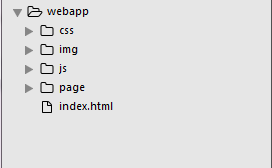
# 结构

## 目录结构

用于项目第一版进行内部评审。前端文档的目录结构按照文档的类型划分，相同类型的文档归为一类，放在同一个文件夹中统一管理。

具体划分如下：



Webapp为根目录，下有

css文件夹，盛放项目中所有的样式表（.css文件）；

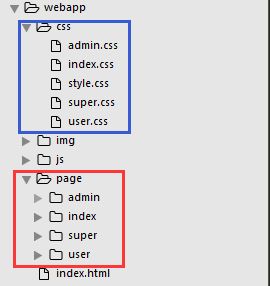
img文件夹，盛放项目中所有的图片及图标文件；

js文件夹，盛放项目中所有的JavaScript文件；

page文件夹，盛放项目中所有的页面文件（.html文件）；

index.html 为项目主页，放在根目录下。

每个文件夹下，可按照项目中功能模块再具体划分文件夹（若该模块文件<=2,则不设文件夹），如下所示：

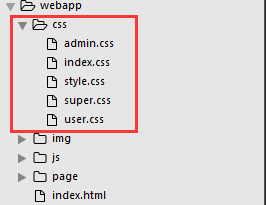


以上为基本文件目录结构，如有添加需求（如添加新的文件夹），请与团队商议后添加。

## 功能结构

每个文件夹中的文件功能，结合每个文件夹说明如下：

### CSS文件夹



style.css为项目中公共样式表，所有公共样式都在此文件中统一编写；

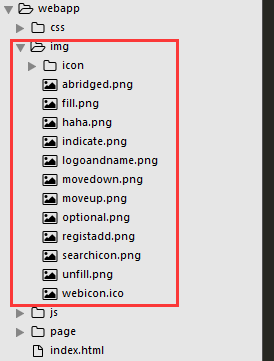
admin.css为管理员模块样式表，在此文件中编写管理员模块中非公共样式；

index.css 为首页模块样式表，在此文件中编写首页模块中样式；

super.css为超级管理员模块样式表，在此文件中编写超级管理员模块中非公共样式；

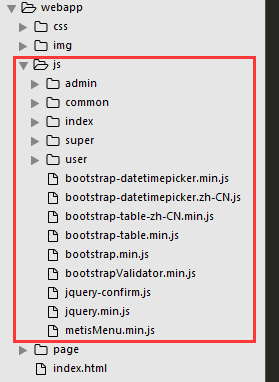
user.css为参会用户模块样式表，在此文件中编写参会用户模块中非公共样式。

### img文件夹



Img文件夹下，存放项目中用到的图片和图标，图标存放在img/icon下。

### JS文件夹

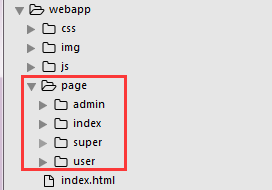


js文件夹下的.js文件，为项目中引用的资源包（第三方框架）。

Common文件夹，盛放项目中公共js方法；

Admin、index、super、user文件夹，分别盛放管理员、首页、超级管理员、参会用户模块的js方法。

### Page文件夹



Admin、index、super、user文件夹，分别盛放管理员、首页、超级管理员、参会用户模块的.html页面文件。

# 元素显示规范

## 按钮

### 显示样式

按钮（button）的显示样式，套用bootstrap框架的样式，具体代码和展示如下：

1. 确定类按钮（如保存、确定）：

“<button class="btn btn-default btn-warning ">保&nbsp存</button>”

‘btn’为必须的类名，大小为默认大小，颜色为橘红色（btn-warning）



1. 取消类按钮（如取消）：

“<button class="btn btn-default">取&nbsp消</button>”

‘btn’为必须的类名，大小为默认大小，颜色为默认颜色



1. 删除类按钮（如取消）：

“<button class="btn btn-default btn-danger">删&nbsp除</button>”

‘btn’为必须的类名，大小为默认大小，颜色为红色（btn-danger）



### 显示位置

1. 间距设置：

按钮文字 <= ‘2’的，在文字之间添加空格（如：删&nbsp除）

按钮之间间距为14px，距离上部内容间距最小为26px



1. 位置设置：

按钮组居中显示（设置父元素的text-align: center;）

## Form表单

### 显示样式

表单（form）的显示样式，套用bootstrap框架的样式，具体代码和展示如下：

<form class="form-horizontal mops-form">

<div class="form-group">

<label for="" class="col-sm-2 control-label">姓名</label>

<div class="col-sm-7">

<input type=" " class="form-control" placeholder="请输入姓名">

</div>

</div>

</form>



说明：左侧label文字，右对齐，宽度为页面2 / 12（class="col-sm-2 control-label"），

右侧input框，宽度为页面7 / 12（class="col-sm-7"）

## table表格

表格（table）的显示样式，套用bootstrap-table.min.js框架的样式，具体代码和展示如下：

1)使用前先引入以下文件

bootstrap.min.css

bootstrap-table.min.css

jquery.min.js

bootstrap.min.js

bootstrap-table.min.js

bootstrap-table-zh-CN.min.js

2) 表格用法及属性

$(function () {

//1.初始化Table

var oTable = new TableInit();

oTable.Init();

});

var TableInit = function () {

var oTableInit = new Object();

//初始化Table

oTableInit.Init = function () {

$('#texttable').bootstrapTable({

url: 'data/customer.json',

//method: 'get', //请求方式（\*）

dataType: 'json',

dataLocale: 'zh-US', //表格汉化

toolbar: '#toolbar', //工具按钮用哪个容器

striped: true, //是否显示行间隔色

cache: false,

//是否使用缓存，默认为true，所以一般情况下需要设置一下这个属性（\*）

pagination: true, //是否显示分页（\*）

sortable: false, //是否启用排序

sortOrder: "desc", //排序方式

searchable: true,

searchFormatter: false,

cellStyle: true,

checkbox : true,

queryParams: oTableInit.queryParams, //传递参数（\*）

sidePagination: "client",

//分页方式：client客户端分页，server服务端分页（\*）

pageNumber:1, //初始化加载第一页，默认第一页

pageSize: 11, //每页的记录行数（\*）

pageList: [10, 15, 20], //可供选择的每页的行数（\*）

search: true,

//是否显示表格搜索，此搜索是客户端搜索，不会进服务端

showColumns: true, //是否显示所有的列

showRefresh: true, //是否显示刷新按钮

minimumCountColumns: 2, //最少允许的列数

clickToSelect: true, //是否启用点击选中行

height: 560,

//行高，如果没有设置height属性，表格自动根据记录条数决定表格高度

uniqueId: "ID", //每一行的唯一标识，一般为主键列

showToggle:true, //是否显示详细视图和列表视图的切换按钮

cardView: false, //是否显示详细视图

detailView: false, //是否显示父子表

columns: [

{

checkbox: true

}, {

field: 'id',

title: '姓名',

align: 'center',

width: 200

},{

field: 'option',

title: '操作',

width: 150,

align: 'center', //文字居中显示

formatter: function(value,row,index){

return "<button class='btn btn-xs'>删除</button>&nbsp<button class='btn btn-xs btn-warning'>添加</button>";

//添加删除按钮

},

events: {

'click .ccc': function (e, value, row, index) {

//删除操作

alert(row.id);

},

'click .mod': function (e, value, row, index) {

//同步

alert(row.CityName);

}

}

}]

});

};

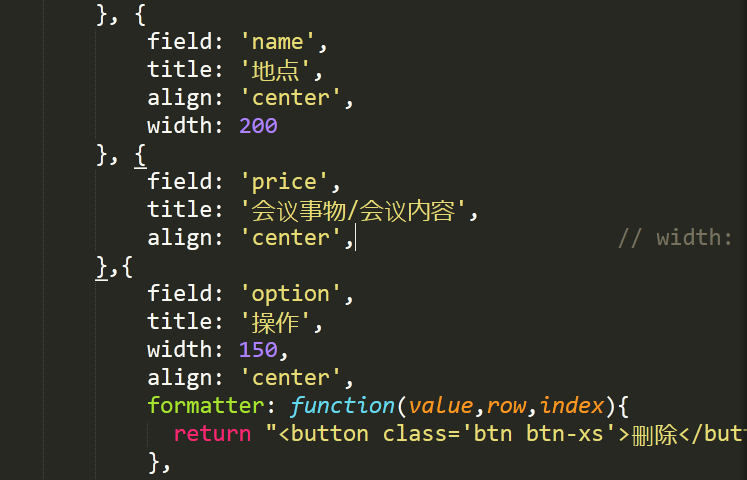
**说明：**上述代码标红部分为表格的配置参数，根据业务的不同需求，采用不同的参数设置（例如：需要排序功能 - > sortable: false, sortOrder: "desc"；分页功能 -> pagination: true 等）

显示效果如下：



3)显示位置

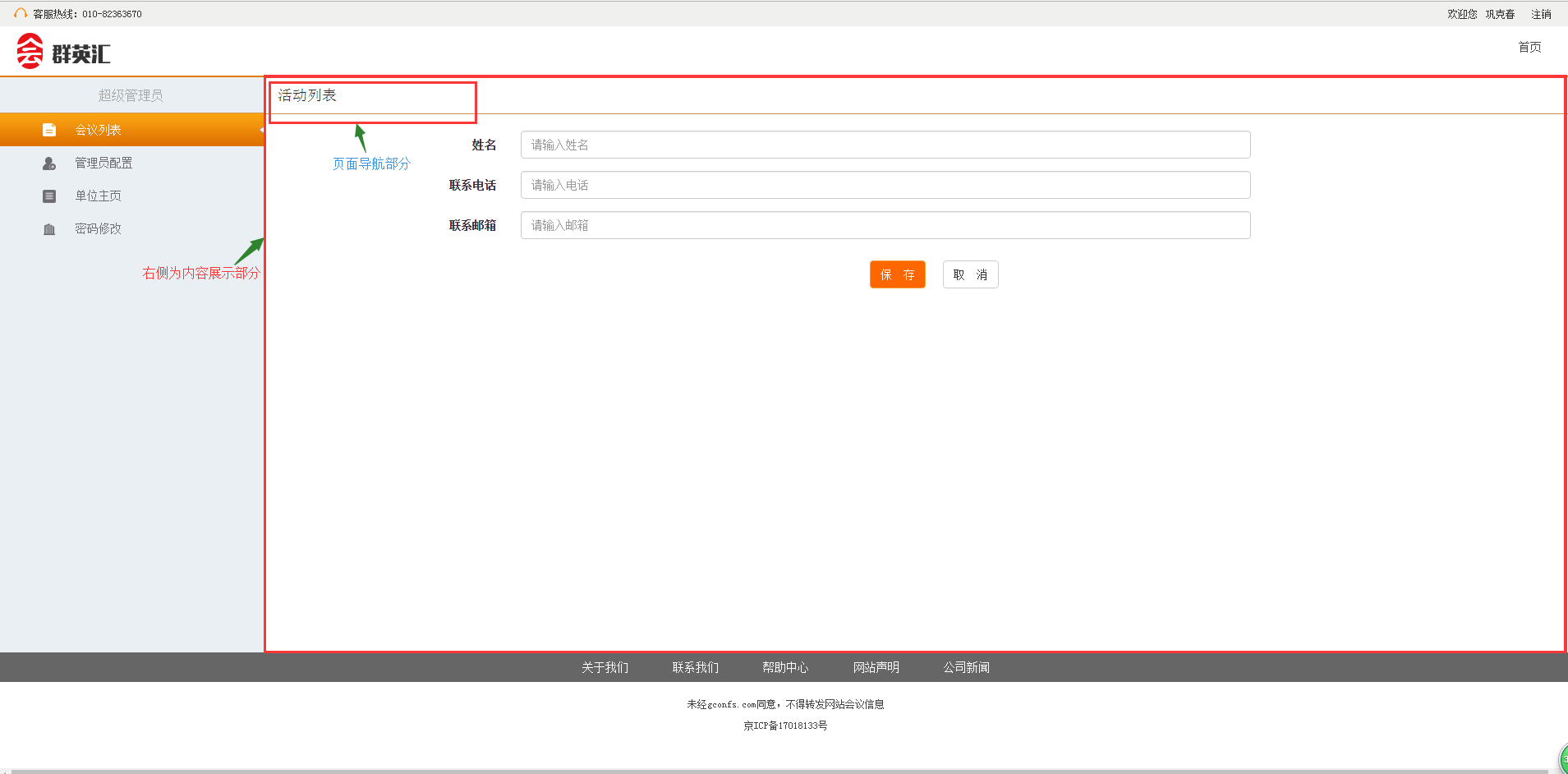
表格标题如果少于2个字(包含2个字),width设置为200;如果大于2个字则width自适应;所有文字居中显示,高度不设置,使用table默认的;字体使用公共部分确定的字体,如果标题是操作的,width设置为150,并且居中显示;如果标题都是少于2个字的,且所设置的宽度不能适应整个table,则只设置操作栏的宽度为150px,其他宽度为系统默认;



# 页面布局规范

## 含左侧带单栏页面

页头和页脚统一，用公共的元素和样式



1）中间部分的布局样式：左侧为导航菜单，右侧为内容展示部分，整体布局样式统一，用公用的样式（main.css），

2）右侧内容部分：整体布局一致，代码规范如下：

<div class=" mops-con"> 内容部分的class : mops-con (必须)

<div class="con-nav">活动列表</div> 内容部分导航头的class : con-nav (必须)

<div class="con-box"></div> 内容部分主体内容的class : con-box (必须)

</div>

.con-box 主体内容部分，外边距设置：margin：24px 3%;

3）内容导航头格式：模块功能->页面功能（或位置）->页面位置（ 例如：后台管理->活动管理->基本信息）。

4）导航头可点击，点击跳转到对应模块，代码格式如下：

<div class="con-nav">

<a href="#">后台管理</a>

<a href="#">->活动管理</a>

<a href="#">->基本信息</a>

</div>

文字颜色：默认和visited 为 黑色（#333），

Hover 和focus为蓝色（#2095F2）。

字体大小为14px, 间距默认。

1. 左侧菜单栏：根据页面功能，显示不同的内容，但是格式统一，用公用的样式（main.css），

带有二级菜单的左侧菜单栏样式如下：

样式为公用样式（main.css），效果可直接引用公共的js（main.js）中的initMenu(url)方法，参数url为菜单数据的请求地址

# ajax请求格式

## 返回数据格式

页面ajax请求后台接口，返回数据统一为json类型的字符串，由页面方法转换格式后，再进行内容展示。

转换后的json数据，格式如下：

{

flag : true/false, // true : 返回结果有数据，false : 返回结果无数据

msg : (flag为false 的)提示信息，

url : 其他提示信息（字段待定） ,

data : value(后台返回的页面数据)

}

# SESSION使用规范

为防止命名冲突，导致数据覆盖，现在作一下规范：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | KEY名字 | 存储的数据说明 | 使用的页面 | 备注 |
| 1 | actId | 活动id值 | 管理员点击活动后的所有页面 |  |
| 2 | venueId | 会场id |  |  |
| 3 | loginInfo | 用户登录信息 |  |  |
| 4 | ResId | 同程酒店ID |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |

# 时间格式规范

1、为统一方便存储及后期分析使用所有时间存储为时间戳：

如：2017-06-26 12:00:00 存储为-> 1498449600000

( timeStamp('2017-06-26 12:00:00') 方法转时间戳 )

2、页面展示格式为 yyyy-MM-dd hh:mm:ss (2017-06-26 12:00:00)