

02

OPEN ORIENTED

凹凸实验室

# Element

王彩暖



element - 饿了么前端团队开发的基于 Vue 的前端样式框架

安装

上手

使用

安装-通过 npm 安装，配合 webpack 使用

```
npm i element-ui -S
```

1、借助 babel-plugin-component，我们可以只引入需要的组件，以达到减小项目体积的目的。

...

**npm install babel-plugin-component -D**

...

# 上手-按需引入element-ui

## 2、.babelrc文件修改

...

```
"plugins": [["component", [ {"libraryName": "element-  
ui", "styleLibraryName": "theme-default"} ]]]
```

...

## 3、主入口文件main.js 引入

...

```
import { Button, Slider, Input, Form, FormItem, Autocomplete } from 'element-ui'
```

```
Vue.use(Button)
```

```
Vue.use(Slider)
```

```
Vue.use(Input)
```

```
Vue.use(Form)
```

```
Vue.use(FormItem)
```

```
Vue.use(Autocomplete)
```

...

注意：用到了什么el-xx-yy标签，就要import什么XxYy组件

如<el-form-item label="项目进度："></el-form-item>

下面主要以form组件举例：

<http://element.eleme.io/#/zh-CN/component/form>



下面主要以form组件举例：

<http://element.eleme.io/#/zh-CN/component/form>

在 Form 组件中，**每一个表单域由一个 Form-Item 组件构成**，  
表单域中可以放置各种类型的**表单控件**，包括  
Input、Slider、Autocomplete、DatePicker、  
TimePicker等

```
<el-form ref="refForm" :model="form" :rules="rules" label-width="130px" :data-daily-id="form.id">

  <el-form-item label="项目名称: " prop="name"> |
</el-form-item>

  <el-form-item label="项目进度: "> 
</el-form-item>

  <el-form-item label="完成细节: "> 
</el-form-item>

  <el-form-item label="问题反馈: "> 
</el-form-item>

  <el-form-item class="edit_list_btn elb"> 
</el-form-item>

</el-form>
```

表单控件slider:

通过拖动滑块在一个固定区间内进行选择

```
<el-form-item label="项目进度: ">
  <div class="edit_list_main">
    <div class="edit_list_input" v-show="editing">
      <el-slider
        v-model="form.progress"
        show-input>
      </el-slider>
    </div>
    <div class="edit_list_inputed" v-text="form.progress+'%'"
      v-show="!editing"></div>
  </div>
</el-form-item>
```

表单控件input:

\* input的类型type, 如``**type="textarea"**``

\* 自适应内容高度 如

``**:autosize="{ minRows: 5 }"**``

```
<el-form-item label="完成细节: ">
  <div class="edit_list_main">
    <div class="edit_list_input" v-show="editing">
      <el-input type="textarea" v-model="form.finished" :
        autosize="{ minRows: 5 }"></el-input>
    </div>
    <div class="edit_list_inputed" v-text="form.finished ||
      主人很懒, 什么也没留下' " v-show="!editing">无</div>
    </div>
  </el-form-item>
```

表单控件autocomplete:

autocomplete 是一个可带输入建议的输入框组件,

**fetch-suggestions** 是一个返回输入建议数据的方法属性, 如 定义  
querySearch(queryString, cb), 该方法中通过  
cb(data) 返回数据到 autocomplete 组件中。

**注意:** 返回数据的每个object必须要有value字段, 否则输入建议为空

```
<el-form-item label="项目名称: " prop="name">
  <div class="edit_list_main">
    <div class="edit_list_input" v-show="editing">
      <el-autocomplete
        class="inline-input"
        v-model="form.name"
        :fetch-suggestions="querySearch"
        placeholder="请输入项目名称"
      ></el-autocomplete>
    </div>
    <div class="edit_list_inputed" v-text="form.name" v-show="!editing"></div>
  </div>
</el-form-item>
```



Form 组件提供了表单验证的功能，只需要通过 rule 属性传入约定的验证规则，同时el-form-item 的 prop 属性设置为需校验的字段名即可，如prop="name"

...

// rule

```
data() {
  return {
    rules: {
      name: [
        { required: true, message: '请输入项目名称', trigger: 'change blur' }
      ]
    }
  }
}
```

...

```
<el-form-item label="项目名称: " prop="name">
  <div class="edit_list_main">
    <div class="edit_list_input" v-show="editing">
      <el-autocomplete
        class="inline-input"
        v-model="form.name"
        :fetch-suggestions="querySearch"
        placeholder="请输入项目名称"
      ></el-autocomplete>
    </div>
    <div class="edit_list_inputed" v-text="form.name" v-show="!editing"></div>
  </div>
</el-form-item>
```

\* 项目名称：

请输入项目名称

项目进度：

0

—

+

完成细节：

问题反馈：

❗ 取消

✓ 确定



02

OPEN ORIENTED

凹凸实验室

THANKS FOR LISTENING

王彩暖