



开发拓展篇2：将UI 界面交给第三方库

当你了解了Vue 项目构建和开发的基本知识后，我认为接下来你一定想亲自在构建出的项目中填充自己的业务和功能逻辑，因为目前其还是空白的。

但是这里我不会教你如何实现一个具体的业务和功能模块，因为每个人想要实现的东西都可能不尽相同。如果你想快速开发一款应用，并且不想过多的操心页面 **UI** 层次的内容，比如你不想去实现一个下拉UI组件或设计一个 **icon** 图标，那么我想你有必要了解下UI库及图标库的应用。

UI 库

UI库是脱离JS框架外的一种“工具”，相比JS框架可以帮助你实现各种业务逻辑，其更关注于页面UI层面的实现，比如提供和业务无关的弹窗、导航、表单组件等，为项目UI层面的功能提供解决方案，比如[jQuery UI](#)。

而由于本小册介绍的JS框架是Vue，所以在Vue项目中我们需要使用基于Vue开发的UI库。本文将比较流行的[Vux](#)为例，其目前github star数已在14 k左右。

Vux是一款是基于[WeUI](#)和 **Vue(2.x)** 开发的移动端UI组件库，主要服务于微信页面。

Vux 的安装和配置

那么我们如何在项目中使用Vux 呢？首先我们先要进行安装：

```
yarn add vux
```

shell

```
# 或者
```

```
npm install vux --save
```



```
yarn add vux-loader --dev
```

或者

```
npm install vux-loader --save-dev
```

安装完成后，我们需要在项目中配置，而由于目前Vux 官网的配置教程未对Vue CLI 3.x 作出说明，我们先看下其目前的介绍：

```
/* build/webpack.base.conf.js */
const vuxLoader = require('vux-loader')
const webpackConfig = originalConfig // 原来的 module.exports 代码赋值给变量 webpackConfig

module.exports = vuxLoader.merge(webpackConfig, {
  plugins: ['vux-ui']
})
```

官方目前的配置是在 Vue CLI 2.x 的 `build/webpack.base.conf.js` 文件中进行修改，merge `vux-loader` 的配置项。那么在 Vue CLI 3.x 中其实原理是一样的，不一样的地方在于我们无法直接修改 webpack 配置文件，而需要通过 `vue.config.js` 中的 `configureWebpack` 配置项来进行修改罢了。代码如下：

```
/* vue.config.js */
const vuxLoader = require('vux-loader')

module.exports = {
  ...

  configureWebpack: config => {
    vuxLoader.merge(config, {
      plugins: ['vux-ui']
    })
  },
  ...
}
```

`configureWebpack` 配置中提供的 `config` 参数便是 webpack 的配置内容，也可以看作是官方文档中提到的原来在 `webpack.base.conf.js` 中的 `module.exports` 代码。



Vue 项目构建与开发入门

当我们配置好 VUX 后，我们便可以在项目中使用它了。VUX 为我们提供了很多项目中常用的组件和工具函数等，比如我们在全局父组件 App.vue 中添加一个底部导航：

html

```
<!-- App.vue -->

<template>
  <div id="app">
    <router-view/>
    <tabbar>
      <tabbar-item :link="{name: 'demo'}">
        <span slot="label">Demo</span>
      </tabbar-item>
      <tabbar-item :link="{name: 'laboratory'}">
        <span slot="label">实验室</span>
      </tabbar-item>
      <tabbar-item :link="{name: 'about'}">
        <span slot="label">关于</span>
      </tabbar-item>
    </tabbar>
  </div>
</template>

<script>
import { Tabbar, TabbarItem } from 'vux'

export default {
  components: {
    Tabbar,
    TabbarItem,
  }
}
</script>

<style lang="less">
@import '~vux/src/styles/reset.less';
</style>
```

我们通过引入组件的方式将导航组 `Tabbar`、`TabbarItem` 件引入并注册到页面中，这样通过 Vux 文档中的介绍我们便可以对相应组件进行配置。呈现效果如下：



需要注意的是我们需要在 App.vue 中引入 Vux 的 `reset` 样式 less 文件以解决样式呈现不统一的问题。关于其他 Vux 组件的配置可以参考官方文档：[组件](#)

其他 UI 库（框架）

除了上方介绍的 Vux 外，类似的 Vue 的第三方 UI 库还有很多，这里我列举几个比较常用的：

- [iview](#)：一套基于 Vue.js 的高质量 UI 组件库（PC端）
- [iView Admin](#)：搭配使用iView UI组件库形成的一套后台集成解决方案（PC端）
- [Element](#)：一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库（PC端）
- [Vue Antd](#)：Ant Design 的 Vue 实现，开发和服务于企业级后台产品（PC端）
- [VueStrap](#)：一款 Bootstrap 风格的 Vue UI 库（PC端）
- [Mint UI](#)：由饿了么前端开发的基于 Vue.js 的移动端组件库（移动端）
- [Vonic](#)：一个基于 vue.js 和 ionic 样式的 UI 框架，用于快速构建移动端单页应用（移动端）
- [Vant](#)：轻量、可靠的移动端 Vue 组件库（移动端）
- [Cube UI](#)：基于 Vue.js 实现的精致移动端组件库（移动端）

图标库

了解完 UI 库，我们再了解下图标库。图标库，顾名思义就是汇聚了大量图标的仓库，在这样的仓库中我们可以查找并下载我们想要的图标，甚至还可以制定颜色和大小。

在项目中使用图标库可以为我们的项目制定统一的图标管理标准，同时一定程度上也可以减少项目图片的数量。下面我们便来介绍下目前最流行的一款图标库 [Iconfont](#)。

使用 Iconfont 下载管理图标

Iconfont 是阿里妈妈 **MUX** 倾力打造的矢量图标管理、交流平台。设计师将图标上传到 Iconfont 平台，用户可以自定义下载多种格式的 icon，平台也可将图标转换为字体，便于前端工程师自由调整与调用。



在 Iconfont 首页，我们可以点击图标库来进行图标的搜索。这里我们可以点击官方图标库后选择 Ant Design 官方图标库进入。

进入对应的图标库后，我们可以选择对应的图标加入购物车，同时购物车会更新添加后的图标数量。

这里我们采用将图标下载到本地的方式进行使用，当然你也可以使用在线链接，但这会受到网络的影响。

Iconfont 的使用

下载到本地后，我们需要将文件夹中的 `iconfont.css`、`iconfont.eot`、`iconfont.svg`、`iconfont.ttf` 和 `iconfont.woff` 文件统一放到项目中去，比如我们可以放入新建的 `assets` 文件夹的 `iconfont` 中去。而 `iconfont.css` 便是管理这样图标字体的样式文件，我们可以将其引入到入口文件中：

```
/* main.js */javascript  
  
import './assets/iconfont/iconfont.css'
```

引入后我们便可以在项目中通过给 `html` 标签添加样式名称的方式来进行图标的使用，比如我们在上方 `Vux` 的导航上添加图标：

```
<!-- App.vue -->html  
  
<template>  
  <div id="app">  
    <router-view/>  
    <tabbar>
```



```
</tabbar-item>
<tabbar-item>
  <span slot="icon" class="iconfont icon-experiment"></span>
  <span slot="label">实验室</span>
</tabbar-item>
<tabbar-item>
  <span slot="icon" class="iconfont icon-deploymentunit"></span>
  <span slot="label">关于</span>
</tabbar-item>
</tabbar>
</div>
</template>
```

按照 Vux 导航文档添加名称为 `icon` 的 `slot` 插槽后，我们还需要在标签上添加对应图标的 class 名称，比如 `iconfont icon-bulb`，最终我们的展示效果如图所示：

其他图标库

除了 Iconfont，常用的图标库还有：

- [Font Awesome](#)：世界上最受欢迎且最易于使用的图标集
- [Ionicons](#)：精美的开源图标库，可以用于Web，iOS，Android和桌面应用程序
- [Themify](#)：一套用于网页设计和应用程序的完整图标

相信以上这些图标库就足以使你应付所有项目了。

结语

本文介绍了 Vue 项目开发中可能会使用到的 UI 库与图标库的应用，以 Vux 和 Iconfont 为例讲解了它们在项目中的使用方法和注意事项，相信大家能够在项目构建和开发的基础上使用 UI 库与图标库快速实现自己的项目 UI 层面的功能和展示，为自己的项目添砖加瓦。

具体实例代码可以参考：[ui-framework-project](#)



- 查看 Vux 源码，尝试自己编写一个 UI 插件
- Iconfont 是矢量图标库，其相比位图的主要区别是什么？

留言

评论将在后台进行审核，审核通过后对所有人可见

666啊

为什么一用 vux-loader 就报错

▲ 0 评论 9天前

漆伟 前端工程师

有没有pc跟移动端都支持的ui框架

▲ 0 评论 1月前

周公来 搬砖工 @ 某教育公司

好像没有讲到. 用CLI 3 构建一个输出组件的项目.

▲ 0 评论 3月前

毛绒玩具皮卡丘01 前端搬砖，努力成为一个超级水的full-...

只喜欢Material Design，所以使用<https://vuetifyjs.com/zh-Hans>

▲ 0 评论 3月前

staven

写的很棒！我也整理了份vue-cli3配置。a添加alias、externals、gzip压缩、注入全局scss变量，以及上传ali-oss。<https://github.com/staven630/vue-cli3-config>

▲ 0 收起评论 3月前



2月前

评论审核通过后显示

评论

丁一 小前端 @ SN

虽然目前前端只停留在解决一定的技术复杂度问题上，但用户体验视觉美观，更是我们前端开发者所做的东西给人的第一印象，小厂和普通大厂对怎么做出好看精致的东西指望着设计师是完全不够的，还是希望我们工程师能有更好的经验做更好的东西，而不是为了用工具而用工具或单纯只想着拼凑了事...

▲ 1 收起评论 3月前

墨月落羽舞苍风 前端小白

技术圈,业务逻辑有插件乱七八糟的，但是css没有啊
回归本质吧

3月前

烈鸟 前端开发 @ 好耶广告

所以，前端开发的审美，对体验的认知很重要~

3月前

评论审核通过后显示

评论

丁一 小前端 @ SN

每次看组件库都令人迷茫，有一些质量不佳，不维护的依然被 罗列，而且组件库单纯的组装也不漂亮

▲ 0 评论 3月前

丁一 小前端 @ SN

希望作者能给出更多的关于用什么组件库好的理由，怎么用好组件库，以及如何基于组件库做出漂亮的应用，



旁卜 前端工程师 @ TC

感谢提议🙏，会往这方面思考

3月前

评论审核通过后显示

评论

alwynxu 后台工程师

竟然不是element

▲ 0

收起评论 3月前

SHERlocked93 前端打字员 @ 中电

我也投element一票，哈哈

3月前

评论审核通过后显示

评论