用 axios (基于 Promise 的 HTTP 请求客户端，可同时在浏览器和 node.js 中使用)

8、axios+tp5进阶中，调用axios.post('api/user')是进行的什么操作？axios.put('api/user/8')呢？

答：跨域，添加用户操作，更新操作。

9、什么是RESTful API？怎么使用？

答：是一个api的标准，无状态请求。请求的路由地址是固定的，如果是tp5则先路由配置中把资源路由配置好。标准有：.post .put .delete

10、vuex是什么？怎么使用？哪种功能场景使用它？

答：vue框架中状态管理。在main.js引入store，注入。新建了一个目录store，.... export 。场景有：单页应用中，组件之间的状态。音乐播放、登录状态、加入购物车

11、mvvm框架是什么？它和其它框架（jquery）的区别是什么？哪些场景适合？

答：一个model+view+viewModel框架，数据模型model，viewModel连接两个

区别：vue数据驱动，通过数据来显示视图层而不是节点操作。

场景：数据操作比较多的场景，更加便捷

12、自定义指令（v-check、v-focus）的方法有哪些？它有哪些钩子函数？还有哪些钩子函数参数？

答：全局定义指令：在vue对象的directive方法里面有两个参数，一个是指令名称，另外一个函数。组件内定义指令：directives

钩子函数：bind（绑定事件触发）、inserted（节点插入的时候触发）、update（组件内相关更新）

钩子函数参数：el、binding

14、vue-router是什么？它有哪些组件？

答：vue用来写路由一个插件。router-link、router-view

15、导航钩子有哪些？它们有哪些参数？

答：导航钩子有：a/全局钩子和组件内独享的钩子。b/beforeRouteEnter、afterEnter、beforeRouterUpdate、beforeRouteLeave

参数：有to（去的那个路由）、from（离开的路由）、next（一定要用这个函数才能去下一个路由，如果不用就拦截）最常用就这几种

16、Vue的双向数据绑定原理是什么？

答：vue.js 是采用数据劫持结合发布者-订阅者模式的方式，通过Object.defineProperty()来劫持各个属性的setter，getter，在数据变动时发布消息给订阅者，触发相应的监听回调。

具体步骤：

第一步：需要observe的数据对象进行递归遍历，包括子属性对象的属性，都加上 setter和getter这样的话，给这个对象的某个值赋值，就会触发setter，那么就能监听到了数据变化

第二步：compile解析模板指令，将模板中的变量替换成数据，然后初始化渲染页面视图，并将每个指令对应的节点绑定更新函数，添加监听数据的订阅者，一旦数据有变动，收到通知，更新视图

第三步：Watcher订阅者是Observer和Compile之间通信的桥梁，主要做的事情是：

- 1、在自身实例化时往属性订阅器(dep)里面添加自己
- 2、自身必须有一个update()方法
- 3、待属性变动dep.notice()通知时，能调用自身的update()方法，并触发Compile中绑定的回调，则功成身退。

第四步：MVVM作为数据绑定的入口，整合Observer、Compile和Watcher三者，通过Observer来监听自己的model数据变化，通过Compile来解析编译模板指令，最终利用Watcher搭起Observer和Compile之间的通信桥梁，达到数据变化->视图更新；视图交互变化(input) -> 数据model变更的双向绑定效果。

ps：16题答案同样适合“vue data是怎么实现的？”此面试题。

18、请说下封装 vue 组件的过程？

答：首先，组件可以提升整个项目的开发效率。能够把页面抽象成多个相对独立的模块，解决了我们传统项目开发：效率低、难维护、复用性等问题。

然后，使用Vue.extend方法创建一个组件，然后使用Vue.component方法注册组件。子组件需要数据，可以在props中接受定义。而子组件修改好数据后，想把数据传递给父组件。可以采用emit方法。

19、你是怎么认识vuex的？

答：vuex可以理解成一种开发模式或框架。比如PHP有thinkphp，java有spring等。通过状态（数据源）集中管理驱动组件的变化（好比spring的IOC容器对bean进行集中管理）。

应用级的状态集中放在store中；改变状态的方式是提交mutations，这是个同步的事物；异步逻辑应该封装在action中。

21、请说出vue.cli项目中src目录每个文件夹和文件的用法？

答：assets文件夹是放静态资源；components是放组件；router是定义路由相关的配置；view视图；app.vue是一个应用主组件；main.js是入口文件

22、vue.cli中怎样使用自定义的组件？有遇到过哪些问题吗？

答：第一步：在components目录新建你的组件文件（smithButton.vue），script一定要export default {

第二步：在需要用的页面（组件）中导入：import smithButton from ‘../components/smithButton.vue’

第三步：注入到vue的子组件的components属性上面，components:{smithButton}

第四步：在template视图view中使用，<smith-button> </smith-button>

问题有：smithButton命名，使用的时候则smith-button。

23、聊聊你对Vue.js的template编译的理解？ <https://www.jianshu.com/p/e1669afa30b8>

答：简而言之，就是先转化成AST树，再得到的render函数返回VNode（Vue的虚拟DOM节点）

详情步骤：

首先，通过compile编译器把template编译成AST语法树（abstract syntax tree 抽象语法树 即 源代码的抽象语法结构的树状表现形式），compile是createCompiler的返回值，createCompiler是用以创建编译器的。另外compile还负责合并option。

然后，AST会经过generate（将AST语法树转化成render function字符串的过程）得到render函数，render的返回值是VNode，VNode是Vue的虚拟DOM节点，里面有（标签名、子节点、文本等等）

挑战一下：

1、vue响应式原理？<https://blog.csdn.net/GitChat/article/details/78516752> 看这篇
<https://blog.csdn.net/aaa333qwe/article/details/80093810>

有一个数据a.b,在vue对象实例化过程中，会给a,b通过ES5的defineProperty()方法，添加getter和setter方法，同时vue.js会对模板做编译，解析生成一个指令对象，比如v-text指令，每个指令对象都会关联一个watcher，当对指令对象求值时，就会触发getter，并将依赖收集到watcher中；当再次改变a.b值时，就会触发setter方法，会通知到对应关联的watcher,watcher则再次对a.b求值，计算对比新旧值，当值改变时，watcher会通知到指令，调用指令的update方法，由于指令是对dom的封装，所以会调用原生dom的方法，去更新视图。

2、vue-router实现原理？

<https://www.cnblogs.com/yanze/p/7644631.html> <https://segmentfault.com/a/1190000015123061>

3、为什么要选vue？与其它框架对比的优势和劣势？

(1)vue.js更轻量，gzip后大小只有20K+,React gzip后大小为44k,Angular gzip后大小有56k,所以对于移动端来说，vue.js更适合；

(2)vue.js**更易上手，学习曲线平稳，而Angular入门较难，概念较多(比如依赖注入)，它使用java写的**，很多思想沿用了后台的技术，react需学习较多东西，附带react全家桶，

(3)吸收两家之长，借用了angular的指令(比如v-show,v-hide对应angular的ng-show,ng-hide)和react的组件化(将一个页面抽成一个组件，组件具有完整的生命周期)

(4)vue还有自己的特点，比如计算属性

4、vue如何实现父子组件通信，以及非父子组件通信？

<http://www.cnblogs.com/sichaoyun/p/6690322.html#4021894>

- 5、vuejs与angularjs以及react的区别？
- 6、vuex是用来做什么的？
- 7、vue源码结构

组件的设计原则

- (1)页面上每个独立的可视/可交互区域视为一个组件(比如页面的头部，尾部，可复用的区块)
- (2)每个组件对应一个工程目录，组件所需要的各种资源在这个目录下就近维护(组件的就近维护思想体现了前端的工程化思想，为前端开发提供了很好的分治策略，在vue.js中，通过.vue文件将组件依赖的模板，js，样式写在一个文件中)
- (每个开发者清楚开发维护的功能单元，它的代码必然存在在对应的组件目录中，在该目录下，可以找到功能单元所有的内部逻辑)
- (3)页面不过是组件的容器，组件可以嵌套自由组合成完整的页面

Virtual DOM算法，简单总结下包括几个步骤：<https://www.cnblogs.com/aaronjs/p/7274965.html>

用JS对象描述出DOM树的结构，然后在初始化构建中，用这个描述树去构建真正的DOM，并实际展现到页面中
当有数据状态变更时，重新构建一个新的JS的DOM树，通过新旧对比DOM数的变化diff，并记录两棵树差异
把步骤2中对应的差异通过步骤1重新构建真正的DOM，并重新渲染到页面中，这样整个虚拟DOM的操作就完成了，视图也就更新了

