Vue.js 响应接口

Vue.js 实例

本章节为大家介绍几个 Vue.js 实例，通过实例练习来巩固学到的知识点。

导航菜单实例

导航菜单

创建一个简单的导航菜单：

```
<div id="main">
<!-- 激活的菜单样式为 active 类 -->
<!-- 为了阻止链接在点击时跳转，我们使用了 "prevent" 修饰符 (preventDefault 的简称)。 -->
<nav v-bind:class="active" v-on:click.prevent>
<!-- 当菜单上的链接被点击时，我们调用了 makeActive 方法，该方法在 Vue 实例中创建。 -->
<a href="#" class="home" v-on:click="makeActive('home')">Home</a>
<a href="#" class="projects" v-on:click="makeActive('projects')">Projects</a>
<a href="#" class="services" v-on:click="makeActive('services')">Services</a>
<a href="#" class="contact" v-on:click="makeActive('contact')">Contact</a>
</nav>
<!-- 以下 "active" 变量会根据当前选中的值来自动变换 -->
<p>您选择了 <b>{{active}} 菜单</b></p>
</div>
<script>
// 创建一个新的 Vue 实例
var demo = new Vue({
// DOM 元素，挂载视图模型
el: '#main',
// 定义属性，并设置初始值
data: {
active: 'home'
},
// 点击菜单使用的函数
methods: {
makeActive: function(item){
// 模型改变，视图会自动更新
this.active = item;
}
}
});
</script>
```

尝试一下 »

编辑文本实例

文本编辑

点击指定文本编辑内容：

```
<!-- v-cloak 隐藏未编译的变量，直到 Vue 实例准备就绪。 -->
<!-- 元素点击后 hideTooltp() 方法被调用 -->
<div id="main" v-cloak v-on:click="hideTooltip">
<!-- 这是一个提示框
v-on:click.stop 是一个点击事件处理器，stop 修饰符用于阻止事件传递
v-if 用来判断 show_tooltip 为 true 时才显示 -->
<div class="tooltip" v-on:click.stop v-if="show_tooltip">
<!-- v-model 绑定了文本域的内容
在文本域内容改变时，对应的变量也会实时改变 -->
<input type="text" v-model="text_content" />
</div>
<!-- 点击后调用 "toggleTooltip" 方法并阻止事件传递 -->
<!-- "text_content" 变量根据文本域内容的变化而变化 -->
<p v-on:click.stop="toggleTooltip">{{text_content}}</p>
</div>
<script>
var demo = new Vue({
el: '#main',
data: {
show_tooltip: false,
text_content: '点我，并编辑内容。'
},
methods: {
hideTooltip: function(){
// 在模型改变时，视图也会自动更新
this.show_tooltip = false;
},
toggleTooltip: function(){
this.show_tooltip = !this.show_tooltip;
}
}
})
</script>
```

尝试一下 »

订单列表实例

订单列表

创建一个订单列表，并计算总价：

```
<form id="main" v-cloak>
<h1>Services</h1>
<ul>
<!-- 循环输出 services 数组，设置选项点击后的样式 -->
<li v-for="service in services" v-on:click="toggleActive(service)" v-bind:class="{ 'active': se
rvice.active}">
<!-- 显示订单中的服务名，价格
Vue.js 定义了货币过滤器，用于格式化价格 -->
{{service.name}} <span>{{service.price | currency}}</span>
</li>
</ul>
<div class="total">
```

```
<!-- 计算所有服务的价格，并格式化货币 -->
Total: <span>{{total() | currency}}</span>
</div>
</form>
<script>
// 自定义过滤器 "currency".
Vue.filter('currency', function (value) {
return '$' + value.toFixed(2);
});
var demo = new Vue({
el: '#main',
data: {
// 定义模型属性 the model properties. The view will loop
// 视图将循环输出数组的数据
services: [
{
name: 'Web Development',
price: 300,
active:true
},{
name: 'Design',
price: 400,
active:false
},{
name: 'Integration',
price: 250,
active:false
},{
name: 'Training',
price: 220,
active:false
}
]
},
methods: {
toggleActive: function(s){
s.active = !s.active;
},
total: function(){
var total = 0;
this.services.forEach(function(s){
if (s.active){
total+= s.price;
}
});
return total;
}
}
});
</script>
```

尝试一下 »

搜索页面实例

搜索页面

在输入框输入搜索内容，列表显示配置的列表：

```
<form id="main" v-cloak>
<div class="bar">
<!-- searchString 模型与文本域创建绑定 -->
<input type="text" v-model="searchString" placeholder="输入搜索内容" />
</div>
<ul>
<!-- 循环输出数据 -->
<li v-for="article in filteredArticles">
<a v-bind:href="article.url"><img v-bind:src="article.image" /></a>
<p>{{article.title}}</p>
</li>
</ul>
</form>
<script>
var demo = new Vue({
el: '#main',
data: {
searchString: "",
// 数据模型，实际环境你可以根据 Ajax 来获取
articles: [
{
"title": "What You Need To Know About CSS Variables",
"url": "https://www.runoob.com/css/css-tutorial.html",
"image": "https://static.runoob.com/images/icon/css.png"
},
{
"title": "Freebie: 4 Great Looking Pricing Tables",
"url": "https://www.runoob.com/html/html-tutorial.html",
"image": "https://static.runoob.com/images/icon/html.png"
},
{
"title": "20 Interesting JavaScript and CSS Libraries for February 2016",
"url": "https://www.runoob.com/css3/css3-tutorial.html",
"image": "https://static.runoob.com/images/icon/css3.png"
},
{
"title": "Quick Tip: The Easiest Way To Make Responsive Headers",
"url": "https://www.runoob.com/css3/css3-tutorial.html",
"image": "https://static.runoob.com/images/icon/css3.png"
},
{
"title": "Learn SQL In 20 Minutes",
"url": "https://www.runoob.com/sql/sql-tutorial.html",
"image": "https://static.runoob.com/images/icon/sql.png"
},
{
"title": "Creating Your First Desktop App With HTML, JS and Electron",
"url": "https://www.runoob.com/js/js-tutorial.html",
"image": "https://static.runoob.com/images/icon/html.png"
```

```
}
]
},
computed: {
// 计算数学，匹配搜索
filteredArticles: function () {
var articles_array = this.articles,
searchString = this.searchString;
if(!searchString){
return articles_array;
}
searchString = searchString.trim().toLowerCase();
articles_array = articles_array.filter(function(item){
if(item.title.toLowerCase().indexOf(searchString) !== -1){
return item;
}
})
// 返回过来后的数组
return articles_array;;
}
}
});
</script>
```

尝试一下 »

切换不同布局实例

切换不同布局

点击右上角的按钮来切换不同页面布局：

```
<form id="main" v-cloak>
<div class="bar">
<!-- 两个按钮用于切换不同的列表布局 -->
<a class="list-icon" v-bind:class="{ 'active': layout == 'list'}" v-on:click="layout = 'list'">
</a>
<a class="grid-icon" v-bind:class="{ 'active': layout == 'grid'}" v-on:click="layout = 'grid'">
</a>
</div>
<!-- 我们设置了两套布局页面。使用哪套依赖于 "layout" 绑定 -->
<ul v-if="layout == 'grid'" class="grid">
<!-- 使用大图，没有文本 -->
<li v-for="a in articles">
<a v-bind:href="a.url" target="_blank"><img v-bind:src="a.image.large" /></a>
</li>
</ul>
<ul v-if="layout == 'list'" class="list">
<!-- 使用小图及标题 -->
<li v-for="a in articles">
<a v-bind:href="a.url" target="_blank"><img v-bind:src="a.image.small" /></a>
<p>{{a.title}}</p>
</li>
</ul>
```

```
</form>
<script>
var demo = new Vue({
el: '#main',
data: {
// 视图模型，可能的值是 "grid" 或 "list"。
layout: 'grid',
articles: [{
"title": "HTML 教程",
"url": "https://www.runoob.com/html/html-tutorial.html",
"image": {
"large": "https://static.runoob.com/images/mix/htmlbig.png",
"small": "https://static.runoob.com/images/icon/html.png"
}
},
{
"title": "CSS 教程",
"url": "https://www.runoob.com/css/css-tutorial.html",
"image": {
"large": "https://static.runoob.com/images/mix/cssbig.png",
"small": "https://static.runoob.com/images/icon/css.png"
}
},
{
"title": "JS 教程",
"url": "https://www.runoob.com/js/js-tutorial.html",
"image": {
"large": "https://static.runoob.com/images/mix/jsbig.jpeg",
"small": "https://static.runoob.com/images/icon/js.png"
}
},
{
"title": "SQL 教程",
"url": "https://www.runoob.com/sql/sql-tutorial.html",
"image": {
"large": "https://static.runoob.com/images/mix/sqlbig.png",
"small": "https://static.runoob.com/images/icon/sql.png"
}
},
{
"title": "Ajax 教程",
"url": "https://www.runoob.com/ajax/ajax-tutorial.html",
"image": {
"large": "https://static.runoob.com/images/mix/ajaxbig.png",
"small": "https://static.runoob.com/images/icon/ajax.png"
}
},
{
"title": "Python 教程",
"url": "https://www.runoob.com/pyhton/pyhton-tutorial.html",
"image": {
"large": "https://static.runoob.com/images/mix/pythonbig.png",
"small": "https://static.runoob.com/images/icon/python.png"
}
```

```
}]
}
});
</script>
```

尝试一下 »

Vue.js 响应接口

1 篇笔记

写笔记

实现一个脚手架（vue-cli）创建的一个简单 to do list 实例。
完整项目下载：

Download

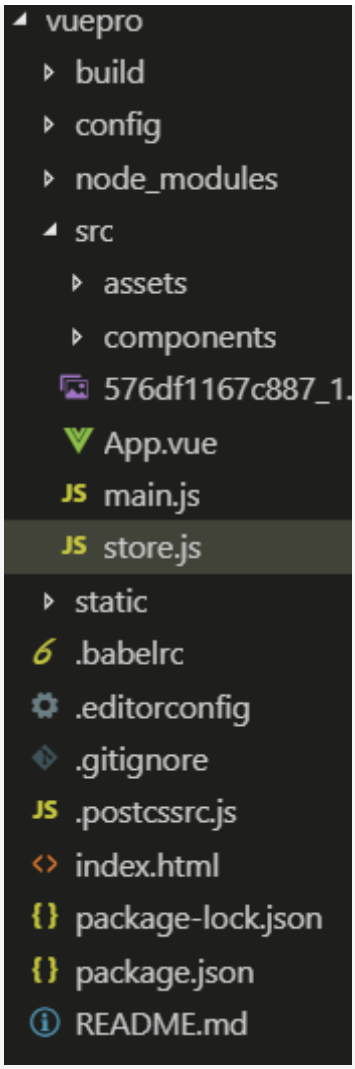
下载后进入项目目录执行 `npm install` 命令导入依赖模块。

执行以下命令启动项目：

```
npm run dev
```

此项目端口号设为 6060。

项目目录结构：



src/App.vue 文件代码：

```
<template>
  <div id="app">
    <h1 v-html = "title"></h1>
    <input v-model="newItem" v-on:keyup.enter="addNew" ></input>
    <ul>
      <li v-for="item in items" v-bind:class="{finished:item.isFinished}" v-on:click="tog
gleFinish(item)">{{item.label}}</li>
    </ul>
  </div>

</template>


<script>
import Store from './store'
export default {
  data:function(){
    return {
      title:"This is a Todolist",
      items:Store.fetch(),
```

```
      newItem:""
    }
  },
  watch:{
    items:{
      handler:function(items){
        Store.save(items)
      },
      deep:true
    }
  },
  methods:{
    toggleFinish:function(item){
      item.isFinished = !item.isFinished
    },
    addNew:function(){
      this.items.push({
        label:this.newItem,
        "isFinished":false
      })
      this.newItem=""
    }
  }
}
</script>

<style>
.finished{
  text-decoration:line-through;
}
li{
  list-style:none;
  font-size:1.6em;
  margin-top:10px;
}

#app {
  background-image:url(./576df1167c887_1024.jpg);
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
input{
  width:230px;
  height:40px;
  border-radius:20px;
```

```
  padding: 0.4em 0.35em;
  border:3px solid #CFCFCF;
  font-size: 1.55em;
}
</style>
```

src/store.js 实现存储，即刷新数据不消失：

```
const STORAGE_KEY='todos-vuejs'
export default {
 fetch:function(){
  return JSON.parse(window.localStorage.getItem(STORAGE_KEY)||'[]');
 },
 save:function(items){
  window.localStorage.setItem(STORAGE_KEY,JSON.stringify(items))
 }
}
```

bm0500 4个月前 (11-02)