|  |
| --- |
| 密级：机密 |
| 总页数：91页 |



**IWT前端框架概要设计**

**概要设计**

**(V 0.1)**

**英泰伟业信息技术股份有限公司**

**应用业务开发部**

**2016年11月**

**版本信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本** | **描述** | **作者/修改人** | **备注** |
| 2016-11-7 | 0.1 | 初稿 | 苏晓燕 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目 录**

[**(V 0.1)** 1](#_Toc466367905)

[1. 前端框架开发目标 1](#_Toc466367906)

[1.1. 功能目标 1](#_Toc466367907)

[1.2. 前端开发安全、可靠性目标 1](#_Toc466367908)

[1.3. 前端开发性能目标 2](#_Toc466367909)

[2. 总体设计 3](#_Toc466367910)

[2.1. 技术体系 3](#_Toc466367911)

[2.2. 基础平台架构 3](#_Toc466367912)

[2.3. IWT前端框架总体结构 5](#_Toc466367913)

[2.3.1. Pc后台管理组件开发 7](#_Toc466367914)

[2.3.1.1. 配置文件：新建项目，项目结构为 7](#_Toc466367915)

[2.3.1.2. Basic基础 7](#_Toc466367916)

[2.3.1.2.1. Layout布局 7](#_Toc466367917)

[2.3.1.2.2. Color色彩的定义：根据公司UI设计进行一下仅供参考 9](#_Toc466367918)

[2.3.1.2.3. Typography 字体： 9](#_Toc466367919)

[2.3.1.2.4. Icon图标 10](#_Toc466367920)

[2.3.1.2.5. 按钮 10](#_Toc466367921)

[2.3.1.3. Form 表单 11](#_Toc466367922)

[2.3.1.4. Data数据 12](#_Toc466367923)

[2.3.1.5. Notice 消息框 12](#_Toc466367924)

[2.3.1.6. NAV导航 12](#_Toc466367925)

[2.3.1.7. 其他 12](#_Toc466367926)

[2.3.2. 移动端UI框架 13](#_Toc466367927)

[2.3.2.1. 配置文件：新建项目，项目结构为 13](#_Toc466367928)

[2.3.2.2. Js 组件 13](#_Toc466367929)

[2.3.2.2.1. Toast： 13](#_Toc466367930)

[2.3.2.2.2. Indicator: 13](#_Toc466367931)

[2.3.2.2.3. Loadmore 14](#_Toc466367932)

[2.3.2.2.4. Infinite scroll 14](#_Toc466367933)

[2.3.2.2.5. Message box 14](#_Toc466367934)

[2.3.2.2.6. Action sheet 14](#_Toc466367935)

[2.3.2.2.7. Popup 14](#_Toc466367936)

[2.3.2.2.8. Swipe 15](#_Toc466367937)

[2.3.2.2.9. Lazy load 15](#_Toc466367938)

[2.3.2.2.10. Range 15](#_Toc466367939)

[2.3.2.2.11. Progress 15](#_Toc466367940)

[2.3.2.2.12. Picker 15](#_Toc466367941)

[2.3.2.2.13. Datetime Picker 16](#_Toc466367942)

[2.3.2.2.14. Index List 16](#_Toc466367943)

[2.3.2.2.15. Palette Button (2.0) 16](#_Toc466367944)

[2.3.2.3. css组件 16](#_Toc466367945)

[2.3.2.3.1. Header 16](#_Toc466367946)

[2.3.2.3.2. Tabbar 16](#_Toc466367947)

[2.3.2.3.3. Navbar 17](#_Toc466367948)

[2.3.2.3.4. Button 17](#_Toc466367949)

[2.3.2.3.5. Cell 17](#_Toc466367950)

[2.3.2.3.6. Cell Swipe 17](#_Toc466367951)

[2.3.2.3.7. Spinner 17](#_Toc466367952)

[2.3.2.3.8. TabContainer 18](#_Toc466367953)

[2.3.2.3.9. Search 18](#_Toc466367954)

[2.3.2.4. Form组件 18](#_Toc466367955)

[2.3.2.4.1. Switch 18](#_Toc466367956)

[2.3.2.4.2. Checklist 18](#_Toc466367957)

[2.3.2.4.3. Radio 18](#_Toc466367958)

[2.3.2.4.4. Field 19](#_Toc466367959)

[2.3.2.4.5. Badge 19](#_Toc466367960)

[2.5. 参考文献 22](#_Toc466367966)

# 前端框架开发目标

前端发展迅速的今天，应运而生了许多的前端框架，开发人员在编写前端代码时，从网上找些可以用的代码，或者别的框架的代码粘贴到自己的项目中，这些代码都出自不同的库或框架，造成文件大，版本可能造成冲突，样式不统一，加载慢，渲染慢，执行慢等，前端人员写原始界面，静态页面和特效，开发人员接着改，加逻辑，发现问题要返工，在谁的基础上改和返工，还是合并，造成和很多问题。

IWT编写前端框架的本质是一种管理，对代码的管理。此框架的目的是可以快速的配合开发人员完成开发。达到：加载快，渲染快，执行快，开发快的目的。

结合我们的经验与实际需求，前端框架应实现的功能、达到的性能以及安全、可靠性目标如下：

## 功能目标

前端框架主要功能：桌面UI框架、移动端UI框架

前端框架可划分为：

1. 后台管理UI框架

基于前端模板开发的javascript函数库，为了能建立一个高互动的网页，采用了Ajax,DHTML和DOM等程式码技术。

1. 移动端UI框架

构建数据驱动的web界面和框架、目标是实现响应式数据绑定和组合的视图插件，其核心是一个响应的数据绑定系统。

## 前端开发安全、可靠性目标

* **Web基本攻击手段**

*Web基本攻击大致可以分为：资源枚举、参数操纵、和其他攻击*

* **安全解决方案**

1. *对用户输入的信息进行编码处理和过滤*
2. *谨慎返回用户输入的信息*
3. *使用黑名单和白名单处理*
4. *检查、验证请求来源，对没一个重要的操作都进行验证*
5. *使用ssl防止第三方通信*
6. *不要将重要文件、备份文件存放在公众可访问到的地方；*
7. *会话ID无序化；*
8. *对用户上传的文件进行验证（不单单是格式验证，比方一张gif图片还应将其转为二进制并验证其每帧颜色值<无符号8位>和宽高值<无符号16位>）；*
9. *WSDL文档应当要求用户注册后才能获取；*

* **安全性：**

*Web的安全性问题大部分依赖于后端的过滤和拦截措施。*

## 前端开发性能目标

* **优化方向**

1. *请求数量：合并脚本和样式表，css Sprites拆分初始化负载，划分主域。*
2. *请求带宽：开启GZip，精简JavaScript，移除重复脚本，图像优化*
3. *缓存利用：使用CDN，使用外部JavaScript和CSS，添加Expires头，减少DNS查找，配置ETag，使AjaX可缓存*
4. *页面结构：将样式表放在顶部，将脚本放在底部，尽早刷新文档的输出*
5. *验证码：避免CSS表达式，避免重定向*
6. *全端工程师可利用yuicompressor 或者 goog closure compiler等压缩工具进行优化。*

# 总体设计

## 技术体系

前端框架开发所使用的技术有：

1. Node.js：一个基于 Chrome V8 引擎的 JavaScript 运行环境。
2. Npm：Node.js 的包管理器 npm，是全球最大的开源库生态系统。
3. cooking：自动化。对于需要反复重复的任务，例如压缩（minification）、编译、单元测试、linting等，自动化工具可以减轻你的劳动，简化你的工作。
4. Vue.js是一个构建数据驱动的web界面的库。
5. 栅格系统：页面重构框架。

### 前端的技术比较

#### 服务器比较：

一直以来Javascript作为一种脚本语言出现，并没有涉及到服务器端的动态语言开发，常见的服务器语言有java,php,asp.net等。Node.js从2009年诞生至今,已经发展了两年有余，Node.js 是运行在服务端的 JavaScript。IWT前端框架开发也是基于node.js为平台开发的。

NPM是随同NodeJS一起安装的包管理工具，能解决NodeJS代码部署上的很多问题。

* 允许用户从NPM服务器下载别人编写的第三方包到本地使用。
* 允许用户从NPM服务器下载并安装别人编写的命令行程序到本地使用。
* 允许用户将自己编写的包或命令行程序上传到NPM服务器供别人使用。

#### 市面上许多框架的比较

* 什么叫框架？

框架也算是库的一种，但倾向于重量级一些，倾向于提供整套的解决方案，倾向于创造一些需要你来遵守的规则和范例。

例如：Angular.js

AngularJS 就算是一个框架，因为它提供了一整套的解决方案，需要你按照它设计好的方式来写代码。

* 什么是库？

 库是一些方法的集合。也可以说是框架。

例如：jquery prototype.js zepto.js

Jquery:我们最常用用到的前端类库。

Prototype.js现在已经被淘汰，原因是他没有dom进行优化。而jquery进行有很有的优化。Zeport.js是移动端的类库，功能和jquery类似。

Zepto最初是为移动端开发的库，是jQuery的轻量级替代品，因为它的API和jQuery相似，而文件更小，对任何项目都是个不错的选择。Zepto最大的优势是它的文件大小，只有8k多一点，是目前功能完备的库中最小的一个，尽管不大，Zepto所提供的工具足以满足开发程序的需要。

* 什么是UI组件？

UI组件是针对javascript库，进行的一些集成的组件的开发。

流行的有YUI和 jqueryUI。他们的目的都是起到了快速开发的目的。

其中YUI Compressor是一个压缩工具，可以将css 和js 进行压缩和合并。

减少http请求。

#### 我们为什么选择Vue.js作为核心库进行扩展？

从设计模式上说：前端开发越来越广泛的推行mvc的前端框架模式

* + - 模型（Model） -“数据模型”（Model）用于封装与应用程序的业务逻辑相关的数据以及对数据的处理方法。 “模型”有对数据直接访问的权力。 “模型”不依赖“视图”和“控制器”，也就是说，模型不关心它会被如何显示或是如何被操作。
    - 视图（View） - 视图层能够实现数据有目的的显示，通常是一个用户界面元素。在视图中一般没有程序上的逻辑。在 Web 应用程序中的 MVC，通常把显示动态数据的 html 页面称为视图。
    - 控制器（Controller） - 处理和响应事件，通常是用户操作，并监控模型上的变化，然后去修改视图。

Jquery等插件仅仅是V层。也就是视图层的操作。

Angula.js虽然是mvc的框架，但是缺点在于大而全，学起来，学习成本非常高。版本之间改变非常大，例如AngularJS 2.0会把之前的推翻了重新。又要从头的学一遍。

Vue.js 是尤玉溪老师写的一个用于创建Web交互界面的库，是一个精简MVVM框架。MVVM模式和MVC模式一样，主要目的是分离视图（view）和（模型）。

1. 低耦合。视图（View）可以独立于Model变化和修改，一个ViewModel可以绑定到不同的"View"上，当View变化的时候Model可以不变，当Model变化的时候View也可以不变。
2. 可重用性。你可以把一些视图逻辑放在一个ViewModel里面，让很多view重用这段视图逻辑。
3. 独立开发。开发人员可以专注于业务逻辑和数据的开发（ViewModel），设计人员可以专注于页面设计，使用Expression Blend可以很容易设计界面并生成xml代码。
4. 可测试。界面素来是比较难于测试的，而现在测试可以针对ViewModel来写。

目前有些公司也做了基于Vue.js的移动端开发组件。例如:《饿了吗》。

IWT前端框架也是基于Vue.js 开发组件和《饿了吗》开源组件，开发适合本公司所需要的应用组件。满足学习成本低、快速开发的需要。

## 基础平台架构

IWT前端框架是英泰伟业开发的性能良好前端框架平台。该框架具有良好的分层、模块化设计的特点。适用于移动端和pc桌面端。具有良好的兼容性和扩展性。

### Node.js平台

我们利用node.js平台进行对前端框架的本地构建。

Npm下载安装包。

### Vue.js的原理与项目搭建

Vue.js的原理：

Vue.js 的核心是一个响应的数据绑定系统，它让数据与 DOM 保持同步非常简单。在使用 jQuery 手工操作 DOM 时，我们的代码常常是命令式的、重复的与易错的。Vue.js 拥抱数据驱动的视图概念。通俗地讲，它意味着我们在普通 HTML 模板中使用特殊的语法将 DOM “绑定”到底层数据。一旦创建了绑定，DOM 将与数据保持同步。每当修改了数据，DOM 便相应地更新。这样我们应用中的逻辑就几乎都是直接修改数据了，不必与 DOM 更新搅在一起。这让我们的代码更容易撰写、理解与维护。

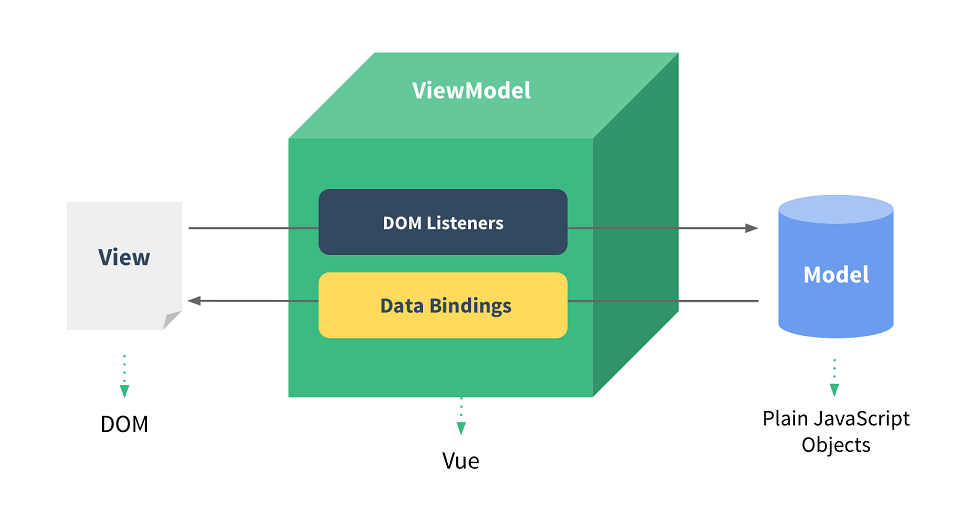


图2‑2-2 Vue.js结构图

* **组件系统**

组件系统是 Vue.js 另一个重要概念，因为它提供了一种抽象，让我们可以用独立可复用的小组件来构建大型应用。如果我们考虑到这点，几乎任意类型的应用的界面都可以抽象为一个组件树：

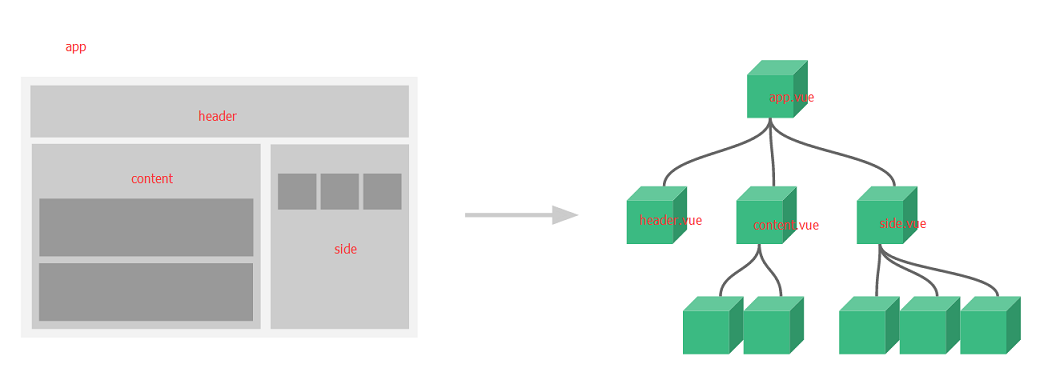
****

图2‑2-2 Vue.js组件树

Web 组件规范仍然远未完成，并且没有浏览器实现。相比之下，Vue.js 组件不需要任何补丁，并且在所有支持的浏览器（IE9 及更高版本）之下表现一致。必要时，Vue.js 组件也可以放在原生自定义元素之内。

Vue.js 组件提供了原生自定义元素所不具备的一些重要功能，比如组件间的数据流，自定义事件系统，以及动态的、带特效的组件替换。

组件系统是用 Vue.js 构建大型应用的基础。另外，Vue.js 生态系统也提供了高级工具与多种支持库，它们和 Vue.js 一起构成了一个更加“框架”性的系统。

* **响应式原理**

Vue.js采用的则是基于依赖收集的观测机制。从原理上来说，和老牌MVVM框架Knockout是一样的。依赖收集的基本原理是：

1.将原生的数据改造成 “可观察对象”。一个可观察对象可以被取值，也可以被赋值。  
2.在watcher的求值过程中，每一个被取值的可观察对象都会将当前的watcher注册为自己的一个订阅者，并成为当前watcher的一个依赖。  
3.当一个被依赖的可观察对象被赋值时，它会通知所有订阅自己的watcher重新求值，并触发相应的更新。  
4.依赖收集的优点在于可以精确、主动地追踪数据的变化，不存在上述提到的脏检查的两个问题。但传统的依赖收集实现，比如Knockout，通常需要包裹原生数据来制造可观察对象，在取值和赋值时需要采用函数调用的形式，在进行数据操作时写法繁琐，不够直观；同时，对复杂嵌套结构的对象支持也不理想。  
Vue.js利用了ES5的Object.defineProperty（数据和视图联动）的方法，直接将原生数据对象的属性改造为getter和setter(这是ES5的特性，需要js解释引擎的支持，无法通过各种打shim补丁来实现。这也是为什么Vue不支持IE8及以下版本的原因)，在这两个函数内部实现依赖的收集和触发，而且完美支持嵌套的对象结构。对于数组，则通过包裹数组的可变方法（比如push）来监听数组的变化。这使得操作Vue.js的数据和操作原生对象几乎没有差别[注:在添加/删除属性，或是修改数组特定位置元素时，需要调用特定的函数，如obj.$add(key, value)才能触发更新。这是受ES5的语言特性所限。在操作对象类型数据的时候一定要注意这点，否则无法实现响应。

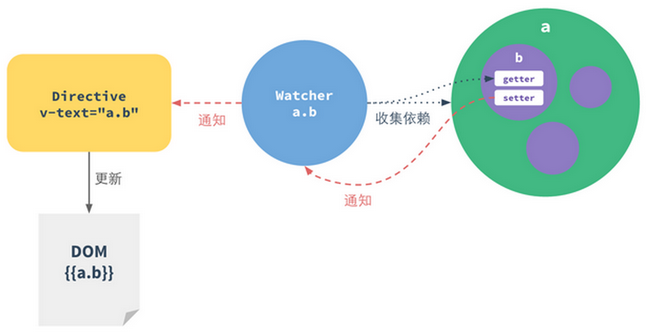


图2‑2-2 Vue.js Object.defineProperty视图联动示意图

### Cooking构建工具的使用快速搭建Vue项

Cooking 项目构建工具具有以下特点：

* 基于 webpack

如果重写原配置文件，只需少量甚至不修改，即可让 cooking 使用已有的 webapck 配置文件。

* 项目零开发依赖

多个项目共用同一个 webpack，省去项目内安装多个开发依赖的过程，从创建项目到启动开发环境花费时间不到十秒。

* 配置简单

将 webpack 的常见配置转换成更直观的参数，简单的配置即可实现 devServer、chunk、ES6 预处理、ESLint 等功能。

* 脚手架

内置脚手架功能，能快速创建包含 cooking 配置文件的空项目。脚手架为 Github 远程仓库，可以轻松定义自己的脚手架。

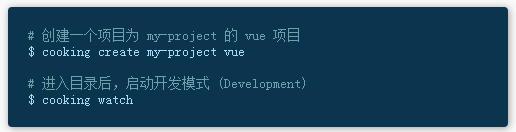
* 插件机制

cooking 只包含最基本的功能，可以用插件机制来增强功能。每个项目可以使用不同的插件，只有在需要时才自动安装。

* 个性化设置

除了插件、脚手架之外，还提供指令、依赖包的扩展，可以将 cooking 打造成适合自己团队使用的构建工具。

快速构建：



## IWT前端框架总体结构

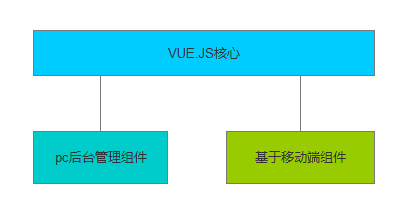


图2‑3IWT框架结构

### Pc后台管理组件开发

**结构图：**

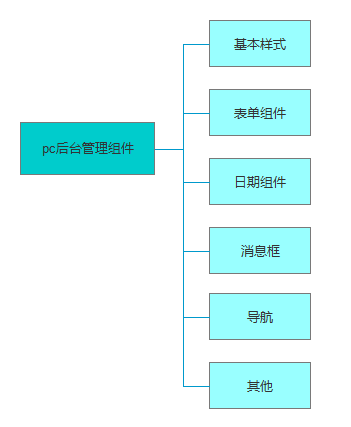
****

图2‑3-1IWT框架结构

#### 配置文件：新建项目，项目结构为

|- src/ --------------------- 项目源代码

|- App.vue --------------

|- main.js -------------- 入口文件

|- .babelrc ----------------- babel 配置文件

|- index.html --------------- HTML 模板

|- package.json ------------- npm 配置文件

|- README.md ---------------- 项目帮助文档

|- webpack.config.json ------ webpack 配置文件

#### Basic基础

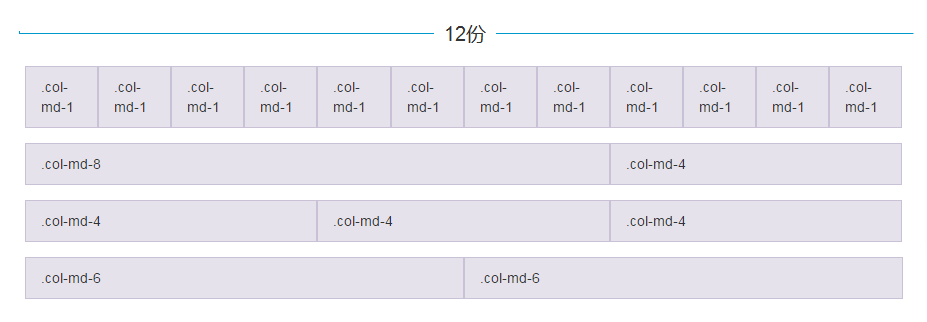
##### Layout布局

栅格布局：通过基础的24分栏，迅速简便的创建布局。

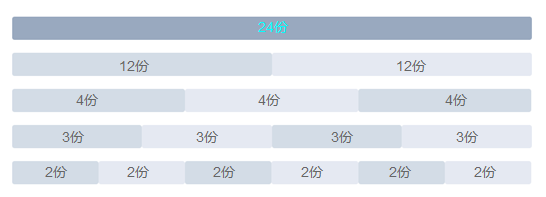
什么是栅格系统定义：

以规则的网格阵列来指导和规范网页中的版面布局以及信息分布。

最古老的栅格系统来自于报纸的布局，国外提出960网格系统就是IWT前端布局的前身。它将屏幕分成均等的12份。来进行布局。



1.1基础布局



1.2分栏间隔

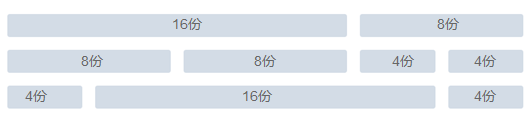
C:\Users\admin\AppData\Local\Temp\AE1.tmp.png

分栏布局的例图：



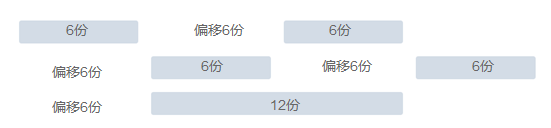
1.3混合布局

通过基础的 1/24 分栏任意扩展组合形成较为复杂的混合布局。



1.4分栏偏移

支持偏移指定的栏数。



1.6行的说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| Gutter | 栅格间隔 | number |  | 0 |
| type | 布局模式，可选 flex，现代浏览器下有效 | string |  |  |
| justify | flex 布局下的水平排列方式 | string | start/  end/  center/  space-around/  space-between | start |
| align | flex 布局下的垂直排列方式 | string | top/middle/bottom | top |

1.7列的说明

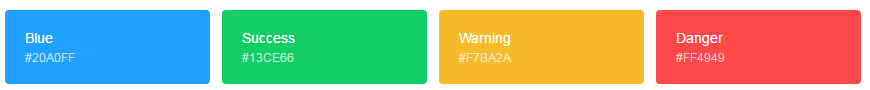
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| Span | 栅格占据的列数，**必选参数** | number |  | - |
| offset | 栅格左侧的间隔格数 | number |  | - |
| Push | 栅格向右移动格数 | number | start/  end/  center/  space-around/  space-between | - |
| Pull | 栅格向左移动格数 | number | top/middle/bottom | - |

##### Color配色方案：根据公司UI设计进行一下仅供参考

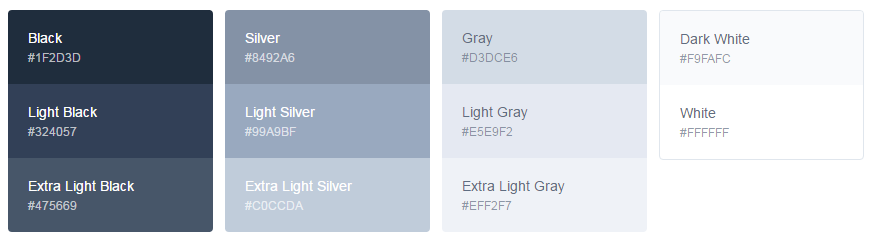
主颜色：



辅助颜色



中性颜色



##### Typography 字体方案：

我们对字体进行统一规范，力求在各个操作系统下都有最佳展示效果。

3.1中文字体：

PingFang SC\ Microsoft YaHei\ Hiragino Sans GB

3.2英文字体：

Helvetica Neue\Helvetica\Arial

3.3 Font-family代码

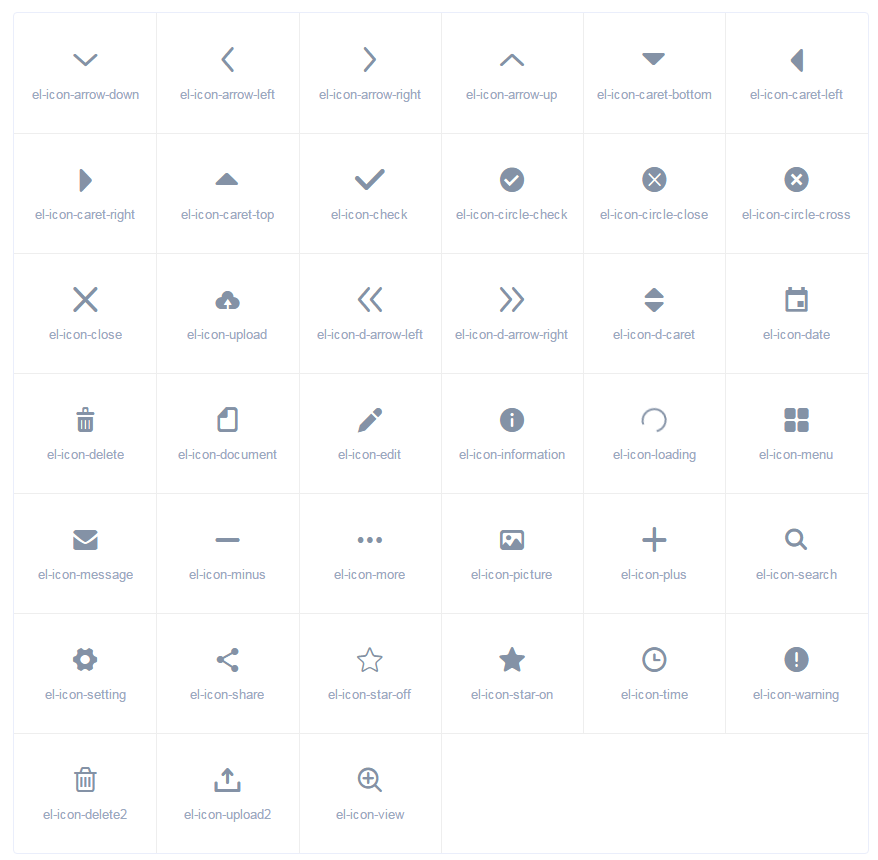
font-family: "Helvetica Neue",Helvetica,"PingFang SC","Hiragino Sans GB","Microsoft YaHei","微软雅黑",Arial,sans-serif;

3.4字体使用规范：

|  |  |  |
| --- | --- | --- |
| 主标题 | 快速搭建页面 | **20px Extra large** |
| 标题 | 快速搭建页面 | **18px Extra large** |
| 小标题 | 快速搭建页面 | **16px Extra large** |
| 正文 | 快速搭建页面 | **14px Extra large** |
| 正文（小） | 快速搭建页面 | **13px Extra large** |
| 辅助文字 | 快速搭建页面 | **12px Extra large** |

##### Icon图标的定义

C:\Users\admin\AppData\Local\Temp\ED56.tmp.png



##### 按钮

5.1 基础用法



5.2禁用状态

C:\Users\admin\AppData\Local\Temp\6B51.tmp.png

5.3 有颜色倾向



5.4图标按钮

C:\Users\admin\AppData\Local\Temp\5D13.tmp.png

5.5 按钮组

C:\Users\admin\AppData\Local\Temp\9B6C.tmp.png

5.6 记载中



5.7不同尺寸的按钮



#### Form 表单

1. Radio 单选表单

基础用法：

C:\Users\admin\AppData\Local\Temp\1A13.tmp.png

禁止状态：

C:\Users\admin\AppData\Local\Temp\3C16.tmp.png

单选组：

C:\Users\admin\AppData\Local\Temp\C6D9.tmp.png

表单样式：

C:\Users\admin\AppData\Local\Temp\346C.tmp.png

1. Checkbox 多选框

基本用法：

C:\Users\admin\AppData\Local\Temp\E381.tmp.png

禁用状态：

C:\Users\admin\AppData\Local\Temp\8129.tmp.png

多选框组：

C:\Users\admin\AppData\Local\Temp\E7D9.tmp.png

3、Input输入框

基本用法：

C:\Users\admin\AppData\Local\Temp\4372.tmp.png

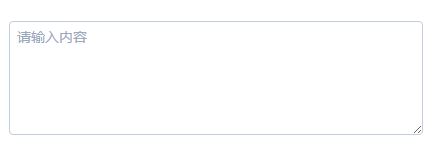
禁用状态：



带图标的输入框：

C:\Users\admin\AppData\Local\Temp\9F1C.tmp.png

文本区域：可调整大小



复合型输入框：



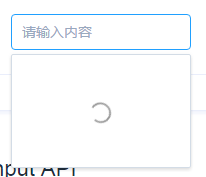
尺寸：



带输入建议：



远程搜索：



4、Input Number 计数器

基本用法：

C:\Users\admin\AppData\Local\Temp\14E6.tmp.png

禁止状态：

C:\Users\admin\AppData\Local\Temp\C0DB.tmp.png

设置步数：

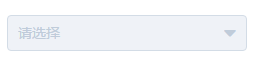


5、Select 选择器

基础用法：



禁用状态：



可清空单选：

C:\Users\admin\AppData\Local\Temp\49F2.tmp.png

基础多线：

C:\Users\admin\AppData\Local\Temp\9F8.tmp.png

自定义模板：

C:\Users\admin\AppData\Local\Temp\7DC2.tmp.png

6、Switch 开关

基本用法：

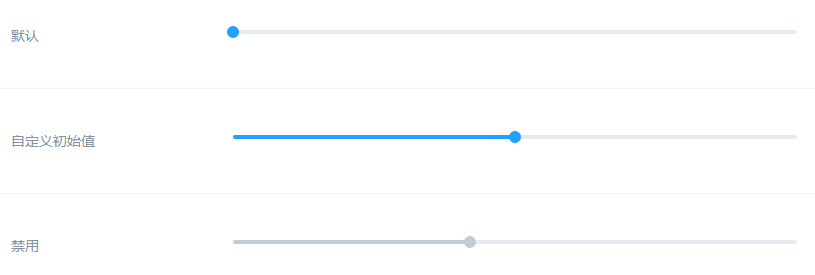
C:\Users\admin\AppData\Local\Temp\D91F.tmp.png

禁用状态：

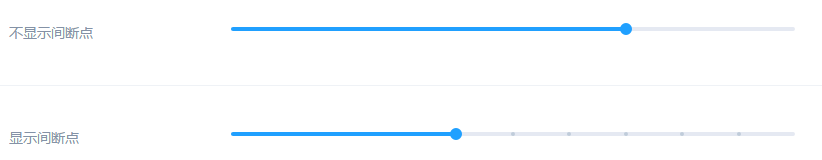
C:\Users\admin\AppData\Local\Temp\3035.tmp.png

7、Slider 滑块

基础用法：



离散值：

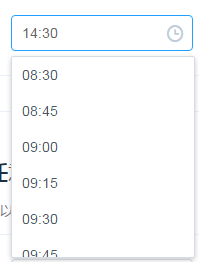


带有输入框：

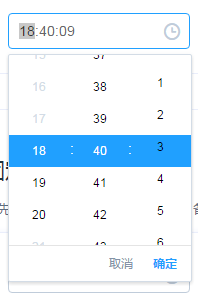


8、Time Picker 时间选择器

固定时间点：



任意时间点：



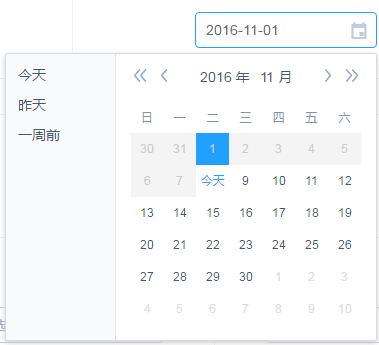
任意时间范围



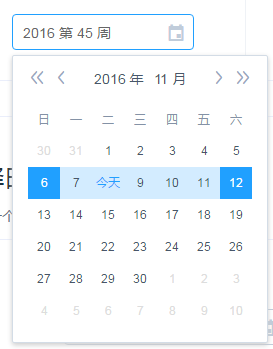
9、Date Picker 日期选择器



带快捷向：



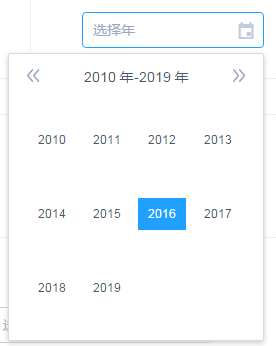
选择周：



选择月：



选择年：



选择日期范围：



10、Datetime Picker 日期时间选择器



11、Upload 上传

点击上传:



拖拽上传：



12、Form 表单



表单验证：



13、Rate 评分

基本用法：



辅助文字：

C:\Users\admin\AppData\Local\Temp\DB88.tmp.png

#### Data数据

1. Table 表格

基本用法：



带斑马纹的表格：



带边框表格：



带状态表格：



固定表头：



固定列：



带固定列和表头：



多选：



排序：



筛选：



1. Tag 标签

基础用法：

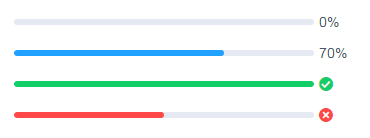
C:\Users\admin\AppData\Local\Temp\46E3.tmp.png

可移除标签：

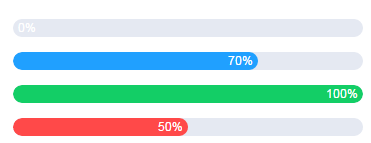
C:\Users\admin\AppData\Local\Temp\8A3B.tmp.png

1. Progress 进度条

线性进度条：



百分比内显：

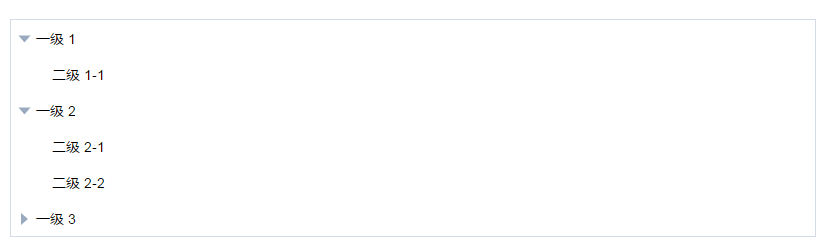


环形进度条：

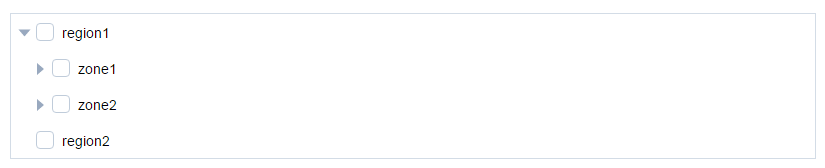


1. Tree 树形控件

基本用法：



可选择：



1. Pagination 分页

基础用法：

C:\Users\admin\AppData\Local\Temp\4A1C.tmp.png

小型分页：

C:\Users\admin\AppData\Local\Temp\FB34.tmp.png

附加功能：显示总页数

C:\Users\admin\AppData\Local\Temp\ABA0.tmp.png

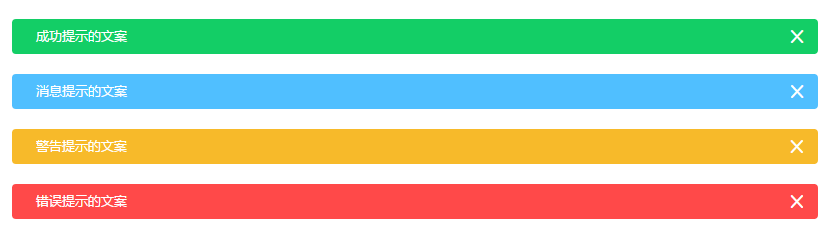
1. Badge 标记

C:\Users\admin\AppData\Local\Temp\8A3D.tmp.png

#### Notice 消息框

1. Alert 警告

基本用法：



1. Loading 加载

区域加载：



整页遮罩加载：

1. Message 消息提示

C:\Users\admin\AppData\Local\Temp\D2B8.tmp.png



1. Message Box 弹框



1. Notification 通知



#### NAV导航

* + - 1. NavMenu 导航菜单



* + - 1. Tabs 标签页



* + - 1. Breadcrumb 面包屑

C:\Users\admin\AppData\Local\Temp\9278.tmp.png

* + - 1. Dropdown 下拉菜单



* + - 1. Steps 步骤



#### 其他

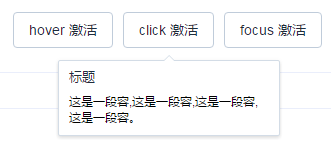
* + - 1. Dialog 对话框



* + - 1. Tooltip 文字提示



* + - 1. Popover 弹出框

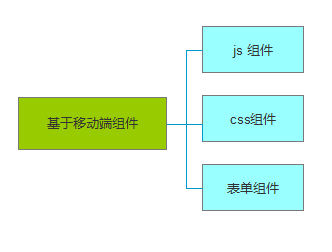


* + - 1. Card 卡片



### 移动端UI框架

结构图：



#### 配置文件：新建项目，项目结构为

|- src/ --------------------- 项目源代码

|- App.vue

|- main.js -------------- 入口文件

|- .babelrc ----------------- babel 配置文件

|- index.html --------------- HTML 模板

|- package.json ------------- npm 配置文件

|- README.md ---------------- 项目帮助文档

|- webpack.config.json ------ webpack 配置文件

#### Js 组件

##### Toast：

简短的消息提示框，支持自定义位置、持续时间和样式。

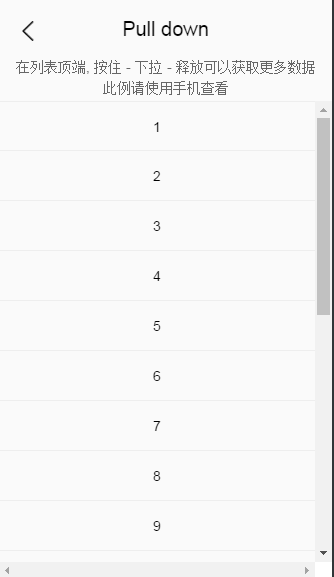


##### Indicator:

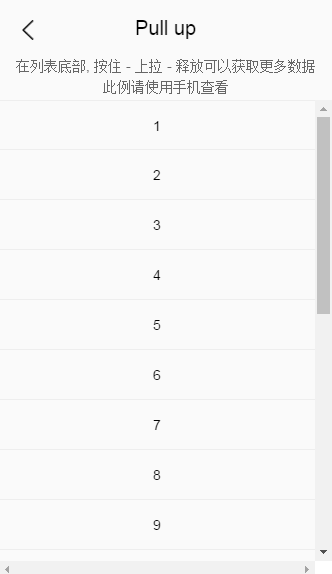
加载提示框，支持自定义文本和加载图标。



##### Loadmore



##### Infinite scroll



##### Message box



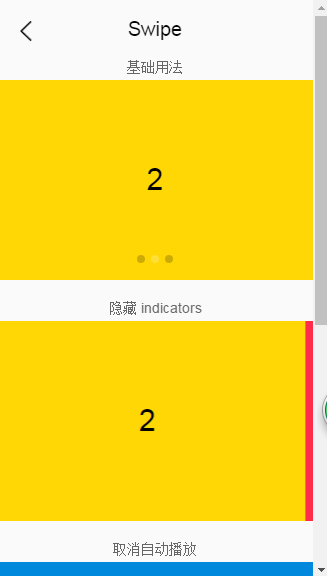
##### Action sheet



##### Popup



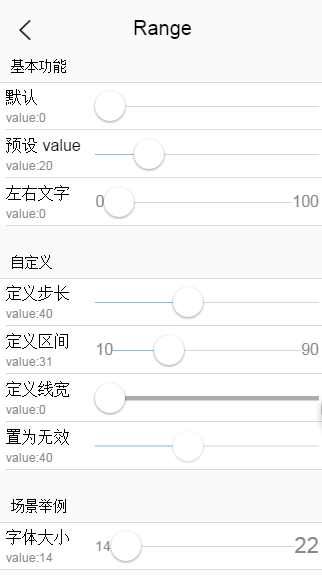
##### Swipe



##### Lazy load



##### Range



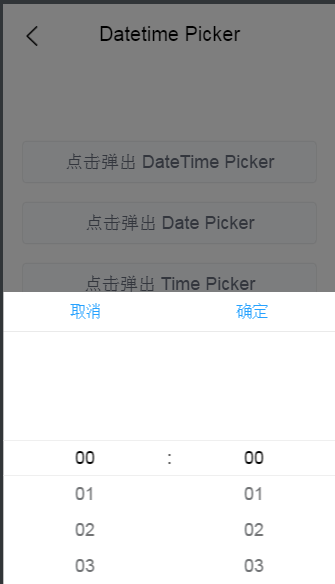
##### Progress



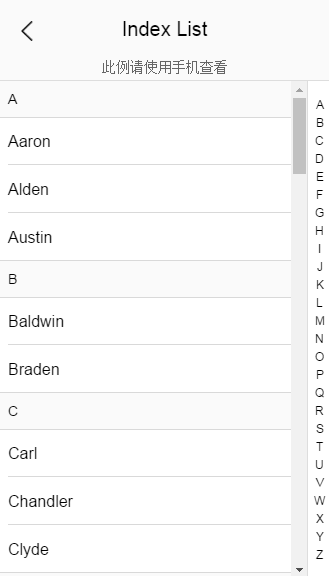
##### Picker



##### Datetime Picker



##### Index List



##### Palette Button (2.0)



#### css组件

##### Header



##### Tabbar



##### Navbar



##### Button



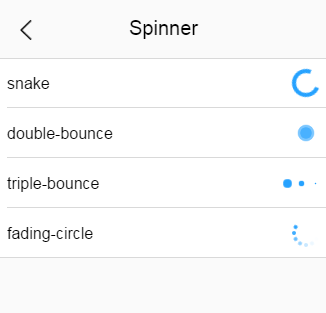
##### Cell



##### Cell Swipe



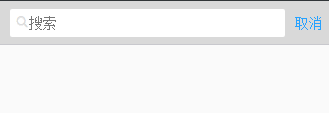
##### Spinner



##### TabContainer

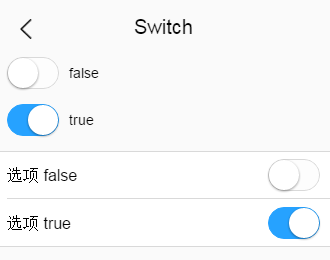


##### Search

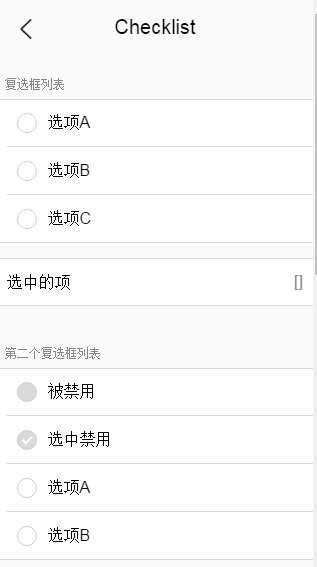


#### Form组件

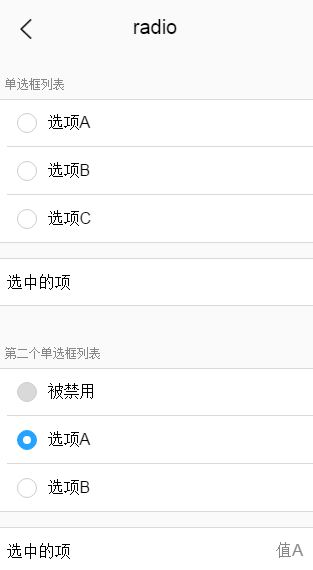
##### Switch



##### Checklist



##### Radio



##### Field



##### Badge



## 参考文献

amaze UI:<http://amazeui.org/getting-started>

ZUI: <http://easysoft.github.io/>

MUI: <http://www.dcloud.io/mui.html>

重置样式表：<http://necolas.github.io/normalize.css/>

后台UI：<http://element.eleme.io/>

Vue mui: <http://mui.yaobieting.com/docs/index.html>

Cooking: <http://cookingjs.github.io/zh-cn/intro.html>