



第 1 章 Vue 基础入门

一、填空题

1. Vue 是一套构建 用户界面 的渐进式框架。
2. MVVM 主要包含 3 个部分, 分别是 Model、View 和 View-Module。
3. Vue 中通过 refs 属性获取相应 DOM 元素。
4. 在进行 Vue 调试时, 通常使用 Vue-devtools 工具来完成项目开发。
5. Vue 中页面结构以 组件 形式存在。

二、判断题

1. Vue 与 Angular 和 React 框架不同的是, Vue 设计为自下而上逐层应用。(✓)
2. Vue 完全能够为复杂的单页应用提供驱动。(✓)
3. Vue 是一套构建用户界面的渐进式框架, Vue 的核心只关注视图层。(✓)
4. Vue 中 MVVM 框架主要包含 3 部分组成: Model、View 和 ViewModule。(✓)
5. Vue 可以在 Node 环境下进行开发, 并借助 npm 包管理器来安装依赖。(✓)

三、选择题

1. 下列关于 Vue 说法错误的是 (A)。
A. Vue 与 Angular 都可以用来创建复杂的前端项目
B. Vue 优势主要包括轻量级、双向数据绑定
C. Vue 在进行实例化之前, 应确保已经引入了核心文件 vue.js
D. Vue 与 React 语法是完全相同的
2. 下列关于 Vue 的优势的说法错误的是 (C)。
A. 双向数据绑定
B. 轻量级框架
C. 增加代码的耦合度
D. 实现组件化
3. 下列不属于 Vue 开发所需工具的是 (D)。
A. Chrome 浏览器
B. VS Code 编辑器
C. vue-devtools
D. 微信开发者工具
4. npm 包管理器是基于 (A) 平台使用的。
A. Node.js
B. Vue



C.Dabel

D.Angular

5. 下列选项中, 用来安装 vue 模块的正确命令是 (A)。

A.npm install vue

B.node.js install vue

C.node install vue

D.npm l vue

四、简答题

1. 请简述什么是 Vue。

② 轻量级: Vue 相对简单, 直接引入一个现有的项目中。

所以 Vue 使用起来更友好

③ 数据绑定: Vue 是一个 MVVM 框架, 即双向数据绑定。

④ 指令: 指令主要包括内置指令和自定义指令, 以 "v-" 开头, 作用于 HTML 元素。

2. 请简述 Vue 优势有哪些。

3. 请简单介绍 Vue、React 之间的区别。

React 是一个适合数据经常变化以及构建大型项目的复杂组件。

Vue 是一个更小更灵活, 适合用来开发单页面程序的简单化组件

第 2 章 Vue 开发基础 (上)

一、填空题

1. Vue 实例对象通过 new 方式来创建。2. Vue 初始数据在实例对象的 data 参数中进行定义。3. Vue 实例对象中 el 参数表示 唯一根标签。4. Vue 中实现双向数据绑定的指令是 v-model5. Vue 事件绑定指令是 v-on

二、判断题

1. 在项目中引入了 vue.js 文件, 才可以创建 Vue 实例。(✓)

2. Vue 实例对象指令主要包括自定义指令和内置指令, 通过指令省去 DOM 操作。(✓)

3. 事件传递方式有冒泡和捕获, 默认是冒泡。(✓)

4. Vue 开发提出了组件化开发思想, 每个组件都是一个孤立的单元。(✓)

5. 在 Vue 中 beforeDestroy 与 destroyed 生命周期函数执行后, 都可以获取到 Vue 实例。(✓)

三、选择题

6. 下列关于 Vue 实例对象说法不正确的是 (D)。



- A. Vue 实例对象是通过 `new Vue({})` 方式创建的
 B. Vue 实例对象只允许有唯一的一个根标签
 C. 通过 `methods` 参数可以定义事件处理函数
 D. Vue 实例对象中 `data` 数据不具有响应特性
7. Vue 实例对象中能够监听状态变化的参数是 (A).
 A. `watch`
 B. `filters`
 C. `watching`
 D. `components`
8. Vue 中实现数据双向绑定的是 (C).
 A. `v-bind`
 B. `v-for`
 C. `v-model`
 D. `v-if`
9. 在 Vue 中, 能够实现页面单击事件绑定的代码是 (B).
 A. `v-on:enter`
 B. `v-on:click`
 C. `v-on:mouseenter`
 D. `v-on:doubleclick`
10. 下面列出的钩子函数会在 Vue 实例销毁完成时执行的是 (B).
 A. `updated`
 B. `destroyed`
 C. `created`
 D. `mounted`
1. 每个 Vue 应用都是通过 Vue 函数创建一个新的 Vue 实例开始, 使用 `new` 关键字可以实例化 Vue 对象.
 2. 在 Vue 中, 组件是构成页面中独立结构单元, 能够减少重复代码的编写.

四、简答题

1. 请简述什么是 Vue 实例对象。
 2. 请简述什么是 Vue 组件化开发。
 3. 请简单介绍 Vue 数据绑定内置指令主要内容有哪些。
3. `v-on`: 监听事件.
`v-bind`: 单向数据绑定.
`v-text`: 插入文本内容.
`v-html`: 插入包含 HTML 的内容.
- `v-for`: 列表渲染
`v-if`: 条件渲染
`v-show`: 显示隐藏
- 第 3 章 Vue 开发基础 (下) 项目功能.
 同组件也具有基本交互功能, 根据业务逻辑实现复杂的项目功能.

一、填空题

1. Vue 实例对象通过 `vm.$root` 方式来获取。
 2. Vue 初始数据通过 `vm.$data` 方式获取。



3. Vue 中通过 vm.\$children 获取当前实例的子组件。 (~~X~~)

4. Vue 中创建插件提供的方法是 install

5. Vue 中通过 Vue.directive() 创建自定义指令。

二、判断题

1. Vue 提供的全局 API 接口 component(), 不能用来注册组件。 (~~X~~)

2. Vue 中 Vue.config 对象用来实现 Vue 全局配置。 (~~Y~~)

3. Vue 中 data 选项中的数据具有响应特性。 (~~Y~~)

4. Vue 中, 可以通过 vm.\$slots 获取子组件实例对象。 (~~X~~)

5. Vue 实例对象中通过 \$options 可以获取到父作用域下的所有属性。 (~~X~~)

三、选择题

1. 下列关于 Vue 实例对象接口说法, 错误的是 (~~D~~)。

A. Vue 实例对象提供了实例可操作方法

B. Vue 实例对象 \$data 数据可以由实例 vm 委托代理

C. 通过 Vue 实例对象可以进行 Vue 全局配置

D. Vue 实例对象接口同样可以通过 Vue 方式调用

2. 下面关于 Vue 全局配置说法, 错误的是 (~~C~~)。

A. Vue.config.devtools 可以设置 devtools 调试工具的启用和关闭

B. Vue.config 是一个对象, 包含 Vue 的全局配置

C. Vue.component() 可以获取 Vue 全局配置对象

D. Vue.set.config 可以获取到全局配置对象

3. 下列不属于 Vue 实例对象属性的是 (~~B~~)。

A. \$data

B. \$component

C. \$props

D. \$root

4. Vue 实例对象获取子组件实例对象的方式是 (~~B~~)。

A. \$parent

B. \$children

C. \$child

D. \$component

5. 下面对于 Vue.mixin 的说法, 错误的是 (~~D~~)。

A. Vue.mixin 是 Vue 提供的全局接口 API

B. Vue.mixin 可以用来注入组件选项

C. 使用 Vue.mixin 可能会影响到所有 Vue 实例

D. Vue.mixin 不可以用来注入自定义选项的处理逻辑



四、简答题

1. 请简述什么是 Vue 插件。
1. Vue 插件就是一个对象或函数，需要有 install 方法，可通过 `Vue.use` 安装。
2. 请简述 Vue 全局 API 接口主要内容。
2. `Vue.directive` 自定义内置指令，例如 `focus`。
3. 请简单介绍 Vue 实例对象属性和方法。

3. `vm.$props`: 接收上级组件向下级传递的数据。
② `Vue.use`: 安装插件。
③ `Vue.extend`: 为 `Vue` 构造器创建一个子类。

④ `vm.$options`: `Vue` 实例初始化时传的自定义选项。
④ `Vue.set`: 向响应式对象中添加一个属性，并确保其也是响应式的。

⑤ `vm.$el`: 访问根 DOM 元素。
⑤ `Vue.mixin`: 混入，影响全局 `Vue` 实例。
⑤ `vm.$children`: 访问当前实例的直接子组件。
1. `Vue` 提供了内置的过渡封装组件是 `transition`。

⑤ `vm.$root`: 组件树的根 `Vue` 实例。
2. 在过渡封装组件中使用 `name` 属性可以重置过渡中切换类名的前缀。

⑤ `vm.$slot`: 查看插槽内容。
3. 通过 `appear` 特性设置节点在初始渲染的过渡。
4. 在离开的过渡中有 `v-leave` `v-leave-active` `v-leave-to` 3 个 class 切换。

⑤ `vm.$slot`: 自定义过渡。
5. 自定义过渡的类名优先级要高于普通的类名。

判断题

1. 函数式组件中的 `render()` 函数用来创建组件模板。(✓)
2. 给过渡元素添加 `v-bind:css="true"`, `Vue` 会跳过 `CSS` 的检测。(✗)
3. 在 `@before-enter` 阶段可以设置元素开始动画之前的起始样式。(✓)
4. 在使用 `animate.css` 库时，基本的 `class` 样式名是 `animate`。(✗)
5. `enter` 和 `leave` 动画钩子函数，除 `el` 参数外还会传入一个 `done` 作为参数。(✓)

三、选择题

1. 下列选项中关于动画钩子函数说法正确的是 (C)
 - A. `@leave-cancelled` 函数只能用于 `v-if` 中
 - B. 对于 `@enter` 来说，当与 `CSS` 结合使用时，回调函数 `done` 是必选的
 - C. `done` 作为参数，作用就是告知 `Vue` 动画结束
 - D. 钩子函数需要结合 `CSS transitions` 或 `animations` 使用，不能单独使用
2. 下列关于 `Vue` 为 `<transition>` 标签提供的过渡类名的说法，错误的是 (D)
 - A. `v-enter` 在元素被插入之前生效，在元素被插入之后的下一帧移除
 - B. `v-leave` 在离开过渡被触发时立刻生效，下一帧被移除
 - C. `v-enter-active` 可以控制进入过渡的不同的缓和曲线
 - D. 如果 `name` 属性为 `my-name`，那么 `my-` 就是在过渡中切换的类名的前缀
3. 下列选项中关于多个元素过渡的说法，错误的是 (D)
 - A. 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

第 4 章 Vue 过渡和动画



- B. 不相同元素之间可以使用 v-if 和 v-else 来进行过渡
 C. <transition>组件的默认行为指定进入和离开同时发生
 D. 不可以给同一个元素的 key 特性设置不同的状态来代替 v-if 和 v-else

四、简答题

1. 11个 *before Enter, Enter, before Leave, leave, after Enter, after Leave, enterCancelled, LeaveCancelled.*
2. 请简述 JavaScript 钩子函数包括哪些。
 3. 请简述 6 个内置的过渡类名。
 3. 请简述自定义过渡类名的属性有哪些。
2. V-enter: 元素插入前
 V-enter-active: 插入过程中
 V-enter-to: 元素插入后, 与此同时 V-enter 被移除
 V-leave: 离开前
 V-leave-active: 离开中
 V-leave-to: 离开后
3. 6个 *enter-class, enter-active-class, enter-to-class, leave-class, leave-active-class, leave-to-class*
- 第 5 章 Vue 路由
1. 在项目中可以通过 npm 命令 *npm install vue-router* 安装路由 *vue-router 3.1.x*。
 2. 使用 *this.\$route* 路由对象获取当前激活的路由的状态信息。
 3. 通过一个名称来标识一个路由的方式叫作 *命名路由*。
 4. 在业务逻辑代码中实现导航跳转的方式称为 *编程式导航*。
 5. 单页面应用主要通过 URL 中的 *hash#* 实现不同页面之间的切换。

二、判断题

1. 后端路由通过用户请求的 URL 导航到具体的 html 页面。(✓)
 2. 开发环境下, 使用 `import VueRouter from 'vueRouter'` 来导入路由。(✗)
 3. 嵌套路由的使用, 主要是由页面结构来决定的。(✓)
 4. params 方式传参类似于 GET 请求。(✗)
 5. 在单页面应用中更新视图可以不用重新请求页面。(✓)

三、选择题

1. 下列 vue-router 插件的安装命令, 正确的是 A。
 A. `npm install vue-router`
 B. `node install vue-router`
 C. `npm Install vueRouter`
 D. `npm I vue-router`
2. 下列关于 query 方式传参的说法, 正确的是 A。
 A. query 方式传递的参数会在地址栏展示
 B. 在页面跳转的时候, 不能在地址栏看到请求参数
 C. 在目标页面中使用 `"this.route.query.参数名"` 来获取参数
 D. 在目标页面中使用 `"this.$route.params.参数名"` 来获取参数
3. 下列关于 params 方式传参的说法, 错误的是 ()。C

1. ① 在指定目录下 `npm install vue-router @ 3.1.x`.
- ② 创建 `main.js` 逻辑入口, 引入 `vue-router`.



- A. 在目标页面中也可以使用 "`$route.params`, 参数名" 来获取参数
- B. 在页面跳转的时候, 不能在地址栏看到请求参数
- C. `params` 方式传递的参数会在地址栏展示
- D. 在目标页面中使用 "`this.$route.params`, 参数名" 来获取参数

四、简答题 3. ~~this~~

2. 前端路由用于切换页面, 却不像后端路由那样每次切换页面都要请求服务器。响应快, 提升用户体验。
1. 请简述 `npm` 方式安装 `vue-router` 的步骤。
2. 请简述 `vue-router` 路由的作用。
3. 请简单列举并说明路由对象包括哪些属性。见背面。

第 6 章 Vuex 状态管理

一、填空题

1. Vuex 实例对象通过 `vm.$store` 方式来获取。
2. Vuex 实例对象中初始数据状态通过 `vm.$store.state` 方式获取。
3. Vuex 实例对象中组件状态通过 `vm.$store.commit()` 方式改变。
4. Vuex 中创建动态模块提供的方法是 `store.registerModule()`
5. Vuex 中通过 `vm.$store.dispatch()` 实现 actions 状态分发。

二、判断题

1. Vuex 实例对象可以调用 Vue 全局接口。(X)
2. Vuex 中 `Vue.config` 对象用来实现 Vuex 全局配置。(X)
3. Vuex 中 `state` 选项中数据是初始数据状态。(✓)
4. Vuex 中插槽可以实现组件任意嵌套, 且在版本 2.2.6+ 以后开始支持。(X)
5. 当在 Vuex 实例对象中调用 `store` 时, 一定会获取到 `store` 实例对象。(✓)

三、选择题

1. 下列关于 Vuex 实例对象接口的说法, 错误的是 (B)。
 A. Vuex 实例对象提供了 `store` 实例对象可操作方法
 B. Vuex 实例对象 `$data` 数据可以由实例委托代理
 C. 通过 Vuex 实例对象实现组件状态的管理维护
 D. Vuex 实例对象初始数据是 `state` 数据
2. 下面关于 Vuex 核心模块说法, 错误的是 (B)。
 A. Vuex 配置对象中, `actions` 选项是异步的
 B. Vuex.config 对象是全局配置对象
 C. Vuex 配置对象中, `mutations` 选项是同步的
 D. 通过 `commit` 完成 `mutations` 提交

1. Vuex是一个专门为Vue.js应用程序开发的状态管理模式,作为Vue插件来用,采用集中式存储管理应用的所有组件的状态,使Vue组件状态更加容易维护。

河南大学软件学院

Web 开发技术

3. 下列不属于 Vuex.Store 配置对象接收参数的是 (A)。
A.data B.state C.mutations D.getters
4. Vuex 实例对象中类似于 computed 计算属性功能的选项是 (D)。
A.state B.mutations C.actions D.getters
5. 下面对于 Vuex 中 actions 说法, 不正确的是 (C)。
A.actions 中事件函数通过 commit 完成分发
B.acitons 中事件处理函数接收 context 对象
C.actions 与 Vue 实例中的 methods 是类似的
D.可以用来注入自定义选项的处理逻辑

四、简答题

1. 请简要分析 Vuex 的设计思想。
2. 简述 Vuex 配置对象中的主要内容有哪些。见背面
3. 简述 Vuex 中的 actions 的含义。见背面。

第 7 章 Vue 开发环境

一、填空题

1. 对于 CLI 类型的插件, 需要以 @vue- 为前缀。
2. 使用 npm 包通过 npm install -g @vue/cli 命令全局安装 @vue/cli 3.x。
3. 使用 vue -v 来查看 vue 的版本号。
4. 使用 yarn 包通过 yarn global add @vue/cli 命令全局安装 @vue/cli 3.x。
5. 在 Vue CLI 3 中使用 vue create 项目名 命令来创建一个 Vue 项目。

二、判断题

1. 卸载 vue-cli 的命令是 npm uninstall vue-cli -g。 () ✓
2. 添加 CLI 插件的命令是 vue add vue-eslint。 () ✗
3. 插件不能修改 webpack 的内部配置, 但是可以向 vue-cli-service 注入命令。 (X)
4. Vue CLI 通过 vue ui 命令来创建图形用户界面。 () ✓
5. 在文件中用 "key=value" (键值对) 的方式来设置环境变量。 () ✓

三、选择题

1. 下列选项中说法正确的是 ()。 B
A. 新版的 Vue CLI 的包名称为 vue-cli
B. 执行 npm uninstall vue-cli -g 命令可以全局删除 vue-cli 包
C. 使用 yarn install add @vue/cli 命令可以全局安装 @vue/cli 工具



- D. 通过 `vue add ui` 命令来创建图形用户界面
2. 关于 CLI 服务, 下列选项说法错误的是 ()。 **C**
- A. 在 `@vue/cli-service` 中安装了一个名为 `vue-cli-service` 的命令
- B. `vue.config.js` 是一个可选的配置文件
- C. 通过 `npm vue-cli-service helps` 查看所有的可用命令
- D. 通过 `vue ui` 使用 GUI 图形用户界面来运行更多的特性脚本
3. 下列选项中说法正确的是 ()。 **AD**
- A. 使用相对路径引入的静态资源文件, 会被 `webpack` 处理解析为模块依赖
- B. 放在 `public` 文件夹下的资源将会经过 `webpack` 的处理
- C. 通过绝对路径被引用的资源将会经过 `webpack` 的处理
- D. URL 以 `~` 开始, 会被认为是模块请求

四、简答题

1. 简述如何安装 Vue CLI 3.x 版本的脚手架。 **`npm install @vue/cli -g`**
2. 简述如何在现有项目中安装 CLI 插件和第三方插件。
- 简单介绍 CLI 服务 `vue-cli-service <command>` 中的 `command` 命令包括哪些。
- `vue add @vue/eslint`**
- `vue add vuetify`** 不带 `@vue` 前缀, 将会安装第三方包

第 8 章 服务器端渲染

`serve`: 启动服务

`build`: 生成用于生产环境的包。

`inspect`: 审查 `webpack` 配置

`lint`: 保护并修复源文件

一、填空题

1. _____ 插件可以用来进行页面的热重载。 **`webpack-hot-middleware`**
2. `hash` 模式路由, 地址栏 URL 中会自带 **`#`** 符号。
3. SSR 的路由需要采用 **`history`** 的方式。
4. **`SEO`** 是利用搜索引擎规则, 提高网站在搜索引擎内自然排名的一种技术。
5. Vue 中使用服务器端渲染, 需要借助 Vue 的扩展工具 **`vue-server-renderer`**。

二、判断题

1. 客户端渲染, 即传统的单页面应用模式。 **(✓)**
2. `webpack-dev-middleware` 中间件对更改的文件进行监控。 **(✓)**
3. 服务端渲染 **不利于** SEO。 **(✗)**
4. 服务器端渲染应用程序, 需要处于 Node.js server 运行环境。 **(✓)**
5. 使用 `git-bash` 命令行工具, 输入指令 **`mkdirs vue-ssr`** 创建项目。 **(✗)**

三、选择题

1. 下列选项中说法正确的是 ()。 **A**
- A. `vue-server-renderer` 的版本要与 Vue 版本相匹配

`vue create`



- B. 客户端渲染, 需要使用 entry-server.js 和 entry-client.js 两个入口文件
 C. app.js 是应用程序的入口, 它对应 vue-cli 创建的项目的 app.js 文件
 D. 客户端应用程序既可以运行在浏览器上, 又可以运行在服务器上
2. 下列关于 SSR 路由说法, 错误的是 (B)
- A. SSR 的路由需要采用 history 的方式
 B. history 模式的路由提交不到服务器上
 C. history 模式完成 URL 跳转而无须重新加载页面
 D. hash 模式路由, 地址栏 URL 中 hash 改变不会重新加载页面
3. 下列关于 Nuxt.js 的说法, 错误的是 (D)
- A. 使用 "create-nuxt-app 项目名" 命令创建项目
 B. 使用 Nuxt.js 搭建的项目中, pages 目录是用来存放应用的路由及视图
 C. 在 Nuxt.js 项目中, 声明式路由在 html 标签中通过 <nuxt-link> 完成路由跳转
 D. Nuxt.js 项目中需要根据目录结构手动完成对应的路由配置

四、简答题

1. 请简述什么是服务器端渲染。见背面
2. 请简述服务器端渲染的代码逻辑和处理步骤。见背面
3. 请简述 Nuxt.js 中, 声明式路由和编程式路由的区别。
- 声明式路由是在页面中使用 <nuxt-link> 完成路由跳转
 编程式路由是在 JavaScript 代码中实现路由的跳转

第 9 章 “微商城” 项目

一、填空题

1. 使用 Mint UI 库的页面, 需要通过 mt- 前缀来定义标签名。
2. Axios 是一个基于 Promise 的 HTTP 库, 可以用在浏览器和 node.js 中。
3. MUI 是最接近原生 APP 体验的高性能前端框架。
4. 使用 Object.defineProperty, 给 Vue 函数添加一个原型属性 \$http, 指向 axios。
5. 使用路由的声明式导航, 在 html 标签中使用 router-link 组件来实现路由的跳转

二、判断题

1. MUI 是一套代码片段, 提供了配套的样式和 HTML 代码段。X
2. 使用 lazy-load 可以实现图片懒加载。() ✓
3. 通过 this.\$store.state.* 可以访问 state 中的数据。() ✓
4. 组件想要修改数据, 需要调用 mutations 提供的方法, 通过 this.\$store.emit(方法名) 实现。() X
5. better-scroll 是一款支持各种滚动场景需求的插件。() ✓



一、选择题

1. 下列选项中, A 指令可用来切换元素的可见状态。
A. v-show B. v-hide C. v-toggle D. v-slideHide
2. 下列关于 ref 的作用说法, 错误的是 (), D
A. ref 在子组件中使用, 使用 this.\$refs.name 获取到组件实例, 可以使用组件的所有属性和方法
B. ref 加在普通的元素上, 用 this.\$refs.name 获取到的是 DOM 元素
C. 可以利用 v-for 和 ref 获取一组数组或者 DOM 节点
D. ref 在 DOM 渲染完成之前就能使用
3. 想要获取购物车小球在页面上的位置, 以下可以使用的是 (), B
A. offset() B. getBoundingClientRect() C. width() D. height()

四、简答题

1. 请简单列举一个项目从开始到上线的开发流程需要的步骤。
2. 请简单列举 6 个“微商城”项目中用到的重点知识。

1.
 - ① 产品创意: 结合公司发展方向及战略, 提出产品创意
 - ② 产品原型: 包括功能、页面, 最重要的是用户体验。
 - ③ 美工设计: 美工设计人员根据产品经理提供的原型图实现符合原型与审美的 psd 设计图稿, 并进行切图。
 - ④ 前端实现: 前端开发工程师拿到美工设计的 psd 图, 负责具体的 HTML, CSS 静态页面的实现, 以及动态特效, 动态数据的绑定和交互的实现。
 - ⑤ 后端实现: 实现数据处理, 业务逻辑代码。
 - ⑥ 测试、试运行、上线: 由测试人员进行项目测试, 将所有问题解决后, 就可试运行, 将项目上线。

2.
 - ① 路由的配置
 - ② Vuex 的配置
 - ③ axios 的封装及接口的调用
 - ④ 底部导航栏的实现
 - ⑤ 购物车功能的实现
 - ⑥ 分类列表的实现。

第八章

1. 服务器渲染 (SSR), 是将组件或页面通过服务器生成 HTML 字符串, 再发送至服务器浏览器, 最后将静态标记 "混合" 为客户端上完全交互的应用程序, 简单理解就是将页面在服务器中完成渲染, 然后在客户端直接展示。

2. Vue 进行服务器渲染时, 需要利用 Node.js 搭建一个服务器, 并添加服务器渲染的代码逻辑。

使用 `webpack-dev-middleware` 中间件对更改的文件进行监控,

使用 `webpack-hot-middleware` 中间件进行页面的热重载,

使用 `vue-server-renderer` 插件来渲染服务器端打包的 `bundle` 文件到客户端

第八章.

- 1. actions: 用来定义事件处理方法, 用于处理 state 数据, 异步.
- mutations: 其事件处理方法接收 state 对象作为参数, 即初始数据.
- getters: 计算属性, 类似于 Vue 实例的 computed.
- modules: 用来在 store 实例中定义模块对象.
- plugins: Vuex 中的插件 配置选项为 plugins, 插件本身为函数.
- devtools: Store 实例配置中的 devtools 选项用来设置是否在 devtools 调试工具中启用 Vuex, 默认值为 true, 表示启用.

3. actions 选项用来定义事件处理方法, 用于处理 state 数据.

actions 类似于 mutations, 不同之处在于 actions 是异步执行的. 事件处理函数可以接收 `commit` 对象, 能简化代码, 直接完成 mutation 提交, 从而方便 devtools 调试工具跟踪状态的 state 变化.

在使用时, 需要在 store 仓库中注册 actions 选项, 在里面定义事件处理方法. 事件处理方法接收 context 作为第 1 个参数, payload 作为第 2 个参数 (根据需要进行选择).

第5章

3. `this.$router` , 全局路由对象
`this.$route` , 当前路由对象.

`$route.path`.

`$route.query` : {key: value} URL查询参数

`$route.params` : {key: value} 路由跳转时携带参数

`$route.hash` : 在 history 模式下获取当前路由 hash 值.

`$route.fullPath` : 完成解析后的 URL, 包含查询参数和 hash 的完整路径

`$route.name`.

`$route.matched` : 路由记录, 当前路由下路由声明的所有信息, 从父路由(若有)到当前路由为止.

`$route.redirectedFrom` : 若有重定向, 即为重定向来源的路由.