**ENVisual前端架构设计文档**

V0.0.1

陈兴林 @ Enlink 数据中心

目录

[1. 概述 3](#_Toc528655196)

[1.1. 文档说明 3](#_Toc528655197)

[1.2. 项目目录概述 3](#_Toc528655198)

[1.2.1. 项目目录结构 3](#_Toc528655199)

[2. 项目架构说明 4](#_Toc528655200)

[2.1. Css模块 4](#_Toc528655201)

[2.2. I18n模块 5](#_Toc528655202)

[2.3. 字段管理模块 5](#_Toc528655203)

[2.4. 全局变量 5](#_Toc528655204)

[2.5. Echarts 5](#_Toc528655205)

[2.6. Element-ui 5](#_Toc528655206)

[2.7. Flex.css 5](#_Toc528655207)

[2.8. Markdown 5](#_Toc528655208)

[2.9. Util 6](#_Toc528655209)

[2.10. Router 6](#_Toc528655210)

[3. 组件及模块 7](#_Toc528655211)

[3.1. 组件 7](#_Toc528655212)

[3.2. 模块 7](#_Toc528655213)

[3.2.1. 数据源模块 7](#_Toc528655214)

[3.2.2. 面板模块 8](#_Toc528655215)

[3.2.3. 图表模块 8](#_Toc528655216)

[3.2.4. 图表分类模块（需求中以模块命名） 8](#_Toc528655217)

[3.2.5. 数据分析模块 8](#_Toc528655218)

# 概述

本文档描述ENVisual（Enlink数据可视化）产品的前端架构设计，帮助（或辅助）研发人员对项目开发工作顺利有序的进行，方便后续接手人员协作开发（或维护）项目。

## 文档说明

本文档仅描述ENVisual的前端架构设计，数据中心交互设计、UI设计、功能设计不（或仅少部分）在本文档描述范围之内。

本文档针对开发流程作出概括和描述，不涉及具体的开发功能点的详细说明，文档不对外开放，仅供给前端开发使用（或非前端开发人员参考）。本文档不做具体实例演示，每个功能模块的具体说明将同步放置在源码目录中。

## 项目目录概述

### 项目目录结构

**说明：**

项目目录结构设计，便于项目文件分类。项目源码的分类管理，方便项目维护、后续接手人员开发以及维护等工作。

项目目录初始结构，为vue-cli生成目录，可参考vue官网查看说明，此处不做赘述。本文档仅对源码目录（src）下的文件目录进行说明。

#### assets文件夹

该文件夹下放置所有静态资源，包括不需要编译打包的图片、css文件、sass文件等

#### components

该文件夹下放置所有公共vue组件，方便全局（注册）引用。

#### i18n

该文件夹下放置国际化语言相关的配置文件以及语言包

#### router

该文件下放置放置路由配置文件

#### util

放置全局格式化的js插件、全局使用的js类库封装、以及全局参数配置文件

#### view

放置所有前端页面组件

#### App.vue

前端入口组件，所有页面基于该组件进行扩展

#### Main.js

前端js入口文件，所有全局依赖均从该文件导入并使用

# 项目架构说明

本章节描述ENVisual的模块划分以及功能，帮助协作开发人员能更好理解项目架构。

## Css模块

**Sass**

sass模块构建方便css统一管理以及全局设置css基本属性。达到统一配置的目的。本项目没有UI主题任意配置的需求，该设计针对后期维护更方便快捷而设计。

1.所有样式相关文件将放置在assets目录下。Assets目录下构建sass以及style文件夹。Sass文件夹下放置scss后缀文件。Style文件夹下放置.css文件

2.sass目录下构建variable.scss文件，对全局样式进行配置。包括背景色、主题色、点缀色、文字颜色、线框颜色、文字大小、文字字体、阴影、圆角等。

3.sass目录下构建mixin.scss，放置样式混合宏（对公共样式的处理）。例如可以定义trangle来生成三角形。

4.sass目录其他的扩展说明后续将会陆续不上

**Style**

Style模块放置所有.css文件，该目录主要用于全局样式编写以及引用scss文件而扩展sass库。

1.style目录下构建common.css文件，用于统一编写公共样式。公共样式将优于其他样式文件加载，在组件中将可对common.css文件的样式做局部修改。建议协作开发的同事，不要在common.css文件样式中加!important，将样式优先级强制提到最高级。这将影响到全局样式。另外，若有其他全局加载的样式文件，也请务必不得加！important修改全局样式修饰局部变化。

2.style目录下其他文件将视项目进度陆续添加，文档说明将在添加时更新。

## I18n模块

该功能构建基于vue-i18n，配置相关文件放在i18n目录下。其中lang文件夹下的zh.js放置中文语言包，en.js文件放置英文语言包。I18n.js为语言包整合文件，提供给main.js文件使用。

## 字段管理模块

字段管理功能均为全局配置，因此所实现的字段格式化功能都必将放置在util目录下。为方便管理，所有文件以实现具体功能的英文组合命名文件。比如，格式化时间相关的功能，统一放置在formatTime.js里。

## 全局变量

根据需求，本项目基于时间序列，所有数据均构建与时间序列中，同时展示数据也必须依赖时间序列。当页面时间序列段变化时，所有数据（除个别对比性数据外）均重新刷新到新时间序列段。因此，本项目将构建global文件夹，其下放置config.js文件。用于全局配置变量。全局变量在vue组件中以this.config获取。例如：this.config.test获取全局变量test的值：xxx。

## Echarts

Echarts引用百度echarts插件，挂载到Vue.prototype下，在项目中通过this.$echarts获取到echarts全局变量。所有属性和方法都挂载到this.$echarts上。Echarts具体的API请参照echarts官网。

## Element-ui

项目引用饿了么UI插件element-ui，用于项目基础插件UI控制。比如时间选择插件，图片上传插件等。使用介绍请参考element-ui官网。

## Flex.css

Flex.css用于全局布局使用，本项目是基于数据的图形化展示，因此并不考虑ie低版本问题，对于布局可选用布局更人性化的flex，而不是float。因此，引入flex.css进行flex布局是更好的选择。使用详情，详见flexReadme.md文件。

## Markdown

Mardown功能依赖于mavonEditor,是一款基于vue的markdown编辑器。

使用文档详见：https://github.com/hinesboy/mavonEditor

## Util

Util目录除放置字段管理相关js文件外，还包括一些原生js构建的插件。例如：拖拽插件dragDialog.js, 改变元素宽高的插件resizeElemet.js等。

## Router

Router模块，构建全局路由。依赖于vue-router，通过配置文件router.js配置并导出，在main.js注入到vue中，然后全局使用。

路由配置遵循vue路由配置规则，并结合后端开放history模式，不使用hash模式。因此本项目必须构建404.html页面，用于路由出错时显示。

# 组件及模块

组件是只一些功能性统一的.vue文件或.js文件。为实现一个单一的功能而设计，目的是提高代码的可复用性，以减少项目样式表现的差异性以及提高后期项目的可维护性。

模块是对一些组件组装后形成的一些功能较为复杂而又与其他功能有根本性差别的总称。与组件相同之处在于同样表现出较强的可复用性。

对组件的封装，需要有以下几点规则必须遵循：

1. 组件复用后必须紧贴父容器的四个边
2. 组件内跟元素不允许设置外边距、内边距
3. 组件最后一个元素（最右一个元素和最下一排）必须清除外边距
4. 组件本身永远不要添加背景、宽度、浮动、内边距和外边距的样式，组件样式是组件内元素的样式。
5. 每个元素都有且只有一个唯一的且作用于只在组件内的CSS类名，所有的样式都是直接应用到这个选择器上， 并且只有上下文和主题能修改元素的样式。
6. 组件内js强制使用 === 和 !== 代替 == 和 !=，js代码禁止拥有过深的嵌套代码，尽量不适用全局变量，全局变量未使用请务必加上警告信息。函数定义之前，务必给出同名函数警告出口。
7. Js组件封装，请将所有属性和方法挂载到一个独一无二的变量上，务必不要在组件用域下内，存在多个变量或函数。
8. Js组件务必严格添加注释，说明组件的作用，参数的作用，js代码段中也必须添加精简的逻辑说明注释。对区块的注释使用/\*\* \*/格式，处于该格式中的其他行，行前加\*行，说明与星号之间加上空格。对单行代码的注释请使用// 加空格的方式注释，注释尽量精简并精准的说明代码的作用或逻辑。
9. 请务必，每个组件都出具相关的说明文档，用于协作开发使用。

## 组件

组件主要是对一些页面元素的封装，比如按钮的基本样式，输入框的样式等，在项目发开工作启动后，将陆续补全该文档。

## 模块

模块是项目功能的划分，是组件的组合以及一些特定功能的实现。原则上模块的构建除了要消除模块与模块之间的耦合性之外，还需要实现模块较高的可扩展性，用于适应项目的复杂环境而做出不同的表现形式。因此，模块中可能嵌套模块，但嵌套之间并不存在必然的耦合关系。

### 数据源模块

数据源模块实现依赖后端规则划分，前端配合后端对数据源进行划分，具体表现为用户可配置数据源、可配置数据源中的字段，划分后的数据源称为一个索引。

数据源模块主要以element-ui结合axios请求后端数据接口实现

### 面板模块

面板模块依赖拖拽插件，element resize插件，element-ui table插件以及百度echarts插件。该模块独立于图表之外，又依赖于图表配置的数据。数据来源于图表配置，但以数据储存、前端获取渲染的方式构建。

### 图表模块

图表模块以完成各种图表的配置为主，其中包括echarts的配置、表格配置、markdown等。配置依赖于数据源的配置，在数据源的配置基础上，根据索引配置图表数据。完成后图表保存到数据库，最后展示到面板模块。对于图表分类（需求中以模块命名）可见的，图表将分类到导航中进行导航。

### 图表分类模块（需求中以模块命名）

图表分类是图表配置中选择分类类型后，图表将以划分类的方式提供包含关系，最终在导航上表现并可链接到相关图表。

### 数据分析模块

数据分析模块依赖于全站的完整性，在全站开发基本成熟的基础上，构建可关联分析、下钻分析、分析关系可管理配置等。

关联分析实现以构建临时过滤条件为基础，可提供给用户进行跨表过滤分析相同数据之间的差异，不同数据之间的共性等。

下钻分析提供图标的构建关系，可实现一个图标中下钻到另一个图表的表现形式。

配置管理主要管理下钻分析关联关系，提供增删改查的功能。