js模块的api v1.3

[js模块的api 1](#_Toc420440269)

[1 表格table（mdtable）/init 1](#_Toc420440270)

[1.1 引入： 1](#_Toc420440271)

[1.2 class选择器监听事件 2](#_Toc420440272)

[2 对话框模块dialog/init 9](#_Toc420440273)

[2.1 引入 9](#_Toc420440274)

[2.2 class选择器监听事件 9](#_Toc420440275)

[3 表单模块form/init 10](#_Toc420440276)

[3.1 引入 10](#_Toc420440277)

[3.2 class选择器监听事件 11](#_Toc420440278)

[4 上传模块upload/init 14](#_Toc420440279)

[4.1 引入 14](#_Toc420440280)

[4.2 class选择器监听事件 15](#_Toc420440281)

# 1 表格table（mdtable）/init

## 1.1 引入：

在顶部添加上：<#include "../../common/\_js/table\_js.ftl" />

require(["modules/table/init"], function(Table){

    $(document).ready(function(){

        Table.init();

    });

});

或

在顶部添加上：<#include "../../common/\_js/model\_table\_js.ftl" />

require(["modules/mdtable/init"], function(MdTable){

    $(document).ready(function(){

        MdTable.init();

    });

});

## 1.2 class选择器监听事件

### 1.2.1分页

<div class="table-responsive">标签加上"paging-table"的class即可；

### 1.2.2 排序

<div class="table-responsive">标签加上"sort-table"的class，并在需要排序的列的th标签添加class="sorting"和key="此列Freemarker的key"；

eg:

<div class="table-responsive paging-table sort-table">

    <input type="hidden" name="path" value="admin/member/list/" /><!-- 分页、排序、查询用到的path，必须加上 -->

    <table class="table table-striped table-bordered table-hover"><!-- 统一风格写法 -->

        <thead>

            <tr>

                <th width="20px">

                    <div class="checker">

                        <span>

                            <input type="checkbox" class="group-checkable" data-set=".table .checkboxes" />

                        </span>

                    </div>

                </th>

                <th>用户名</th>

                <th class="sorting" key="meinIntegration">积分</th>

                <th class="sorting" key="m1.mein\_create\_time">创建时间</th>

                <th>操作</th>

            </tr>

        </thead>

        <tbody>

            <#list list as entity >

                <tr>

                    <td>

                        <div class="checker">

                            <span>

                                <input type="checkbox" class="checkboxes" value="${entity.meinId!}">

                            </span>

                        </div>

                    </td>

                    <td>${entity.meinWechatNickname!}</td>

                    <td>${entity.meinIntegration!}</td>

                    <td>${entity.meinCreateTime!}</td>

                    <td>

                        <a href="javascript:;" confirm="确定删除？" url="admin/member/delete/"

                            title="删除" class="op-btnbootbox-confirm">

                            <i class="icon-trash"></i>

                        </a>

                    </td>

                </tr>

            </#list>

        </tbody>

    </table>

</div>

### 1.2.3 查询区域

在table上方加上<div class="form-inline">，并在其中写<form class="query-form">查询表单。

<form class="query-form">表单项规则如下：

eg:

<div class="form-inline">

    <form class="query-form">

        <#include "../\_template/pagination\_control.ftl" /><!-- 引入每页显示多少条的模板文件 -->

        div class="form-group">

            <label class="control-label">用户名：</label>

            <input name="meinWechatNickname" operator="like" data-type="string" class="form-control input-small" />

        </div>

<div class="form-group form-right">

<button type="button" class="btnbtn-info btn-sm reset">清空条件</button>

<button type="button" class="btnbtn-info btn-sm query">查询</button>

</div>

    </form>

</div>

其中operator为查询运算符，取值范围包括：

"eq"代表"="；

"gt"代表">"；

"ge"代表">="；

"lt"代表"<"；

"le"代表"<="；

"like"代表模糊查询。

其中data-type为查询字段类型，取值范围包括：

"string"代表字符串；

"number"代表数值；

"date"代表日期。

其中清空条件按钮加上"reset"的class，查询按钮加上"query"的class

### 1.2.4 批量操作

在portlet-title中的action区域加上下拉按钮，操作项a标签加上url属性、"bootbox-batch-confirm"的class和confirm="提示文本"的属性

eg:

<div class="portlet-title">

    <div class="caption">

        <i class="icon-list-alt"></i>用户列表

    </div>

    <div class="actions">

        <div class="btn-group">

            <a class="btnbtn-success dropdown-toggle" href="javascript:;" data-toggle="dropdown">

                批量操作

                <i class="icon-angle-down"></i>

            </a>

            <ul class="dropdown-menu pull-right">

                <li>

                    <a url="admin/member/batchDelete" confirm="确定要删除吗？" class="bootbox-batch-confirm" href="javascript:;">

                        <i class="icon-trash"></i>

                        删除

                    </a>

                </li>

            </ul>

        </div>

    </div>

</div>

批量操作需要加上表头复选框和行内复选框：

* 表头复选

class="group-checkable",加上data-set=".table .checkboxes":

<th width="20px">

    <div class="checker">

        <span>

            <input type="checkbox" class="group-checkable" data-set=".table .checkboxes" />

        </span>

    </div>

</th>

* 行内复选框class="checkboxes"：

<td>

    <div class="checker">

        <span>

            <input type="checkbox" class="checkboxes" value="${entity.meinId!}">

        </span>

    </div>

</td>

### 1.2.5 其他

* 表格的复选框列和操作列使用固定宽度，其他列宽使用百分比来进行调整。
* 查询表单有关范围的都加上from和to样式。
* 操作列的按钮统一使用icon图标+op-btn样式。

# 2 对话框模块dialog/init

## 2.1 引入

require(["modules/dialog/init"], function(Dialog){

    $(document).ready(function(){

        Dialog.init();

    });

});

## 2.2class选择器监听事件

### 2.2.1 modal

我们使用的modal统一都是ajax请求的modal：在触发modal的按钮加上url属性指定请求路径，加上"modal-ajax"或"modal-ajax-static"的class；"modal-ajax"生成的modal点击遮罩层会关闭，一般使用在查看详情等情况；"modal-ajax-static"生成的modal点击遮罩层不会关闭，一般使用在填写表单或进行一些复杂操作的情况。

### 2.2.2 bootbox

bootbox是触发动作前的确认操作，只要在触发动作的按钮加上url属性指定请求路径，加上"bootbox-confirm"的class和confirm="提示文本"的属性；点击确认则进行url的请求，取消则关闭对话框不做任何操作。

# 3 表单模块form/init

## 3.1 引入

require(["modules/form/init"], function(Form){

    $(document).ready(function(){

        Form.init();

    });

});

## 3.2class选择器监听事件

### 3.2.1 表单提交

我们使用的表单统一都是使用ajax提交的，所以首先要在form标签中加上"ajax-form"的class：表单进行提交只要为提交按钮加上"ajax-submit"（page里面的）或"ajax-submit-modal"（modal里面的）的class和url属性即可；

### 3.2.2 表单确认提交

如需表单提交前进行确认操作，则为提交按钮加上"ajax-submit-confirm"（page里面的）或"ajax-submit-modal-confirm"（modal里面的）的class、url属性和confirm属性即可；

### 3.2.3 表单验证

如果表单需要提交前验证输入信息，则在form标签中加上"valid"的class，表单的验证规则可以直接以属性形式写在输入标签中，验证规则可参照jquery-validation的默认规则，required、equalTo、maxlength、minlength、rangelength、max、min、range使用属性键值对方式，email、url、date、number、digits、creditcard等格式类型使用type=""方式替换原来input的type属性。另外，我们额外添加了phone="true"的验证手机号码的规则（待补充）。参考：<http://www.w3cschool.cc/jquery/jquery-plugin-validate.html>



### 3.2.4 表单提示

如需为表单输入信息添加提示，只需在输入标签的上一级div中加上tip属性指定提示文本和tip-length属性（一般我们都统一“345”的风格，省略则默认为col-md-5）指定提示文本所占的宽度

eg:  
<div class="portlet-body form">

    <form class="form-horizontal ajax-form valid">

        <div class="form-body">

            <div class="form-group">

                <label class="control-label col-md-3">基本积分<span class="required">\*</span></label>

                <div class="col-md-4" tip="注：基本积分指每个用户账号注册后的初始积分" tip-length="col-md-5"><!-- 此处tip-length可省略 -->

                    <input name="integration" value="${integration}" type="text" class="form-control placeholder="请输入100-1000之前的数值"  required="true" min="100" max="10000" />

                </div>

            </div>

        </div>

        <div class="form-actions fluid">

            <div class="col-md-offset-3 col-md-9">

                <button type="button" class="btnbtn-success ajax-submit" url="admin/integration/save">保存</button>

            </div>

        </div>

    </form>

</div>

# 4上传模块upload/init

## 4.1 引入

require(["modules/upload/init"], function(Upload){

    $(document).ready(function(){

       Upload.init();

    });

});

## 4.2 class选择器监听事件

### 4.2.1 图片上传

图片上传采用统一的格式，在<a>或<button>标签加上class="plupload\_image"即可初始化文件上传按钮，同时需加上data-name代表后台接受上传图片url数组；如果图片需要压缩，则可以在<a>或<button>标签加上data-width、data-height属性来压缩图片的宽、高。

需要引入一个样式：<link href="${base}/assets/plugins/plupload/uploadify.min.css" rel="stylesheet" type="text/css"/>

例子：

<!-- start image upload -->

<div>

<a href="javascript:;" class="plupload-imagebtnbtn-default" data-width="500" data-height="500" data-name="images0">商品图片</a>

</div>

<#include "../../common/\_template/show\_image.ftl" />

<!-- end image upload -->

或者

<!-- start image upload -->

<div>

<a href="javascript:;" class="plupload-image btnbtn-default" data-name="images1">上传图片</a>

</div>

<#include "../../common/\_template/show\_image.ftl" />

<!-- end image upload -->

### 4.2.2 一般文件上传

文件上传采用统一的格式，在<a>或<button>标签加上class="plupload\_file"即可初始化文件上传按钮；同时需加上data-name代表后台接受上传文件url数组；

例子：

<!-- start file upload -->

<div>

<a href="javascript:;" class="plupload-filebtnbtn-default" data-name="files0">文件上传0</a>

</div>

<#include "../../common/\_template/show\_file.ftl" />

<!-- end file upload -->