

案例

其他的组件我们通过完成一个页面来学习。

我们要完成如下页面效果

当前状态

当前状态

企业名称

企业名称

品牌名称

品牌名称

查询

批量删除

新增

<input type="checkbox"/>		品牌名称	企业名称	排序	当前状态	操作
<input type="checkbox"/>	1	小米	小米科技有限公司	10	启用	<div>修改删除</div>
<input checked="" type="checkbox"/>	2	小米	小米科技有限公司	10	启用	<div>修改删除</div>
<input type="checkbox"/>	3	小米	小米科技有限公司	10	启用	<div>修改删除</div>
<input checked="" type="checkbox"/>	4	小米	小米科技有限公司	10	启用	<div>修改删除</div>
<input type="checkbox"/>	5	小米	小米科技有限公司	10	启用	<div>修改删除</div>
<input type="checkbox"/>	6	小米	小米科技有限公司	10	启用	<div>修改删除</div>

共 400 条

5条/页

<

1

2

3

4

5

6

...

80

>

前往

1

页

要完成该页面，我们需要先对这个页面进行分析，看页面由哪儿几部分组成，然后到官网进行拷贝并修改。页面总共有如下组成部分



还有一个是当我们点击 `新增` 按钮，会在页面正中间弹出一个对话框，如下

编辑品牌

×

* 品牌名称

* 企业名称

排序

备注

状态

☐

提交

取消

准备基本页面

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<div id="app">

</div>

<script src="js/vue.js"></script>
<script src="element-ui/lib/index.js"></script>
<link rel="stylesheet" href="element-ui/lib/theme-chalk/index.css">

<script>
  new Vue({
    el: "#app"
  })
</script>
</body>
</html>
```

完成表格展示

使用 Element 整体的思路就是 ==拷贝 + 修改==。

拷贝

Element 2.15.5 中文

可将表格内容 highlight 显示，方便区分「成功、信息、警告、危险」等内容。

日期	姓名	地址
2016-05-02	王小虎	上海市普陀区金沙江路 1518 弄
2016-05-04	王小虎	上海市普陀区金沙江路 1518 弄
2016-05-01	王小虎	上海市普陀区金沙江路 1518 弄
2016-05-03	王小虎	上海市普陀区金沙江路 1518 弄

显示代码 在线运行

在左菜单栏找到 `Table 表格` 并点击，右边主体就会定位到表格这一块，找到我们需要的表格效果（如上图），点击 `显示代码` 就可以看到这个表格的代码了。

将html标签拷贝到 `<div id="app"></div>` 中，如下：

```

<template>
  <el-table
    :data="tableData"
    style="width: 100%"
    :row-class-name="tableRowClassName">
    <el-table-column
      prop="date"
      label="日期"
      width="180">
    </el-table-column>
    <el-table-column
      prop="name"
      label="姓名"
      width="180">
    </el-table-column>
    <el-table-column
      prop="address"
      label="地址">
    </el-table-column>
  </el-table>
</template>

```

将css样式拷贝到我们页面的 `head` 标签中，如下

```

<style>
  .el-table .warning-row {
    background: oldlace;
  }

  .el-table .success-row {
    background: #f0f9eb;
  }
</style>

```

将方法和模型数据拷贝到 Vue 对象指定的位置

```
<script>
export default {
  methods: {
    tableRowClassName({row, rowIndex}) {
      if (rowIndex === 1) {
        return 'warning-row';
      } else if (rowIndex === 3) {
        return 'success-row';
      }
      return '';
    }
  },
  data() {
    return {
      tableData: [{
        date: '2016-05-02',
        name: '王小虎',
        address: '上海市普陀区金沙江路 1518 弄',
      }, {
        date: '2016-05-04',
        name: '王小虎',
        address: '上海市普陀区金沙江路 1518 弄',
      }, {
        date: '2016-05-01',
        name: '王小虎',
        address: '上海市普陀区金沙江路 1518 弄',
      }, {

```

拷贝完成后通过浏览器打开可以看到表格的效果

日期	姓名	地址
2016-05-02	王小虎	上海市普陀区金沙江路 1518 弄
2016-05-04	王小虎	上海市普陀区金沙江路 1518 弄
2016-05-01	王小虎	上海市普陀区金沙江路 1518 弄
2016-05-03	王小虎	上海市普陀区金沙江路 1518 弄

表格效果出来了，但是显示的表头和数据并不是我们想要的，所以接下来就需要对页面代码进行修改了。

修改

1. 修改表头和数据

下面是对表格代码进行分析的图解。根据下图说明修改自己的列数和列名

```

<el-table
  :data="tableData"
  style="..."
  :row-class-name="tableRowClassName">
  <el-table-column
    prop="brandName"
    label="品牌名称"
    align="center">
  </el-table-column>
  <el-table-column
    prop="companyName"
    label="企业名称"
    align="center">
  </el-table-column>

```

表示一行，需要几列就需要几个该标签

绑定的数据模型

表头上显示的列名

使数据居中

修改完页面后，还需要对绑定的模型数据进行修改，下图是对模型数据进行分析的图解

```

new Vue({
  el: "#app",
  methods: {...},
  data() {
    return {
      // 表格数据
      tableData: [{
        brandName: '华为',
        companyName: '华为科技有限公司',
        ordered: '100',
        status: "1"
      }, {
        brandName: '华为',
        companyName: '华为科技有限公司',
        ordered: '100',
        status: "1"
      }, {

```

该属性名和标签 `<el-table-column>` 标签的 `prop` 属性值保持一致

表示一条数据

2. 给表格添加操作列

从之前的表格拷贝一列出来并对其进行修改。按钮是从官网的 `Button` 按钮 组件中拷贝并修改的

```

<el-table-column
  align="center"
  label="操作">
  <el-row>
    <el-button type="primary">修改</el-button>
    <el-button type="danger">删除</el-button>
  </el-row>
</el-table-column>

```

按钮组件

3. 给表格添加复选框列和标号列

给表格添加复选框和标号列，效果如下

<input type="checkbox"/>	品牌名称	企业名称	排序	当前状态	操作
<input type="checkbox"/> 1	华为	华为科技有限公司	100	1	修改 删除
<input type="checkbox"/> 2	华为	华为科技有限公司	100	1	修改 删除
<input type="checkbox"/> 3	华为	华为科技有限公司	100	1	修改 删除
<input type="checkbox"/> 4	华为	华为科技有限公司	100	1	修改 删除

此效果也是从 Element 官网进行拷贝，先找到对应的表格效果，然后将其对应代码拷贝到我们的代码中，如下是复选框列官网效果图和代码

<input type="checkbox"/>	日期	姓名	地址
<input type="checkbox"/>	2016-05-03	王小虎	上海市普陀区金沙江路 1518 弄
<input type="checkbox"/>	2016-05-02	王小虎	上海市普陀区金沙江路 1518 弄
<input type="checkbox"/>	2016-05-04	王小虎	上海市普陀区金沙江路 1518 弄

```
<template>
  <el-table
    ref="multipleTable"
    :data="tableData"
    tooltip-effect="dark"
    style="width: 100%"
    @selection-change="handleSelectionChange">
    <el-table-column
      type="selection"
      width="55">
    </el-table-column>
    <el-table-column
      label="日期"
      width="120">
```

这里需要注意在 `<el-table>` 标签上有一个事件 `@selection-change="handleSelectionChange"`，这里绑定的函数也需要从官网拷贝到我们自己的页面代码中，函数代码如下：

```
handleSelectionChange(val) {
  this.multipleSelection = val;
}
```

从该函数中又发现还需要一个模型数据 `multipleSelection`，所以还需要定义出该模型数据
标号列也用同样的方式进行拷贝并修改。

完成搜索表单展示

在 Element 官网找到横排的表单效果，然后拷贝代码并进行修改

Element

搜索文档

指南

组件

主题

资源

2.15.5

中文

Upload 上传

Rate 评分

ColorPicker 颜色选择器

Transfer 穿梭框

Form 表单

当垂直方向空间受限且表单较简单时，可以在一行内放置表单。

审批人

审批人

活动区域

活动区域

查询

显示代码

在线运行

点击上面的 显示代码 后，就会展示出对应的代码，下面是对这部分代码进行分析的图解

```

<el-form :inline="true" :model="formInline" class="demo-form-inline">
  <el-form-item label="审批人">
    <el-input v-model="formInline.user" placeholder="审批人"></el-input>
  </el-form-item>
  <el-form-item label="活动区域">
    <el-select v-model="formInline.region" placeholder="活动区域">
      <el-option label="区域一" value="shanghai"></el-option>
      <el-option label="区域二" value="beijing"></el-option>
    </el-select>
  </el-form-item>
  <el-form-item>
    <el-button type="primary" @click="onSubmit">查询</el-button>
  </el-form-item>
</el-form>
<script>
  export default {
    data() {
      return {
        formInline: {
          user: '',
          region: ''
        }
      }
    },
    methods: {
      onSubmit() {
        console.log('submit!');
      }
    }
  }

```

绑定的模型数据

模型数据

绑定的函数

然后根据我们要的效果修改代码。

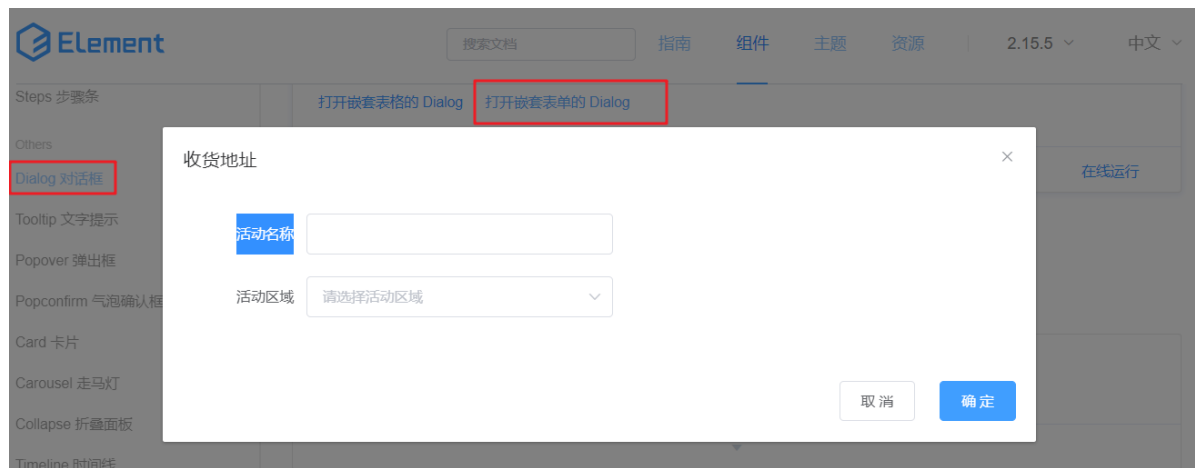
完成批量删除和新增按钮展示

从 Element 官网找具有着色效果的按钮，并将代码拷贝到我们自己的页面上



完成对话框展示

在 Element 官网找对话框，如下：



下面对官网提供的代码进行分析



上图分析出来的模型数据需要在 Vue 对象中进行定义。

完成分页条展示

在 Element 官网找到 `Pagination 分页`，在页面主体部分找到我们需要的效果，如下



点击 `显示代码`，找到 `完整功能` 对应的代码，接下来对该代码进行分析



上面代码属性说明：

- `page-size`：每页显示的条目数
- `page-sizes`：每页显示个数选择器的选项设置。

`:page-sizes="[100,200,300,400]"` 对应的页面效果如下：



- `currentPage`：当前页码。我们点击那个页码，此属性值就是几。
- `total`：总记录数。用来设置总的的数据条目数，该属性设置后，Element 会自动计算出需分多少页并给我们展示对应的页码。

事件说明：

- `size-change`：page-size 改变时会触发。也就是当我们改变了每页显示的条目数后，该事件会触发。
- `current-change`：currentPage 改变时会触发。也就是当我们点击了其他的页码后，该事件会触发。

完整页面代码

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
    .el-table .warning-row {
      background: oldlace;
    }
    .el-table .success-row {
      background: #f0f9eb;
    }
  </style>
</head>
<body>
<div id="app">
  <!--搜索表单-->
  <el-form :inline="true" :model="brand" class="demo-form-inline">
    <el-form-item label="当前状态">
      <el-select v-model="brand.status" placeholder="当前状态">
        <el-option label="启用" value="1"></el-option>
        <el-option label="禁用" value="0"></el-option>
      </el-select>
    </el-form-item>

    <el-form-item label="企业名称">
      <el-input v-model="brand.companyName" placeholder="企业名称"></el-
input>
    </el-form-item>

    <el-form-item label="品牌名称">
      <el-input v-model="brand.brandName" placeholder="品牌名称"></el-input>
    </el-form-item>
```

```

        <el-form-item>
            <el-button type="primary" @click="onSubmit">查询</el-button>
        </el-form-item>
    </el-form>

    <!--按钮-->
    <el-row>
        <el-button type="danger" plain>批量删除</el-button>
        <el-button type="primary" plain @click="dialogVisible = true">新增</el-
button>
    </el-row>

    <!--添加数据对话框表单-->
    <el-dialog
        title="编辑品牌"
        :visible.sync="dialogVisible"
        width="30%">
        <el-form ref="form" :model="brand" label-width="80px">
            <el-form-item label="品牌名称">
                <el-input v-model="brand.brandName"></el-input>
            </el-form-item>

            <el-form-item label="企业名称">
                <el-input v-model="brand.companyName"></el-input>
            </el-form-item>

            <el-form-item label="排序">
                <el-input v-model="brand.ordered"></el-input>
            </el-form-item>

            <el-form-item label="备注">
                <el-input type="textarea" v-model="brand.description"></el-
input>
            </el-form-item>

            <el-form-item label="状态">
                <el-switch v-model="brand.status"
                    active-value="1"
                    inactive-value="0"
                ></el-switch>
            </el-form-item>
            <el-form-item>
                <el-button type="primary" @click="addBrand">提交</el-button>
                <el-button @click="dialogVisible = false">取消</el-button>
            </el-form-item>
        </el-form>
    </el-dialog>

    <!--表格-->
    <template>
        <el-table
            :data="tableData"
            style="width: 100%"
            :row-class-name="tableRowClassName"
            @selection-change="handleSelectionChange">
            <el-table-column
                type="selection"

```

```

        width="55">
    </el-table-column>
    <el-table-column
        type="index"
        width="50">
    </el-table-column>
    <el-table-column
        prop="brandName"
        label="品牌名称"
        align="center">
    </el-table-column>
    <el-table-column
        prop="companyName"
        label="企业名称"
        align="center">
    </el-table-column>
    <el-table-column
        prop="ordered"
        align="center"
        label="排序">
    </el-table-column>
    <el-table-column
        prop="status"
        align="center"
        label="当前状态">
    </el-table-column>
    <el-table-column
        align="center"
        label="操作">
        <el-row>
            <el-button type="primary">修改</el-button>
            <el-button type="danger">删除</el-button>
        </el-row>
    </el-table-column>

</el-table>
</template>

<!--分页工具条-->
<el-pagination
    @size-change="handleSizeChange"
    @current-change="handleCurrentChange"
    :current-page="currentPage"
    :page-sizes="[5, 10, 15, 20]"
    :page-size="5"
    layout="total, sizes, prev, pager, next, jumper"
    :total="400">
</el-pagination>

</div>
<script src="js/vue.js"></script>
<script src="element-ui/lib/index.js"></script>
<link rel="stylesheet" href="element-ui/lib/theme-chalk/index.css">
<script>
    new Vue({
        el: "#app",
        methods: {
            tableRowClassName({row, rowIndex}) {

```

```

        if (rowIndex === 1) {
            return 'warning-row';
        } else if (rowIndex === 3) {
            return 'success-row';
        }
        return '';
    },
    // 复选框选中后执行的方法
    handleSelectionChange(val) {
        this.multipleSelection = val;

        console.log(this.multipleSelection)
    },
    // 查询方法
    onSubmit() {
        console.log(this.brand);
    },
    // 添加数据
    addBrand(){
        console.log(this.brand);
    },
    //分页
    handleSizeChange(val) {
        console.log(`每页 ${val} 条`);
    },
    handleCurrentChange(val) {
        console.log(`当前页: ${val}`);
    }
},
data() {
    return {
        // 当前页码
        currentPage: 4,
        // 添加数据对话框是否展示的标记
        dialogVisible: false,

        // 品牌模型数据
        brand: {
            status: '',
            brandName: '',
            companyName: '',
            id: '',
            ordered: '',
            description: ''
        },
        // 复选框选中数据集合
        multipleSelection: [],
        // 表格数据
        tableData: [{
            brandName: '华为',
            companyName: '华为科技有限公司',
            ordered: '100',
            status: '1'
        }, {
            brandName: '华为',
            companyName: '华为科技有限公司',
            ordered: '100',
            status: '1'
        }
    ]
}

```

```
    }, {
      brandName: '华为',
      companyName: '华为科技有限公司',
      ordered: '100',
      status: "1"
    }, {
      brandName: '华为',
      companyName: '华为科技有限公司',
      ordered: '100',
      status: "1"
    }
  ]
}

})
</script>
</body>
</html>
```
