ioc-ui-alpha

初始化

安装依赖

```
npm install 或者 yarn install
```

开发环境运行

```
npm run serve
```

打包

```
npm run build
```

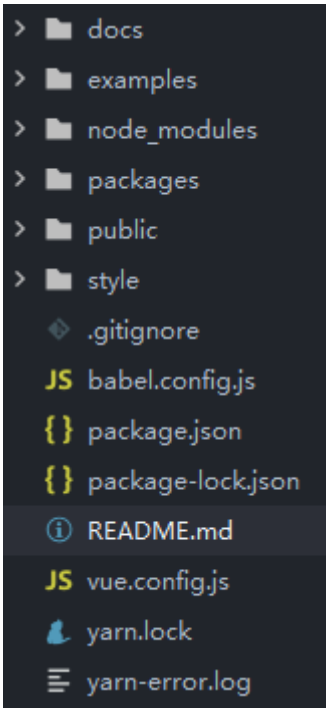
组件说明文档开发环境运行

```
npm run docs:dev
```

组件说明文档打包

```
npm run docs:build
```

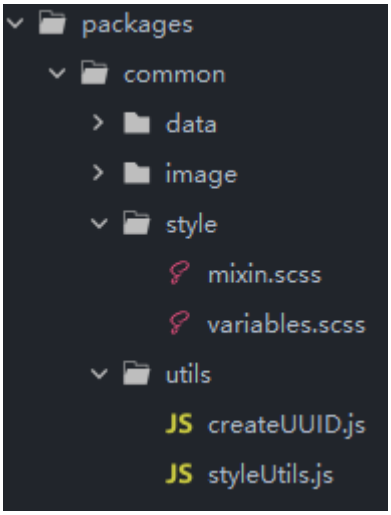
目录结构



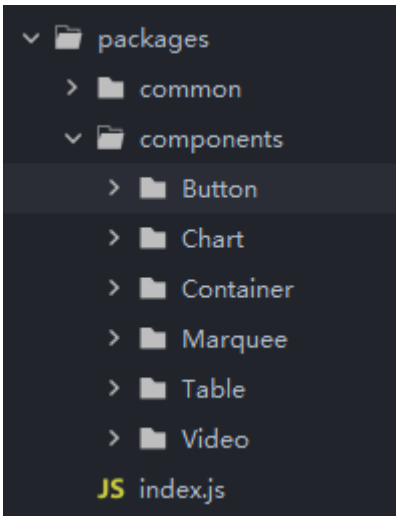

```
-- docs            项目文档说明
-- examples        组件使用案例，展示组件用法
-- packages        组件开发目录，所有的组件开发在此处
-- style           自定义全局样式文件
-- vue.config.js   vue.config配置文件
```

关于 packages目录说明

`common` 文件夹里是常用的样式、静态数据、图片、以及其他函数工具

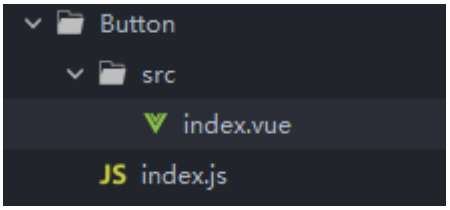


`commponents` 文件是所有开发的组件



关于`commponents`中具体的组件目录，以`Button`组件为例

> 文件夹命名用组件名称，且用大驼峰式



- 其中 **src** 目录里是具体的组件文件

> 如果只有一种类型的该组件，建议以**index.vue**命名；若有多种类型，可以用例如**Table02.vue**、**Table02.vue**命名

> 组件的 **name**属性 必须以**ioc**开头，如**ioc-button**、**ioc-chart**、**ioc-table01**

> 组件名称的**className**也需加上**ioc**前缀

```
<template>
  <div class="ioc-button">
    <slot></slot>
  </div>
</template>

<script>
export default {
  name: 'ioc-button',
  props: {
    type: String
  }
}
</script>

<style scoped>
  .ioc-button {
      display: inline-block;
```

```
      padding: 3px 6px;
      background: #000;
      color: #fff;
  }
</style>
```

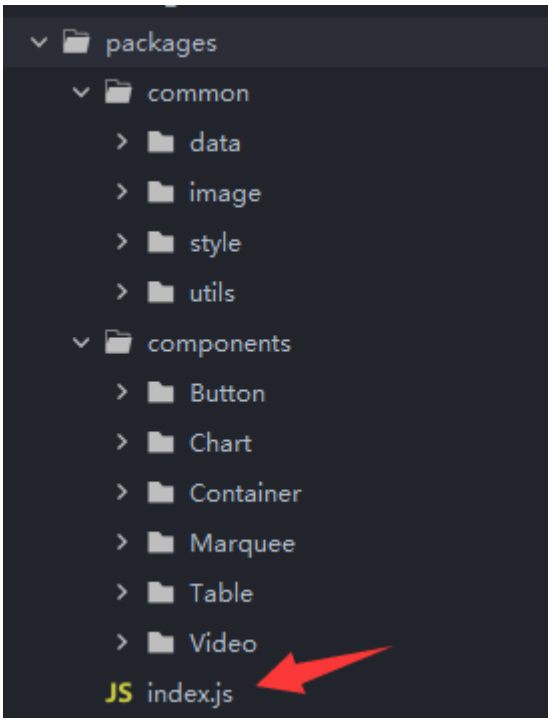
- **index.js**是该组件的出口

```
// 导入组件，组件必须声明 name
import IButton from './src/index.vue'

// 为组件提供 install 安装方法，供按需引入
IButton.install = function (Vue) {
  Vue.component(IButton.name, IButton)
}

// 导出组件
export default IButton
```

若有多种类型组件，需要分类导出

```
// 导入组件，组件必须声明 name
import Table01 from './src/Table01.vue'
import Table02 from './src/Table02.vue'


const tables=[
  Table01,
  Table02
]

// 为组件提供 install 安装方法，供按需引入

tables.map((item)=>{
  item.install = function (Vue) {
    Vue.component(item.name, item)
  }
})

// 导出组件
export {Table01,Table02}
```

- package文件中的`index.js`是组件入口文件，里面汇集每个组件文件夹中的`index.js`



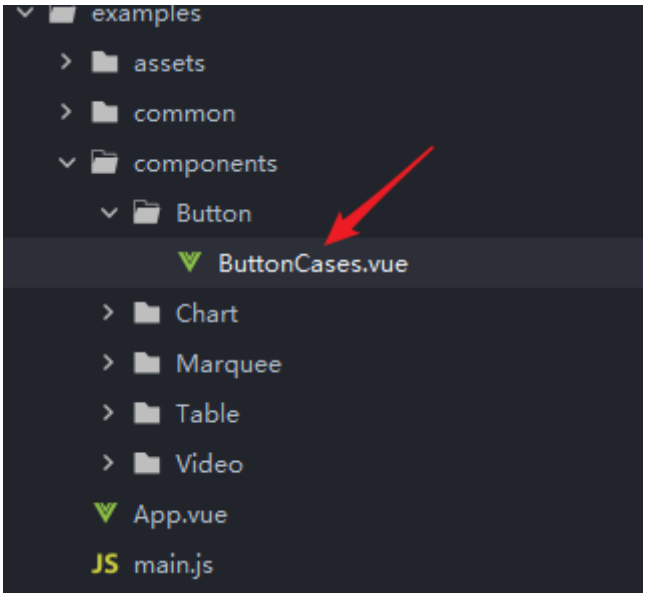

```
//组件入口文件。统一注册

// 导入组件
import IButton from './components/Button/index.js'
import {Chart} from './components/Chart/index.js'
import {Table01,Table02} from './components/Table/index.js'
import {Video} from './components/Video/index'
import {Marquee} from './components/Marquee/index'
// 组件列表
const components = [
  IButton,
  Chart,
  Table01,
  Table02,
  Video,
  Marquee
]

// 定义 install 方法，接收 Vue 作为参数。如果使用 use 注册插件，那么所有的组件都会被注册
const install = function (Vue) {
  // 判断是否安装
  if (install.installed) return
  // 遍历注册全局组件(组件必须有name属性)
  components.map(component => Vue.component(component.name, component))
}

// 判断是否是直接引入文件
if (typeof window !== 'undefined' && window.Vue) {
  install(window.Vue)
}

export default {
  // 导出的对象必须具有 install，才能被 Vue.use() 方法安装
  install,
```

```
  // 以下是具体的组件列表
  IButton,
  Chart,
  Table01,
  Table02,
  Video,
  Marquee
}
```

关于examples说明

该目录类似于一般的vue项目，用于展示packages里面的组件的用法

组件的展示文件夹命名和packages里面保持一致



以**Button**组件用法为例

- 在main.js使用组件

```
// 导入组件库
import iocui from '../packages'
// 注册组件库
Vue.use(iocui)
```

- 新建**ButtonCases.vue**,直接使用ioc-button组件

```
<template>
    <ioc-button type="primary">btn</ioc-button>
</template>

<script>
  export default {
```

```
    name: "ButtonCases"
  }
</script>

<style lang="scss" scoped="scoped">

</style>
```

- 在App.js展示

```
<template>
  <div id="app">
      <ButtonCases />
  </div>
</template>

<script>
import ButtonCases from "./components/Button/ButtonCases";

export default {
  name: 'App',
  components: {
    ButtonCases,

  },
  data(){
    return{

    };
  }
}
</script>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  background-color: #555a65;
  margin: 10px;
}

</style>
```