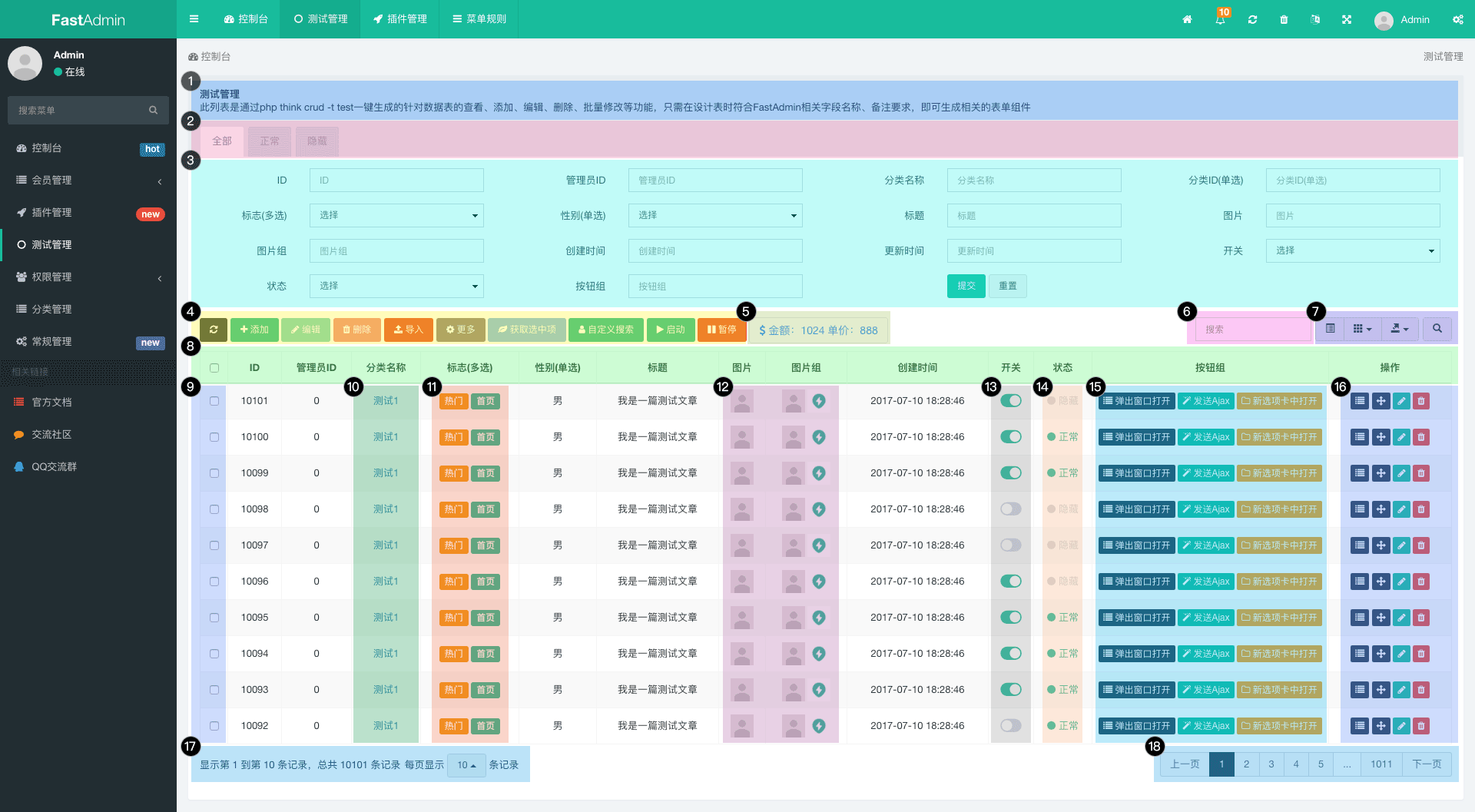
[PHP]\_TP5.0\_FastAdmin\_图解FastAdmin表格列表的功能；

--王尔贝(我是害虫)

(图 https://cdn.forum.fastadmin.net/uploads/201809/15/a6309f467fcade6c47d226f2890ebe29)



--

--

**一、功能描述**

(请根据图片上的数字索引查看对应功能说明)

--

**1.菜单名称和描述**

默认生成的CRUD是没有菜单名称和描述显示的，如果需要显示则可以修改权限管理->菜单规则，给对应菜单的添加上备注信息后即可显示，支持HTML。

--

**2.TAB过滤选项卡**

在一键生成CRUD时，如果表中存在status字段且为ENUM类型，则会生成相应的TAB过滤选项卡，如果需要生成其它字段的过滤选项卡则可以在使用php think crud时使用--headingfilterfield=你的字段名称来指定字段。

--

**3.通用搜索**

通用搜索的内容是根据bootstrap-table配置的字段columns决定的，渲染的内容及格式由FastAdmin自动进行渲染，如果需要禁用或删除某一选项，可以在JS中配置operate:false来删除通用搜索中的选项。例如通常情况下我们的在JS中进行字段的配置如“{field: 'createtime', title: \_\_('Create Time')},”。

**(1)、通用搜索常用配置**

这里默认是启用的通用搜索，针对通用搜索，有以下几个常用的配置。

//// 常用配置

operate: '=' //用于查询时的操作符，默认为=，为false表示禁用此字段的通用搜索，支持!=、LIKE、NOT LIKE、>、<、>=、<=、FIND\_IN\_SET、IN、NOT IN、BETWEEN、NOT BETWEEN、RANGE、NOT RANGE、NULL、NOT NULL、false；

searchList: //用于渲染列表项的数据，支持的格式有JSON Array、JSON Object、$.getJSON、Function；

addclass: //用于给input或select添加额外的class属性；

type: //用于定义input文本框的类型,默认为text；

data: //用于给input或select添加额外的属性；

**(2)、常用配置示例**

// 时间区间搜索

{field: 'createtime', title: \_\_('Createtime'), operate: 'RANGE', addclass: 'datetimerange', formatter: Table.api.formatter.datetime},

// 金额区间搜索

{field: 'money', title: \_\_('Money'), operate: 'RANGE'},

// 下拉列表搜索

{field: 'flag', title: \_\_('Flag'), searchList: {"hot": \_\_('Flag hot'), "index": \_\_('Flag index'), "recommend": \_\_('Flag recommend')}, operate: 'FIND\_IN\_SET', formatter: Table.api.formatter.label},

// 动态下拉列表搜索

{field: 'type', title: \_\_('Type'), searchList: $.getJSON("ajax/dynamicselect")},

// 禁用通用搜索

{field: 'keywords', title: \_\_('Keywords'), operate: false},

//

如果我们需要完全自定义我们的通用搜索栏，我们可以在配置bootstrap-table时定义searchFormTemplate选项来完全重写我们的通用搜索栏，具体请参考开发示例插件中的自定义搜索示例。

--

**4.工具栏按**钮

FastAdmin在一键CRUD时会自动生成添加、编辑、删除、导入、更多按钮的HTML，这些按钮会根据用户所拥有的权限来控制是否显示或隐藏。我们可以在控制器对应的index.html视图文件中任意修改或删除对应的按钮。请特别注意这几个自动生成的按钮都通过拥有的class属性来绑定相关的事件，例如添加按钮拥有btn-add这个class、框架所已经占用的class如下。

**(1)、已被占用class**

btn-add: 添加按钮使用；

btn-edit: 编辑按钮使用；

btn-del: 删除按钮使用；

btn-import: 导入按钮使用；

btn-more: 更多按钮使用；

btn-multi: 指操作使用；

btn-disabled: 添加此class后则只有在列表有选中数据时按钮才会变为可使用；

//

如果我们想点击添加按钮后默认全屏，则可以给添加按钮加上data-area='["100%","100%"]'即可默认全屏。如果我们想自定义按钮并添加事件，我们需要在视图中添加相应的HTML代码，然后在对应的JS文件中添加按钮的执行事件，切记不可在视图中直接编写JS或jQuery代码来绑定事件。

--

**5.动态渲染统计信息**

很多时候我们需要在页面额外显示服务端传回的动态数据，此时我们只需要在index.html视图中添加如下信息。

<a href="javascript:;" class="btn btn-default" style="font-size:14px; color:dodgerblue;">

<i class="fa fa-dollar"></i>

<span class="extend">

金额：<span id="money">0</span>

单价：<span id="price">0</span>

</span>

</a>

//

//// 然后在控制器对应的JS中的index方法中添加以下的JS。

// 当表格数据加载完成时

table.on('load-success.bs.table', function (e, data) {

// 这里可以获取从服务端获取的JSON数据

console.log(data);

// 这里我们手动设置底部的值

$("#money").text(data.extend.money);

$("#price").text(data.extend.price);

});

//

//// 注意务必将这段代码添加在“var table = $("#table");”之后，其中data.extend.money和data.extend.price就是我们在服务端动态返回的数据，如下。

$result = array("total" => $total, "rows" => $list, "extend" => ['money' => 1024, 'price' => 888]);

return json($result);

//

通过以上配置即可动态显示服务端返回的额外数据。

--

**6.快速搜索**

快速搜索在键入关键词时将实时从服务端搜索数据，如果你的数据表数据较大，建议关闭此功能，关闭的方法是使用search:false，其次快速搜索默认只会搜索主键id这个字段，如果你需要搜索其它字段，则需要在服务端你的控制器中定义$searchFields这个值，如“protected $searchFields = 'id,name,title';//快速搜索字段;”。这样在快速搜索时将会搜索“id,name,title”这三个字段。如果需要修改默认文本框的placeholder，可以在表格初始化前定义如下。

$.fn.bootstrapTable.locales[Table.defaults.locale]['formatSearch'] = function(){return "自定义placeholder文本";};

--

**7.浏览模式、显示隐藏列、导出、通用搜索**

**浏览模式**可以切换卡片视图和表格视图两种模式，如果不需要此功能，可以设置“showToggle: false”。

**显示隐藏列**可以快速切换字段列的显示和隐藏，如果不需要此功能，可以设置“showColumns: false”，如果想要表格中的字段列默认隐藏可以设置字段属性“visible: false”即可默认隐藏

**导出按钮**默认将导出整个表的所有行，如果需要仅导出当前分页的数据，需要设置“exportDataType: 'basic'”，如果想导出选中的行，则可以设置为“exportDataType: 'selected'”，如果不需要此功能，可以设置“showExport: false”。

**通用搜索**指表格上方的搜索，通用搜索的表单默认是隐藏的，如果需要默认显示，需要设置"searchFormVisible: true"，如果不需要通用搜索功能，可以设置"commonSearch: false"。如果想要控制字段列不参与搜索则可以设置字段列属性为"operate: false"即可。

--

**8.字段配置**

默认字段的控制是根据控制器对应的JS来配置的，因此字段配置是通过JS，而在我们的视图index.html中是没有任何字段配置的，通常我们的配置如下。

columns: [

[

{checkbox: true},

{field: 'id', title: \_\_('Id')},

{field: 'admin\_id', title: \_\_('Admin\_id')},

{field: 'category.name', title: \_\_('分类名称'), formatter:Table.api.formatter.search},

{field: 'category\_id', title: \_\_('Category\_id'), visible: false},

{field: 'flag', title: \_\_('Flag'), searchList: {"hot": \_\_('Flag hot'), "index": \_\_('Flag index'), "recommend": \_\_('Flag recommend')}, operate: 'FIND\_IN\_SET', formatter: Table.api.formatter.label},

{field: 'genderdata', title: \_\_('Genderdata'), searchList: {"male": \_\_('Genderdata male'), "female": \_\_('Genderdata female')}, formatter: Table.api.formatter.normal},

{field: 'title', title: \_\_('Title')},

{field: 'image', title: \_\_('Image'), formatter: Table.api.formatter.image},

{field: 'images', title: \_\_('Images'), formatter: Table.api.formatter.images},

{field: 'createtime', title: \_\_('Createtime'), operate: 'RANGE', addclass: 'datetimerange', formatter: Table.api.formatter.datetime},

{field: 'updatetime', title: \_\_('Updatetime'), operate: 'RANGE', addclass: 'datetimerange', formatter: Table.api.formatter.datetime, visible: false},

{field: 'weigh', title: \_\_('Weigh'), operate: false, visible: false},

{field: 'switch', title: \_\_('Switch'), searchList: {"1": \_\_('Yes'), "0": \_\_('No')}, formatter: Table.api.formatter.toggle},

{field: 'status', title: \_\_('Status'), searchList: {"normal": \_\_('Normal'), "hidden": \_\_('Hidden')}, formatter: Table.api.formatter.status},

]

]

//

**(1)、字段配置参数**

checkbox:true, //是否为首列复选框;

field:'name' //字段名称,如果启用了关联查询这里可以使用别名,比如:category.name,请注意服务端返回的字段一一对应，如果使用了一个不存在的字段，将不会渲染任何数据;

title:'名称' //字段标题,显示于头部的标题;

operate:'=' //通用搜索的操作符,详见上方通用搜索介绍;

visible:false //字段是否可见，为false时将默认不可见;

formatter:Table.api.formatter.search //格式化显示的内容,FastAdmin内部定义了许多通用的格式化方法;

events: //定义元素响应的事件;

searchList: //定义通用搜索下拉列表的数据;

addclass: //通用搜索文本框或下拉列表的额外class;

type: //通用搜索文本框的类型;

data: //通用搜索文本框或下拉列表的额外属性;

//

//// FastAdmin封装了许多常用的formatter方法，我们可以快速的调用即可。

> `Table.api.formatter.icon` 快速将字段渲染成一个按钮，仅支持Fontawesome按钮;

> `Table.api.formatter.image` 快速将字段渲染成图片展示的形式;

> `Table.api.formatter.images` 快速将字段渲染成多图片展示的形式,字段数据请以`,`进行分隔;

> `Table.api.formatter.status` 快速将字段渲染成状态，默认`normal/hidden/deleted/locked`这四个状态;

> `Table.api.formatter.url` 快速将字段渲染成URL框;

> `Table.api.formatter.search` 快速将字段渲染成可搜索的链接，点击后将执行搜索;

> `Table.api.formatter.addtabs` 快速将字段渲染成可添加到选项卡的链接，点击后将把链接添加到选项卡;

> `Table.api.formatter.flag` 快速将字段渲染成标志，仅支持`index/hot/recommend/new`这四种标志;

> `Table.api.formatter.label` 快速将字段渲染Label标签;

> `Table.api.formatter.datetime` 快速时间戳数据渲染成日期时间数据;

> `Table.api.formatter.operate` 操作栏固定按钮;

> `Table.api.formatter.buttons` 快速生成多个按钮;

> `Table.api.formatter.toggle` 快速生成切换按钮;

//

--

**9.复选框**

如果我们需要不需要复选框则移除{checkbox: true}即可。

--

**10.分类名称(关联搜索出分类表的名称)**

这里显示的分类名称是根据分类表关联查询出来的结果，如果我们启用关联查询，我们必须在当前控制器中设置属性"protected $relationSearch = true;"，同时我们的index方法也需要重写，请参考下方的完整代码中PHP部分。如果我们启用了关联查询，当两个表中的字段有冲突时，我们必须在字段中加上别名。请参考下方的完整代码中JS部分。

--

**11.标志**

我们可以使用"formatter:Table.api.formatter.flag"来渲染标志字段，默认会将数据库的值渲染以下几种颜色，{index: 'success', hot: 'warning', recommend: 'danger', 'new': 'info'}。如果我们需要扩展额外的颜色，则可以使用“custom:{aaa: 'info', bbb:'danger'}”，这样当值为aaa时会显示为蓝色，bbb时显示为红色。

--

**12.图片和图片组**

我们可以使用“formatter:Table.api.formatter.image”、“formatter:Table.api.formatter.images”。以上两种方式来渲染图片或图片组，请注意如果是图片组的情况下数据值应该是以“,”进行分隔的。

--

**13.开关**

我们可以使用“formatter:Table.api.formatter.toggle”来生成开关组件，默认情况下是根据数据库值1和0来表示开和关，我们可以通过额外的配置和定义开和关，比如“yes: 'open', no: 'close'”，则此时会根据数据库值是open还是close来展示开关，开关在点击的时候默认是只允许修改数据库的status字段的，如果我们开关不是status字段，我们需要在服务端对应的控制器中定义‘protected $multiFields="id,name,swith";’，多个字段以“,”进行分隔。

--

**14.状态渲染**

我们可以使用“formatter:Table.api.formatter.status”来渲染状态，默认根据以下值进行状态的颜色渲染，{normal: 'success', hidden: 'gray', deleted: 'danger', locked: 'info'}。如果我们状态有额外的值，我们可以使用custom来进行扩展显示的颜色，如下“custom: {rejected:'danger', agreed:'success'}”，状态渲染显示的文本是根据searchList配置的值进行渲染的。

--

**15.自定义按钮**

按钮组的功能是根据第8项中的Table.api.formatter.buttons进行生成的，代码如下。

{

field: 'buttons',

width: "120px",

title: \_\_('按钮组'),

table: table,

events: Table.api.events.operate,

buttons: [

{

name: 'detail',

text: \_\_('弹出窗口打开'),

title: \_\_('弹出窗口打开'),

classname: 'btn btn-xs btn-primary btn-dialog',

icon: 'fa fa-list',

url: 'example/bootstraptable/detail',

callback: function (data) {

Layer.alert("接收到回传数据：" + JSON.stringify(data), {title: "回传数据"});

},

visible: function (row) {

//返回true时按钮显示,返回false隐藏

return true;

}

},

{

name: 'ajax',

text: \_\_('发送Ajax'),

title: \_\_('发送Ajax'),

classname: 'btn btn-xs btn-success btn-magic btn-ajax',

icon: 'fa fa-magic',

url: 'example/bootstraptable/detail',

confirm: '确认发送',

success: function (data, ret) {

Layer.alert(ret.msg + ",返回数据：" + JSON.stringify(data));

//如果需要阻止成功提示，则必须使用return false;

//return false;

},

error: function (data, ret) {

console.log(data, ret);

Layer.alert(ret.msg);

return false;

}

},

{

name: 'addtabs',

text: \_\_('新选项卡中打开'),

title: \_\_('新选项卡中打开'),

classname: 'btn btn-xs btn-warning btn-addtabs',

icon: 'fa fa-folder-o',

url: 'example/bootstraptable/detail'

}

],

formatter: Table.api.formatter.buttons

}

//

//// 按钮配置支持的参数

name 按钮唯一标识,其中add/edit/del/dragsort已经被占用,请勿使用。我们可以在HTML视图文件的table使用data-buttons-标识来控制显示。

text 按钮的文本内容,如果不需要显示文本可忽略。

title 鼠标移上去的标题或弹窗/选项显示的标题。

icon 按钮的图标,请使用font-awesome图标库。

classname 按钮的class, 其中classname中的btn-dialog、btn-ajax、btn-addtabs、btn-click，FastAdmin已经为这几个固定的Class注册了事件，所以可以直接使用，如果想要实现其它功能，需要自己手动编写代码绑定事件才可使用。

url 按钮的链接/Ajax事件请求的URL/弹窗链接/选项卡链接，直接function和string类型,此链接会自动在链接后添加ids/{ids},{ids}为当行主键ID，如果需要传递其它字段值，请在URL中使用{字段名}占位即可。

refresh 自动刷新,只针对btn-ajax事件。

confirm 确认框提示文字,配置后会在确认操作再执行对应的事件，只针对btn-ajax/btn-dialog/btn-addtabs事件。

success 事件成功的回调,只针对btn-ajax事件。

error 事件失败的回调,只针对btn-ajax事件。

callback 弹窗回传的回调,只针对btn-dialog事件,需要在对应打开的页面中使用“Fast.api.close(data);”进行回传数据。

hidden 是否隐藏按钮,按钮默认显示,支持function和bool类型。

visible 是否显示按钮,按钮默认显示,支持function和bool类型。

click 当classname包含btn-click时的点击回调事件。

extend 按钮扩展信息,可以任意定制按钮的参数,比如我们想在新窗口中打开链接，则配置extend:' target="\_blank"'即可。

--

**16.操作**

操作区域默认是排序、编辑、删除这三个按钮，此功能也是根据第8项中Table.api.formatter.operate来实现的。排序按钮只在表中存在weigh字段时才会出现，编辑按钮和删除按钮会根据管理员所拥有的权限进行按需显示。如果我们需要重写编辑(排序、删除)按钮的相关属性，则可以使用buttons来定义编辑(排序、删除)的相关属性。其次Table.api.formatter.operate也支持buttons属性来配置多个其它按钮，如示例图中的详情按钮。请参考下方完整代码中JS部分。

--

**17.分页信息**

分页信息显示的文字可以通过在表格初始化前定义。

$.fn.bootstrapTable.locales[Table.defaults.locale]['formatSearch'] = function (pageFrom, pageTo, totalRows) {

return '显示第 ' + pageFrom + ' 到第 ' + pageTo + ' 条记录，总共 ' + totalRows + ' 条记录';

};

分页大小或分页大小选项可以在表格初始化时传入以下参数进行配置，“pageSize: 10, pageList: [10, 25, 50, 'All'],”。

--

**18.翻页信息**

翻页信息会根据服务端返回的数据行数自动进行渲染，如果返回的行数不满足分页条件时，此项是不会显示的。

--

--

**二、完整代码**

--

**1、PHP代码**

<?php

namespace app\admin\controller;

use app\common\controller\Backend;

/\*\*

\* 测试管理

\*

\* @icon fa fa-circle-o

\* @remark 此列表是通过php think crud -t test一键生成的针对数据表的查看、添加、编辑、删除、批量修改等功能，只需在设计表时符合FastAdmin相关字段名称、备注要求，即可生成相关的表单组件。

\*/

class Test extends Backend

{

// 模型

protected $model = null;

// 是否关联查询

protected $relationSearch = true;

// 初始化

public function \_initialize()

{

parent::\_initialize();

$this->model = model('Test');

}

/\*\*

\* 查看

\*/

public function index()

{

if ($this->request->isAjax())

{

list($where, $sort, $order, $offset, $limit) = $this->buildparams();

$total = $this->model

->with("category")

->where($where)

->order($sort, $order)

->count();

$list = $this->model

->with("category")

->where($where)

->order($sort, $order)

->limit($offset, $limit)

->select();

$result = array("total" => $total, "rows" => $list, "extend" => ['money' => 1024, 'price' => 888]);

return json($result);

}

return $this->view->fetch();

}

}1、PHP代码

<?php

namespace app\admin\controller;

use app\common\controller\Backend;

/\*\*

\* 测试管理

\*

\* @icon fa fa-circle-o

\* @remark 此列表是通过php think crud -t test一键生成的针对数据表的查看、添加、编辑、删除、批量修改等功能，只需在设计表时符合FastAdmin相关字段名称、备注要求，即可生成相关的表单组件。

\*/

class Test extends Backend

{

// 模型

protected $model = null;

// 是否关联查询

protected $relationSearch = true;

// 初始化

public function \_initialize()

{

parent::\_initialize();

$this->model = model('Test');

}

/\*\*

\* 查看

\*/

public function index()

{

if ($this->request->isAjax())

{

list($where, $sort, $order, $offset, $limit) = $this->buildparams();

$total = $this->model

->with("category")

->where($where)

->order($sort, $order)

->count();

$list = $this->model

->with("category")

->where($where)

->order($sort, $order)

->limit($offset, $limit)

->select();

$result = array("total" => $total, "rows" => $list, "extend" => ['money' => 1024, 'price' => 888]);

return json($result);

}

return $this->view->fetch();

}

}

--

**2、JS代码**

define(['jquery', 'bootstrap', 'backend', 'table', 'form'], function ($, undefined, Backend, Table, Form) {

var Controller = {

index: function () {

// 初始化表格参数配置

Table.api.init({

extend: {

index\_url: 'test/index',

add\_url: 'test/add',

edit\_url: 'test/edit',

del\_url: 'test/del',

multi\_url: 'test/multi',

table: 'test',

}

});

var table = $("#table");

//当表格数据加载完成时

table.on('load-success.bs.table', function (e, data) {

//这里可以获取从服务端获取的JSON数据

console.log(data);

//这里我们手动设置底部的值

$("#money").text(data.extend.money);

$("#price").text(data.extend.price);

});

// 初始化表格

table.bootstrapTable({

url: $.fn.bootstrapTable.defaults.extend.index\_url,

pk: 'id',

sortName: 'weigh',

columns: [

[

{checkbox: true},

{field: 'id', title: \_\_('Id')},

{field: 'admin\_id', title: \_\_('Admin\_id')},

{field: 'category.name', title: \_\_('分类名称'), formatter:Table.api.formatter.search},

{field: 'category\_id', title: \_\_('Category\_id'), visible: false},

{field: 'flag', title: \_\_('Flag'), searchList: {"hot": \_\_('Flag hot'), "index": \_\_('Flag index'), "recommend": \_\_('Flag recommend')}, operate: 'FIND\_IN\_SET', formatter: Table.api.formatter.label},

{field: 'genderdata', title: \_\_('Genderdata'), searchList: {"male": \_\_('Genderdata male'), "female": \_\_('Genderdata female')}, formatter: Table.api.formatter.normal},

{field: 'title', title: \_\_('Title')},

{field: 'image', title: \_\_('Image'), formatter: Table.api.formatter.image},

{field: 'images', title: \_\_('Images'), formatter: Table.api.formatter.images},

{field: 'createtime', title: \_\_('Createtime'), operate: 'RANGE', addclass: 'datetimerange', formatter: Table.api.formatter.datetime},

{field: 'updatetime', title: \_\_('Updatetime'), operate: 'RANGE', addclass: 'datetimerange', formatter: Table.api.formatter.datetime, visible: false},

{field: 'weigh', title: \_\_('Weigh'), operate: false, visible: false},

{field: 'switch', title: \_\_('Switch'), searchList: {"1": \_\_('Yes'), "0": \_\_('No')}, formatter: Table.api.formatter.toggle},

{field: 'status', title: \_\_('Status'), searchList: {"normal": \_\_('Normal'), "hidden": \_\_('Hidden')}, formatter: Table.api.formatter.status},

{

field: 'buttons',

width: "120px",

title: \_\_('按钮组'),

table: table,

events: Table.api.events.operate,

buttons: [

{

name: 'detail',

text: \_\_('弹出窗口打开'),

title: \_\_('弹出窗口打开'),

classname: 'btn btn-xs btn-primary btn-dialog',

icon: 'fa fa-list',

url: 'example/bootstraptable/detail',

callback: function (data) {

Layer.alert("接收到回传数据：" + JSON.stringify(data), {title: "回传数据"});

},

visible: function (row) {

//返回true时按钮显示,返回false隐藏

return true;

}

},

{

name: 'ajax',

text: \_\_('发送Ajax'),

title: \_\_('发送Ajax'),

classname: 'btn btn-xs btn-success btn-magic btn-ajax',

icon: 'fa fa-magic',

url: 'example/bootstraptable/detail',

confirm: '确认发送',

success: function (data, ret) {

Layer.alert(ret.msg + ",返回数据：" + JSON.stringify(data));

//如果需要阻止成功提示，则必须使用return false;

//return false;

},

error: function (data, ret) {

console.log(data, ret);

Layer.alert(ret.msg);

return false;

}

},

{

name: 'addtabs',

text: \_\_('新选项卡中打开'),

title: \_\_('新选项卡中打开'),

classname: 'btn btn-xs btn-warning btn-addtabs',

icon: 'fa fa-folder-o',

url: 'example/bootstraptable/detail'

}

],

formatter: Table.api.formatter.buttons

},

{

field: 'operate', title: \_\_('Operate'), table: table, events: Table.api.events.operate,

buttons: [

{

name: 'detail',

title: \_\_('详情'),

classname: 'btn btn-xs btn-primary btn-dialog',

icon: 'fa fa-list',

url: 'test/detail',

callback: function (data) {

Layer.alert("接收到回传数据：" + JSON.stringify(data), {title: "回传数据"});

}

}],

formatter: Table.api.formatter.operate

}

]

]

});

// 绑定TAB事件(刷新表格数据)

$('.panel-heading a[data-toggle="tab"]').on('shown.bs.tab', function (e) {

var field = $(this).closest("ul").data("field");

var value = $(this).data("value");

var options = table.bootstrapTable('getOptions');

options.pageNumber = 1;

options.queryParams = function (params) {

var filter = {};

if (value !== '') {

filter[field] = value;

}

params.filter = JSON.stringify(filter);

return params;

};

table.bootstrapTable('refresh', {});

return false;

});

// 为表格绑定事件

Table.api.bindevent(table);

},

add: function () {

Controller.api.bindevent();

},

edit: function () {

Controller.api.bindevent();

},

api: {

bindevent: function () {

Form.api.bindevent($("form[role=form]"));

}

}

};

return Controller;

});

--

3、HTML代码

<div class="panel panel-default panel-intro">

<!-- tab页 -->

<div class="panel-heading">

{:build\_heading(null,FALSE)}

<ul class="nav nav-tabs" data-field="status">

<li class="active"><a href="#t-all" data-value="" data-toggle="tab">{:\_\_('All')}</a></li>

{foreach name="statusList" item="vo"}

<li><a href="#t-{$key}" data-value="{$key}" data-toggle="tab">{$vo}</a></li>

{/foreach}

</ul>

</div>

<div class="panel-body">

<div id="myTabContent" class="tab-content">

<div class="tab-pane fade active in" id="one">

<div class="widget-body no-padding">

<div id="toolbar" class="toolbar">

<a href="javascript:;" class="btn btn-primary btn-refresh" title="{:\_\_('Refresh')}" ><i class="fa fa-refresh"></i> </a>

<a href="javascript:;" class="btn btn-success btn-add {:$auth->check('test/add')?'':'hide'}" title="{:\_\_('Add')}" ><i class="fa fa-plus"></i> {:\_\_('Add')}</a>

<a href="javascript:;" class="btn btn-success btn-edit btn-disabled disabled {:$auth->check('test/edit')?'':'hide'}" title="{:\_\_('Edit')}" ><i class="fa fa-pencil"></i> {:\_\_('Edit')}</a>

<a href="javascript:;" class="btn btn-danger btn-del btn-disabled disabled {:$auth->check('test/del')?'':'hide'}" title="{:\_\_('Delete')}" ><i class="fa fa-trash"></i> {:\_\_('Delete')}</a>

<a href="javascript:;" class="btn btn-danger btn-import {:$auth->check('test/import')?'':'hide'}" title="{:\_\_('Import')}" id="btn-import-file" data-url="ajax/upload" data-mimetype="csv,xls,xlsx" data-multiple="false"><i class="fa fa-upload"></i> {:\_\_('Import')}</a>

<div class="dropdown btn-group {:$auth->check('test/multi')?'':'hide'}">

<a class="btn btn-primary btn-more dropdown-toggle btn-disabled disabled" data-toggle="dropdown"><i class="fa fa-cog"></i> {:\_\_('More')}</a>

<ul class="dropdown-menu text-left" role="menu">

<li><a class="btn btn-link btn-multi btn-disabled disabled" href="javascript:;" data-params="status=normal"><i class="fa fa-eye"></i> {:\_\_('Set to normal')}</a></li>

<li><a class="btn btn-link btn-multi btn-disabled disabled" href="javascript:;" data-params="status=hidden"><i class="fa fa-eye-slash"></i> {:\_\_('Set to hidden')}</a></li>

</ul>

</div>

<a class="btn btn-info btn-disabled disabled btn-selected" href="javascript:;"><i class="fa fa-leaf"></i> 获取选中项</a>

<a class="btn btn-success btn-singlesearch" href="javascript:;"><i class="fa fa-user"></i> 自定义搜索</a>

<a class="btn btn-success btn-change btn-start" data-params="action=start" data-url="example/bootstraptable/start" href="javascript:;"><i class="fa fa-play"></i> 启动</a>

<a class="btn btn-danger btn-change btn-pause" data-params="action=pause" data-url="example/bootstraptable/pause" href="javascript:;"><i class="fa fa-pause"></i> 暂停</a>

<a href="javascript:;" class="btn btn-default" style="font-size:14px;color:dodgerblue;">

<i class="fa fa-dollar"></i>

<span class="extend">

金额：<span id="money">0</span>

单价：<span id="price">0</span>

</span>

</a>

</div>

<table id="table" class="table table-striped table-bordered table-hover table-nowrap"

data-operate-edit="{:$auth->check('test/edit')}"

data-operate-del="{:$auth->check('test/del')}"

width="100%">

</table>

</div>

</div>

</div>

</div>

</div>

--

--

--

整理( <https://ask.fastadmin.net/article/323.html> )