

Vue 组件库 - Unit01

1. 组件库基础

1.1 什么是组件库?

组件库是企业为提升开发效率开发的能够完成特定功能的结构(`<template>`)、表现(`<style>`)及行为(`<script>`)的封装; 其可以实现“一次定义, 多次调用”。

1.2 为什么要使用组件库?

- 提升开发效率
- 便于团队协作
- 提升项目的可维护性

1.3 组件库分类

1.3.1 移动端组件库

- Mint UI (饿了么) -- <http://mint-ui.github.io/#!/zh-cn>
- Vant UI (有赞团队) -- <https://youzan.github.io/vant/#/zh-CN/>
- Cube UI (滴滴团队) -- <https://didi.github.io/cube-ui/#/zh-CN>

1.3.2 桌面端组件库

- Element UI (饿了么) -- <https://element.eleme.cn/#/zh-CN>
- AT-UI (凹凸实验室) -- <https://at-ui.github.io/at-ui/#/zh>
- View UI (视图更新) -- <https://www.iviewui.com/>

1.4 移动端与桌面端的区别

- 屏幕尺寸不同(<https://uiuiuiui.com/screen/>)
- 操作方式不同(PC 是通过键盘或鼠标来实现交互; 而移动端则通过手指的动作完成交互)
- 事件触发方式不同(PC 中常用的事件如 `mouseover`、`mousemove` 等, 移动端如 `touchstart`、`touchmove` 等)
- 网络环境不同(PC 一般情况下处于比较稳定网络环境; 而移动端则处于不稳定的网络环境, 如地铁);

2. Mint UI 基础

2.1 概述

Mint UI 是基于 Vue.js 2.0 的移动端的组件库。

2.2 安装

A、启动操作系统的命令行，并且切换到 `package.json` 文件所在的目录

B、输入以下命令，进行 Mint UI 的安装

```
npm install --save mint-ui
```

提示：当前脚手架安装的依赖信息可以从 `package.json` 中查看

2.3 使用 Mint UI

A.编辑脚手架的 `src/main.js` 文件

B.书写以下代码：

```
//导入Mint UI组件库
import MintUI from 'mint-ui';

//导入Mint UI的样式表文件
import 'mint-ui/lib/style.min.css';

//通过Vue.use()方法将其注册为组件
Vue.use(MintUI);
```

JS main.js

```
6 import qs from 'qs'
7 axios.defaults.baseURL = 'http://127.0.0.1'
8 Vue.prototype.axios = axios;
9 Vue.config.productionTip = false
10
11 //导入Mint UI组件库
12 import MintUI from 'mint-ui';
13
14 //导入Mint UI的样式表文件
15 import 'mint-ui/lib/style.min.css';
16
17 //通过Vue.use()方法将MintUI注册为组件
18 Vue.use(MintUI);
19
```

扩展：组件库的安装命令结构如下：

```
npm install --save 组件库名称
```

2.4 Mint UI 组件库构成

- CSS 组件
- JS 组件
- 表单组件

3.CSS 组件

• Header 组件

Header 组件用于实现顶部导航，其语法结构是：

```
<mt-header title="标题信息" fixed>

</mt-header>
```

title 属性用于设置顶部导航的标题信息，string 类型

fixed 属性用于控制是否固定顶部导航，boolean 类型

当将顶部导航设置 fixed 属性后，其将脱离标准文档流。其位置靠后的元素应设置顶部的内/外边距

顶部导航的高度为 40px

可以在 <mt-header> 组件中嵌套子组件，并且子组件通过 slot="left" 或 slot="right" 显示在顶部导航的左侧或右侧。

src/components/MintUI/Header.vue 的示例代码如下：

```
<template>
  <div>
    <mt-header title="学前端,到学问">
    </mt-header>
  </div>
</template>
```

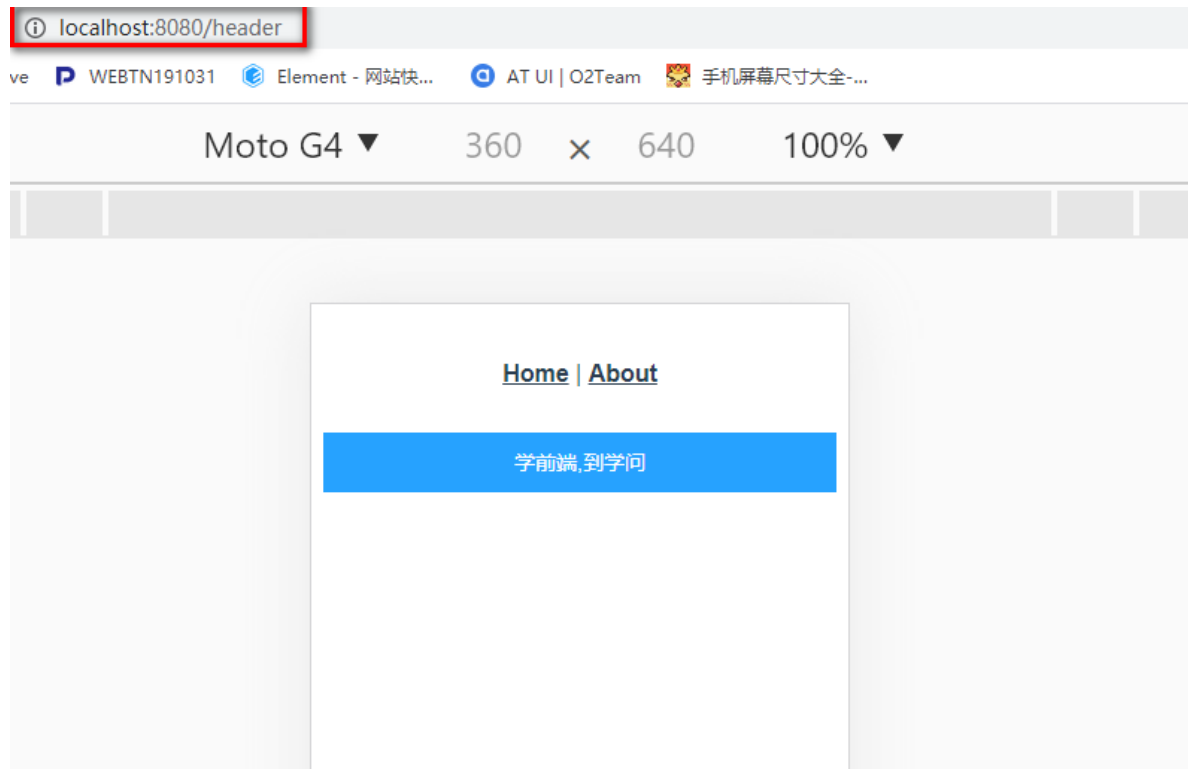
为能够在浏览器中进行访问，所以需要修改 src/router/index.js，示例代码如下：

```
//导入组件
import Header from '../components/MintUI/Header.vue';

//设定路由信息
```

```
const routes = [
  //设定路由信息
  {
    path: '/header',
    component: Header
  }
]
```

此时浏览器的效果如下图所示：



src/components/MintUI/Header.vue 完整示例代码如下：

```
<template>
  <div>
    <mt-header title="学前端,到学问" fixed>
      <span slot="left">返回</span>
      <span slot="right">更多</span>
    </mt-header>
    <div class="container">
      <p v-for="(v,k) in 50" :key="k">
        第{{v}}个段落
      </p>
    </div>
  </div>
</template>
<style scoped>
.container{
  margin-top:40px;
}
</style>
```

• reset.css

每个浏览器对于 HTML 元素都存在默认样式，但是不同的浏览器对于默认样式的规定是不同的，于是需要统一样式来保证所有浏览器的默认样式是相同的 -- reset.css

将 reset.css 存储到根目录下的 style|styles|css 目录内

在 public/index.html 中引入 reset.css 的代码如下：

```
<link rel="stylesheet" href="/styles/reset.css">
```

• button 组件

button 组件用于实现按钮，其语法结构是：

```
<mt-button
  type="按钮类型"
  size="按钮尺寸"
  icon="按钮图标"
  disabled
  plain>
  ...
</mt-button>
```

按钮类型包括：default(默认按钮)、primary(主要按钮)、danger(危险按钮)

按钮尺寸包括：normal(标准)、small(小的)、large(大的)

按钮图标包括：back(返回)、more(更多)

disabled 属性用于控制按钮是否禁用，boolean 类型

plain 属性用于控制按钮是否为朴素按钮，boolean 类型

可以在按钮内嵌套图像，并且为图像设置 slot="icon" 属性，以实现按钮的自定义图标

如果**不为**按钮内的图像添加 slot="icon" 属性的话，形成的结构如下：

```
<button>
  <label>按钮文本</label>
</button>
```

如果**为**按钮内的图像添加 slot="icon" 属性的话，形成的结构如下：

```
<button>
  <span>自定义图标的图像</span>
  <label>按钮文本</label>
</button>
```

4. 表单组件

• Field 组件

Field 用于实现表单输入类型的控件，其语法结构是：

```
<mt-field
  type="输入框的类型"
  label="标签文本"
  placeholder="占位内容"
  state="检测状态"
  disabled
  readonly
  disableClear
  :attr="{原生HTML属性}"
  v-model="变量名称">
</mt-field>
```

输入框的类型有：

- `text`，单行文本框
- `password`，密码框
- `textarea`，多行文本框
- `number`，数字
- `tel`，电话号码
- `email`，电子邮箱
- `url`，URL 地址(如网址，FTP 地址等)

检测状态包括：

- `error` 表示错误
- `success` 表示成功
- `warning` 表示警告

`disabled` 属性用于控制表单控件是否禁用，`boolean` 类型

`readonly` 属性用于控制表单控件是否为只读，`boolean` 类型

`disableClear` 属性用于控制是否禁用清除按钮，`boolean` 类型

`:attr(attribute)` 属性用于控制表单控件的原生属性，`object` 类型，如：

```
<mt-field
  type="password"
  label="密码"
  placeholder="请输入密码"
  v-model="password"
  :attr="{maxLength:'10',autocomplete:'off'}">
</mt-field>
```

作业：

1.完成用户的注册界面

2.只要用户输入完成表单控件之后，应立即检测是否合法，并且显示出相关的状态

3.如何实现表单控件的获取焦点及失去焦点的事件