# IDE

https://code.visualstudio.com/docs?start=true

# 应用实例

<http://blog.gdfengshuo.com/example/#/>

# 基本配置

## package.json

依赖的包关系。里面包括了很多vue的组件。

## vue.config.js

定义了开发服务器的代理，所有restful的API都根据URL前缀路由到对应的服务器上。

## babel.config.js

## postcss.config.js

# 代码

## ./main.js

程序入口。加载了所有vue，css，JavaScript的资源。

会加载下面对象形成一个Vue对象，根据输入的URL来根据router调用对应的vue组件。

|  |
| --- |
| Vue.config.productionTip = false  Vue.use(VueI18n);  Vue.use(ElementUI, {  size: 'small'  });  new Vue({  router,  i18n,  render: h => h(App)  }).$mount('#app') |

## ./App.vue

主页面的模板和CSS。

## ./router/index.js

页面路由设置。定义了各个URL和Vue组件的对应关系。映射关系如下：

<http://localhost:8081/#/dashboard>

对应

../components/page/Dashboard.vue

## 组件

### ./components/common/Home.vue

这是第一个Vue的模板。连包括了若干子模板。给页面划分成三大块：header, sidebar, content-box。

|  |
| --- |
| <template>  <div class="wrapper">  <v-head></v-head>  <v-sidebar></v-sidebar>  <div class="content-box" :class="{'content-collapse':collapse}">  <v-tags></v-tags>  <div class="content">  <transition name="move" mode="out-in">  <keep-alive :include="tagsList">  <router-view></router-view>  </keep-alive>  </transition>  </div>  </div>  </div>  </template> |
| 上面每一块对应一个Vue： |
| import vHead from './Header.vue';  import vSidebar from './Sidebar.vue';  import vTags from './Tags.vue';  import bus from './bus'; |

### ./components/common/Header.vue

### ./components/common/Sidebar.vue

### ./components/common/Tags.vue

### ./components/common/bus.js

### ./components/common/directives.js

### ./components/common/i18n.js

## 页面

### ./components/page/403.vue

### ./components/page/404.vue

### ./components/page/BaseCharts.vue

### ./components/page/BaseForm.vue

### ./components/page/BaseTable.vue

### ./components/page/Dashboard.vue

### ./components/page/DragDialog.vue

### ./components/page/DragList.vue

### ./components/page/I18n.vue

### ./components/page/Icon.vue

### ./components/page/Login.vue

### ./components/page/Markdown.vue

### ./components/page/Permission.vue

### ./components/page/Tabs.vue

### ./components/page/Upload.vue

### ./components/page/VueEditor.vue

## 资源

一些静态的资源，例如CSS，图片，json数据等。

1. ./assets/css/color-dark.css
2. ./assets/css/icon.css
3. ./assets/css/main.css
4. ./assets/css/theme-green/color-green.css
5. ./assets/css/theme-green/fonts/element-icons.ttf
6. ./assets/css/theme-green/fonts/element-icons.woff
7. ./assets/css/theme-green/index.css
8. ./assets/img/img.jpg
9. ./assets/img/login-bg.jpg
10. ./assets/logo.png
11. ./assets/vuetable.json

# 登录机制

在登录成功的时候会设置localStorage。

在main.js里面设置了权限校验的代码：

|  |
| --- |
| router.beforeEach((to, from, next) => {  const role = localStorage.getItem('ms\_username');  if (!role && to.path !== '/login') {  next('/login');  } else if (to.meta.permission) {  // 如果是管理员权限则可进入，这里只是简单的模拟管理员权限而已  role === 'admin' ? next() : next('/403');  } else {  // 简单的判断IE10及以下不进入富文本编辑器，该组件不兼容  if (navigator.userAgent.indexOf('MSIE') > -1 && to.path === '/editor') {  Vue.prototype.$alert('vue-quill-editor组件不兼容IE10及以下浏览器，请使用更高版本的浏览器查看', '浏览器不兼容通知', {  confirmButtonText: '确定'  });  } else {  next();  }  }  }) |
| 如果vue中没有设置perission就要检查localStorage中的role属性，如果不是admin就会重定向到/403页面中去。 |

# 依赖组件

## element-ui

整套框架其实都是基于Element-UI的，起步学习文档在这里：

<https://element.eleme.cn/#/zh-CN/component/quickstart>

下面是main.js中的Element-UI引入：

|  |
| --- |
| import Vue from 'vue';  import ElementUI from 'element-ui';  import 'element-ui/lib/theme-chalk/index.css';  import App from './App.vue';  Vue.use(ElementUI);  new Vue({  el: '#app',  render: h => h(App)  }); |

使用element-ui的标记做Vue的模板。

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

下面是组件库：

<https://element.eleme.cn/#/zh-CN/component/table>

<https://element.eleme.cn/#/zh-CN/component/layout>

打开可以显示代码。

### Element-UI清单

|  |
| --- |
| Pagination,  Dialog,  Autocomplete,  Dropdown,  DropdownMenu,  DropdownItem,  Menu,  Submenu,  MenuItem,  MenuItemGroup,  Input,  InputNumber,  Radio,  RadioGroup,  RadioButton,  Checkbox,  CheckboxButton,  CheckboxGroup,  Switch,  Select,  Option,  OptionGroup,  Button,  ButtonGroup,  Table,  TableColumn,  DatePicker,  TimeSelect,  TimePicker,  Popover,  Tooltip,  Breadcrumb,  BreadcrumbItem,  Form,  FormItem,  Tabs,  TabPane,  Tag,  Tree,  Alert,  Slider,  Icon,  Row,  Col,  Upload,  Progress,  Badge,  Card,  Rate,  Steps,  Step,  Carousel,  CarouselItem,  Collapse,  CollapseItem,  Cascader,  ColorPicker,  Transfer,  Container,  Header,  Aside,  Main,  Footer,  Loading,  MessageBox,  Message,  Notification |

#### Pagination

#### Dialog

#### Autocomplete

#### Dropdown

#### DropdownMenu

#### DropdownItem

#### Menu

#### Submenu

#### MenuItem

#### MenuItemGroup

#### Input

#### InputNumber

#### Radio

#### RadioGroup

#### RadioButton

#### Checkbox

#### CheckboxButton

#### CheckboxGroup

#### Switch

#### Select

#### Option

#### OptionGroup

#### Button

#### ButtonGroup

#### Table

#### TableColumn

#### DatePicker

#### TimeSelect

#### TimePicker

#### Popover

#### Tooltip

#### Breadcrumb

#### BreadcrumbItem

#### Form

#### FormItem

#### Tabs

#### TabPane

#### Tag

#### Tree

#### Alert

#### Slider

#### Icon

#### Row

#### Col

#### Upload

#### Progress

#### Badge

#### Card

#### Rate

#### Steps

#### Step

#### Carousel

#### CarouselItem

#### Collapse

#### CollapseItem

#### Cascader

#### ColorPicker

#### Transfer

#### Container

#### Header

#### Aside

#### Main

#### Footer

#### Loading

#### MessageBox

#### Message

#### Notification

### Element-UI Starter

<https://github.com/ElementUI/element-starter>

代码清单：

|  |
| --- |
| $ find Makefile package.json postcss.config.js README.md src/ webpack.config.js  Makefile  package.json  postcss.config.js  README.md  src/  src/App.vue  src/assets  src/assets/logo.png  src/index.html  src/main.js  src/vendor.js  webpack.config.js |

#### webpack.config.js 整个UI项目的打包配置

定义了入口点：渲染入口和程序入口，打包输出的文件配置，各种资源文件的打包加载方法，

|  |
| --- |
| const resolve = require('path').resolve  const webpack = require('webpack')  const HtmlWebpackPlugin = require('html-webpack-plugin')  const url = require('url')  const publicPath = ''  module.exports = (options = {}) => ({  entry: {  vendor: './src/vendor',  index: './src/main.js'  },  output: {  path: resolve(\_\_dirname, 'dist'),  filename: options.dev ? '[name].js' : '[name].js?[chunkhash]',  chunkFilename: '[id].js?[chunkhash]',  publicPath: options.dev ? '/assets/' : publicPath  },  module: {  rules: [{  test: /\.vue$/,  use: ['vue-loader']  },  {  test: /\.js$/,  use: ['babel-loader'],  exclude: /node\_modules/  },  {  test: /\.css$/,  use: ['style-loader', 'css-loader', 'postcss-loader']  },  {  test: /\.(png|jpg|jpeg|gif|eot|ttf|woff|woff2|svg|svgz)(\?.+)?$/,  use: [{  loader: 'url-loader',  options: {  limit: 10000  }  }]  }  ]  },  plugins: [  new webpack.optimize.CommonsChunkPlugin({  names: ['vendor', 'manifest']  }),  new HtmlWebpackPlugin({  template: 'src/index.html'  })  ],  resolve: {  alias: {  '~': resolve(\_\_dirname, 'src')  },  extensions: ['.js', '.vue', '.json', '.css']  },  devServer: {  host: '127.0.0.1',  port: 8011,  proxy: {  '/api/': {  target: 'http://127.0.0.1:8080',  changeOrigin: true,  pathRewrite: {  '^/api': ''  }  }  },  historyApiFallback: {  index: url.parse(options.dev ? '/assets/' : publicPath).pathname  }  },  devtool: options.dev ? '#eval-source-map' : '#source-map'  }) |

#### Makefile NPM的命令

|  |
| --- |
| .PHONY: dist build  install:  @npm install  dev: install  @npm run dev  build:  @npm run build |

#### package.json 包的依赖关系

包的依赖关系

|  |
| --- |
| {  "name": "element-starter",  "description": "A Vue.js project",  "author": "yi.shyang@ele.me",  "private": true,  "scripts": {  "dev": "webpack-dev-server --inline --hot --env.dev",  "build": "rimraf dist && webpack -p --progress --hide-modules"  },  "dependencies": {  "element-ui": "^2.3.4",  "vue": "^2.5.16"  },  "engines": {  "node": ">=6"  },  "devDependencies": {  "autoprefixer": "^6.6.0",  "babel-core": "^6.24.1",  "babel-loader": "^6.4.0",  "babel-preset-vue-app": "^1.2.0",  "css-loader": "^0.27.0",  "file-loader": "^0.10.1",  "html-webpack-plugin": "^2.24.1",  "postcss-loader": "^1.3.3",  "rimraf": "^2.5.4",  "style-loader": "^0.13.2",  "url-loader": "^0.5.8",  "vue-loader": "^13.3.0",  "vue-template-compiler": "^2.5.16",  "webpack": "^2.4.1",  "webpack-dev-server": "^2.4.2"  }  } |

#### postcss.config.js CSS兼容性转换工具

PostCSS是一个利用JS插件来对CSS进行转换的工具，这些插件非常强大，强大到无所不能。其中，Autoprefixer就是众多PostCSS插件中最流行的一个。

<https://github.com/postcss/postcss>

<https://github.com/postcss/postcss/blob/master/README-cn.md>

PostCSS 是一个允许使用 JS 插件转换样式的工具。 这些插件可以检查（lint）你的 CSS，支持 CSS Variables 和 Mixins， 编译尚未被浏览器广泛支持的先进的 CSS 语法，内联图片，以及其它很多优秀的功能。

PostCSS 在工业界被广泛地应用，其中不乏很多有名的行业领导者，如：维基百科，Twitter，阿里巴巴， JetBrains。PostCSS 的 Autoprefixer 插件是最流行的 CSS 处理工具之一。

PostCSS 接收一个 CSS 文件并提供了一个 API 来分析、修改它的规则（通过把 CSS 规则转换成一个抽象语法树的方式）。在这之后，这个 API 便可被许多插件利用来做有用的事情，比如寻错或自动添加 CSS vendor 前缀。

<https://github.com/postcss/autoprefixer>

感觉是做CSS的兼容的。

细节可以看看：

<https://segmentfault.com/a/1190000010934375?utm_source=tag-newest>

|  |
| --- |
| module.exports = {  plugins: [  require('autoprefixer')()  ]  } |

#### src/main.js

整个应用的main函数入口。

#### src/App.vue

首页UI的加载设置。这些设置会加载到index.html中。

#### src/assets/logo.png

#### src/index.html

首页的静态html信息。

根据webpack.cofig.js的设置，最终的WEB应用会在index.html的body容器中插入JavaScript的引用。例如：

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <title>element-starter</title>  </head>  <body>  <div id="app"></div>  <script type="text/javascript" src="manifest.js?0dd1702b93bf7f27e9e4"></script>  <script type="text/javascript" src="vendor.js?b075ad790bc31c78a2b2"></script>  <script type="text/javascript" src="index.js?f0c9ddb23d55a26a5a2e"></script>  </body>  </html> |
| 相关的配置是： |
| new webpack.optimize.CommonsChunkPlugin({  names: ['vendor', 'manifest']  }),  new HtmlWebpackPlugin({  template: 'src/index.html'  })  ], |

#### src/vendor.js

|  |
| --- |
| import Vue from 'vue'  import ElementUI from 'element-ui' |

### 设置Vue基本组建

可以设定Vue的基本组建。

|  |
| --- |
| Vue.prototype.$loading = Loading.service;  Vue.prototype.$msgbox = MessageBox;  Vue.prototype.$alert = MessageBox.alert;  Vue.prototype.$confirm = MessageBox.confirm;  Vue.prototype.$prompt = MessageBox.prompt;  Vue.prototype.$notify = Notification;  Vue.prototype.$message = Message; |

## babel

Babel is a tool that helps you write code in the latest version of JavaScript. When your supported environments don't support certain features natively, Babel will help you compile those features down to a supported version.

JavaScript的版本兼容工具。

<https://babeljs.io/>

## axios

HTTP客户端库。

## vue-i18n

https://github.com/kazupon/vue-i18n

<https://kazupon.github.io/vue-i18n/>

<https://kazupon.github.io/vue-i18n/introduction.html#sponsors>

代码是一个JavaScript对象：

|  |
| --- |
| const i18n = new VueI18n({  locale: 'zh',  messages  }) |

Messages的格式大致如下：

|  |
| --- |
| export const messages = {  'zh': {  i18n: {  breadcrumb: '国际化产品',  tips: '通过切换语言按钮，来改变当前内容的语言。',  btn: '切换英文',  title1: '常用用法',  p1: '要是你把你的秘密告诉了风，那就别怪风把它带给树。',  p2: '没有什么比信念更能支撑我们度过艰难的时光了。',  p3: '只要能把自己的事做好，并让自己快乐，你就领先于大多数人了。',  title2: '组件插值',  info: 'Element组件需要国际化，请参考 {action}。',  value: '文档'  }  },  'en': {  i18n: {  breadcrumb: 'International Products',  tips: 'Click on the button to change the current language. ',  btn: 'Switch Chinese',  title1: 'Common usage',  p1: "If you reveal your secrets to the wind you should not blame the wind for revealing them to the trees.",  p2: "Nothing can help us endure dark times better than our faith. ",  p3: "If you can do what you do best and be happy, you're further along in life than most people.",  title2: 'Component interpolation',  info: 'The default language of Element is Chinese. If you wish to use another language, please refer to the {action}.',  value: 'documentation'  }  }  } |

Element-ui的国际化

<https://element.eleme.cn/#/zh-CN/component/i18n>

## Echart && Schart

[https://echarts.baidu.com/tutorial.html](https://echarts.baidu.com/tutorial.html#%E5%BC%82%E6%AD%A5%E6%95%B0%E6%8D%AE%E5%8A%A0%E8%BD%BD%E5%92%8C%E6%9B%B4%E6%96%B0)

官方实例：

<https://echarts.baidu.com/examples/>

多行的曲线图：

<https://echarts.baidu.com/examples/editor.html?c=line-stack>

下面是一个对echart的Vue封装

<https://github.com/ecomfe/vue-echarts>

Vue中使用Echart的中文介绍

<https://segmentfault.com/a/1190000015453413>

<https://blog.csdn.net/guozhangqiang/article/details/82379306>

Schart是一个很简陋的chart，只有四种简单的chart图。

<http://blog.gdfengshuo.com/example/sChart/>

## nuxtjs

<https://nuxtjs.org/>

ElEMENT-UI推荐的，做单页应用的框架。