顺丰金融前端开发框架 - 基于VUE 2.x

> 本框架使用官方脚手架 **VUE-CLI** 搭建， 异步请求使用 **axios**， 加解密使用 **crypto-js** 。

如何运行项目

```
# 安装依赖
npm install

# 在 localhost:8080 上运行本地开发环境
npm run dev

# 打生产包
npm run build

# build for production and view the bundle analyzer report
npm run build --report

# 测试环境目前还没有进行配置，敬请期待
```

前置条件

使用该基础框架，需要你先掌握以下内容，其中 ES6、VUE，建议通读官方文档：

VUE、ES2015（ES6）、 Webpack、 Sass、Flex布局、HTML5、 CSS3、ESlint

约定

本框架服务于顺丰金融 UED 团队，为确保团队合作的高效，保证代码风格的一致性、可读性，请严格遵守以下约定

1. 本框架内置了官方推荐的非常严格的语法检查器，请不要关闭 ESlint 语法检查，也不要修改相关的配置，这套语法检查遵循的是目前最流行的 ES6 、TypeScript 、Python 等语言的代码风格，从团队的效率以及个人技术成长出发，都应该严格遵守。

```
//以下是一些小Tips，可以让你的 ESlint 不再报错：
// 1、请把你的编辑器的默认缩进改为两个空格
// 2、任何地方都不要使用分号
let a = 1
let b = 2
// 3、为确保代码美观及易读，所有括号的前后，都要加空格
function foo () {/* do something */}
if (true) {/* do something */}
// 4、不要留多余的空行
// 5、不要出现已声明但未使用的变量
// 6、任何注释的前后，都要加上空格，就像我打的这段注释一样
// 7、任何的改变，前期都是痛苦的，过后就是一片光明。ESlint的提示非常准确，请保持耐心坚持下去
```

2. 尽量使用标准的 ES6 语法

```
// 正确示范
import UserInfo from 'UserModel'
let foo = {
    UserInfo,
    todo () {/* do something */}
}

// 错误示范
var UserInfo = require('UserModel');
var foo = {
    UserInfo: UserInfo,
    todo: function () {/* do something */}
}
```

3. 使用 base.sass 中预定义好的样式，尽量不要自己写样式，除非 base 不能满足你的需求，如果你对 base 有什么改进建议，欢迎随时提出。

```
<style lang="scss">
/* 以下是一些例子，请详细阅读 base.sass 中的内容 */
/* 字体大小 .fs-12 ~ .fs-36 */
$i: 12;
@while $i <= 36 {
  .fs-#{$i} {
    font-size: .01rem * $i!important;
  }
  $i: $i + 2;
}
.txt-black{/* 字体颜色 */}
```

```
.bg-red{/* 背景颜色 */}
.border-top{/* 线条 */}
.horizontal-view{ /*水平布局*/}
</style>
<div class="horizontal-view border-bottom align-items-center">
    <div class="card-title vertical-view flex1">
        <h4 class="fs-30 txt-black">建设银行信用卡（1008）</h4>
        <span class="fs-24 txt-gray">单笔额度1万元，单日额度5万元</span>
    </div>
    <base-checkbox :checked="true" />
</div>
```

4. 如非必要，**不要随意改变、添加框架的目录结构**，具体每个目录的作用，请参考 框架指南 - 框架目录结构。
5. ** [非常重要] 为确保组件的高复用性以及可维护性，请使用 单文件组件 ，即布局、样式、逻辑都写在一个文件中，如果实在要把 scss 写到外部文件，请在 components 中单独建立一个目录，把 vue文件、js文件、scss文件放在同一个目录，禁止在公共的目录中，存放非公共的样式文件及脚本文件。**

框架指南

一、框架目录结构

- build -- *与打包相关的文件，由脚手架生成，Webpack的配置基本都在这里*
- config -- *各个环境的配置文件，目前只有开发环境、生产环境，测试环境敬请期待*
- src -- *项目资源目录*
  - assets -- *静态资源目录，与 static 的区别是，该目录的资源会经过webpack进行处理，static不会*
    - images -- *图片存放目录*
    - plugins -- *插件存放目录，关于插件的定义，请参阅 VUE 官方文档*
    - scss -- *样式文件目录*
      - base.scss -- *基础样式库，内置了布局、字体、线条、背景等等的基础样式*
      - variables.scss -- *CSS变量在这里定义*
  - components -- *请把所有组件存放到此，业务组件，请每一个页面单独建立一个文件夹存放，如 home*
    - base -- *基础组件，例如输入框、单选框等*
    - common -- *公共组件，如密码键盘、银行卡卡列表等*
    - home -- *业务组件，主要是各个路由需要使用的组件，如首页、列表页、个人中心等*

    - router -- *路由相关的文件*
    - App.vue -- *项目的根组件*
    - main.js -- *主入口文件*
- static -- *静态资源目录，与 assets 的区别是，assets 的资源会经过webpack进行处理，static不会*

二、页面布局

1、水平布局

2、垂直布局

3、对齐方式

三、基础样式

1、颜色

2、字体

3、线条

四、自定义的 UI 组件

1、关于自定义 UI 的规则和约定

2、BaseCheckbox

3、CommonSecureKeyboard

五、基于 VUX 的 UI 组件

六、请求服务器数据

该框架使用 **crypto-js** 进行加解密，使用 **axios** 进行http请求。

1、关于 search 和 secretKey

该框架遵循顺丰金融前后端约定的交互规则，在从前置进入应用时，会从url中带入部分参数，其中包括 **secretKey** ，该值用于接口通讯、生成摘要、跳转至外部系统（如转入、绑卡等）时使用。

secretKey在发送HTTP请求时会被使用，请与前置进行沟通，严格按照这一名称进行获取

```
// 在任何一个Vue实例中，使用 this.$search 可以获取url中所携带的数据，其中包括secretKey
let search = this.$search
let secretKey = search.secretKey
```

2、发送 HTTP 请求

```
<!--
在任何一个Vue实例中，使用 this.$post 即可发起一个 POST 请求,使用 this.$get 即可发起一个 GET 请求:
this.$post(txCode, reqParams, callback, errorCallback, opts)
txCode[接口名], reqParams[业务参数], callback[成功回调] , errorCallback[失败回调], opts[拓展参数，可不传])
-->
<template>
    <div>Example</div>
</template>

<script>
export default {
    data () {
        return {userId: '123456'}
    },
    methods: {
        getUserInfo () {
            this.$post('getUserInfo', {userid: this.userid}, data => {
                //do something
            }, result => {
                //do something while failed
            }
        }
    }
}
</script>
```