[管理后台-前端-AngularJS](http://blog.csdn.net/iamnieo/article/details/50474399)

标签： [AngularJS](http://www.csdn.net/tag/AngularJS)[bootstrap](http://www.csdn.net/tag/bootstrap)[restful](http://www.csdn.net/tag/restful)[crud](http://www.csdn.net/tag/crud)[后台](http://www.csdn.net/tag/%e5%90%8e%e5%8f%b0)

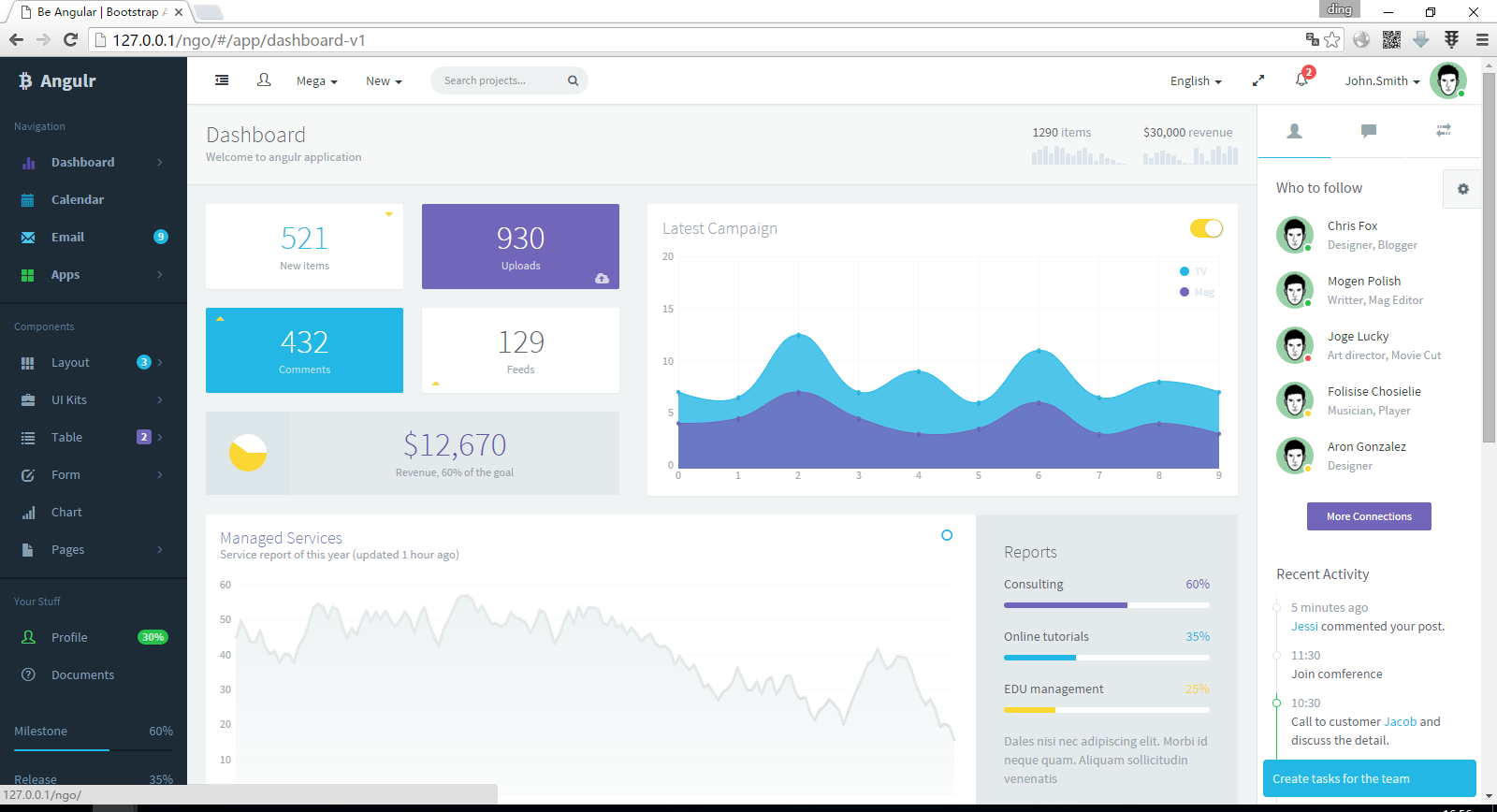
2016-01-07 10:26 25699人阅读 [评论](http://blog.csdn.net/iamnieo/article/details/50474399#comments)(19) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/iamnieo/article/details/50474399#report)

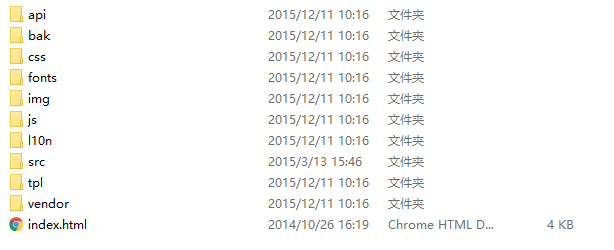
版权声明：本文为博主原创文章，未经博主允许不得转载。

http://blog.csdn.net/iamnieo/article/details/50474399

http://blog.csdn.net/iamnieo/article/details/50441806

ng+bootstrap可以做出很漂亮的管理系统出来，<https://wrapbootstrap.com/>可以付费购买，下文会提供一个免费的，要讲解如何从0到1把ng前端结构搭出来是很漫长的教程，本文仅仅介绍一下这个免费后台模版的结构，然后重点讲解如何改写这个结构。   
开始阅读之前，假设读者已经ng入门并且对于[ui.router](https://github.com/angular-ui/ui-router),[bootstrap](http://getbootstrap.com/)有一定了解。

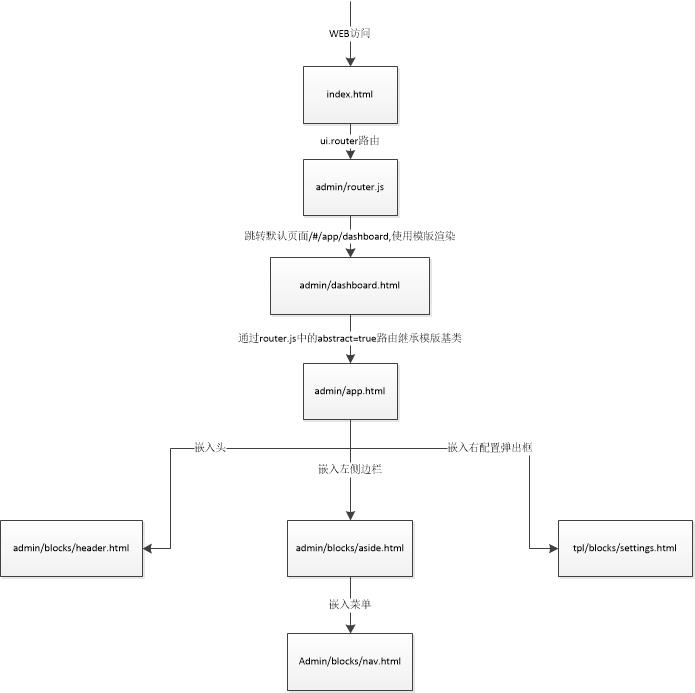
下载链接<http://pan.baidu.com/s/1o73ewfK>   
下载下来以后我们可以挂载到IIS里面看看这个模版的运行效果   


1) 后台结构   


其中我们需要关注的如下：

index.html： 入口   
js： 存放所有业务逻辑代码   
js/app.js：定义ng需要加载的模块   
js/main.js：定义ng的全局配置信息   
js/config.router.js：ng的路由器   
tpl：存放所有页面模版   
tpl/blocks：页面框架（头、尾、侧边栏…)   
vendor：存放所有第三方模块

2) 改写结构，接管路由

结构简图如下   
   
这里接管的原则是尽量不改动原始结构,首先在根目录创建我们自己的目录结构.

> mkdir admin

> mkdir admin\blocks

> copy js\main.js admin\

编写页面框架模版，我们也可以直接从tpl目录复制过来再做修改。简单起见，我们在模版里面直接使用了中文，这样会导致页面乱码，解决方法：html文件用utf-8编码格式存储。

<!--admin/blocks/aside.html-->

<div class="aside-wrap">

<div class="navi-wrap">

<!-- nav -->

<nav ui-nav class="navi" ng-include="'admin/blocks/nav.html'"></nav>

<!-- nav -->

</div>

</div><!--admin/blocks/header.html-->

<!-- navbar header -->

<div class="navbar-header {{app.settings.navbarHeaderColor}}">

<button class="pull-right visible-xs dk" ui-toggle-class="show" data-target=".navbar-collapse">

<i class="glyphicon glyphicon-cog"></i>

</button>

<button class="pull-right visible-xs" ui-toggle-class="off-screen" data-target=".app-aside" ui-scroll="app">

<i class="glyphicon glyphicon-align-justify"></i>

</button>

<!-- brand -->

<a href="#/" class="navbar-brand text-lt">

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

<img src="img/logo.png" alt="." class="hide">

<span class="hidden-folded m-l-xs">{{app.name}}</span>

</a>

<!-- / brand -->

</div>

<!-- / navbar header -->

<!-- navbar collapse -->

<div class="collapse pos-rlt navbar-collapse box-shadow {{app.settings.navbarCollapseColor}}">

<!-- buttons -->

<div class="nav navbar-nav hidden-xs">

<a href class="btn no-shadow navbar-btn" ng-click="app.settings.asideFolded = !app.settings.asideFolded">

<i class="fa {{app.settings.asideFolded ? 'fa-indent' : 'fa-dedent'}} fa-fw"></i>

</a>

</div>

<!-- / buttons -->

<!-- nabar right -->

<ul class="nav navbar-nav navbar-right">

<li class="hidden-xs">

<a ui-fullscreen></a>

</li>

<li class="dropdown" dropdown>

<a href class="dropdown-toggle clear" dropdown-toggle>

<span class="thumb-sm avatar pull-right m-t-n-sm m-b-n-sm m-l-sm">

<img src="img/a0.jpg" alt="...">

<i class="on md b-white bottom"></i>

</span>

<span class="hidden-sm hidden-md"></span> <b class="caret"></b>

</a>

<!-- dropdown -->

<ul class="dropdown-menu animated fadeInRight w">

<li>

<a href="#">Logout</a>

</li>

</ul>

<!-- / dropdown -->

</li>

</ul>

<!-- / navbar right -->

</div>

<!-- / navbar collapse -->

<!--admin/blocks/nav.html-->

<!-- list -->

<ul class="nav">

<li class="hidden-folded padder m-t m-b-sm text-muted text-xs">

<span translate="导航">导航</span>

</li>

</ul>

<!-- / list -->

<!--admin/app.html-->

<!-- navbar -->

<div data-ng-include=" 'admin/blocks/header.html' " class="app-header navbar">

</div>

<!-- / navbar -->

<!-- menu -->

<div data-ng-include=" 'admin/blocks/aside.html' " class="app-aside hidden-xs {{app.settings.asideColor}}">

</div>

<!-- / menu -->

<!-- content -->

<div class="app-content">

<div ui-butterbar></div>

<a href class="off-screen-toggle hide" ui-toggle-class="off-screen" data-target=".app-aside" ></a>

<div ncy-breadcrumb></div>

<div class="app-content-body fade-in-up" ui-view></div>

</div>

<!-- /content -->

<!-- footer -->

<div class="app-footer wrapper b-t bg-light">

<span class="pull-right">{{app.version}} <a href ui-scroll="app" class="m-l-sm text-muted"><i class="fa fa-long-arrow-up"></i></a></span>

&copy; 2016 Copyright.

</div>

<!-- / footer -->

<div data-ng-include=" 'tpl/blocks/settings.html' " class="settings panel panel-default">

</div>

给我们的首页创建一个空白模版admin/dashboard.html

<!--admin/dashboard.html-->

* 1

写我们新的路由

// admin/router.js

'use strict';

app

.run(

function ($rootScope, $state, $stateParams) {

$rootScope.$state = $state;

$rootScope.$stateParams = $stateParams;

}

)

.config(

function ($stateProvider, $urlRouterProvider) {

$urlRouterProvider

.otherwise('/app/dashboard');

$stateProvider

.state('app', {

abstract: true,

url: '/app',

templateUrl: 'admin/app.html',

})

.state('app.dashboard', {

url: '/dashboard',

templateUrl: 'admin/dashboard.html',

})

}

);

* 2修改入口，注释或删除掉对原config.router.js、main.js的引用，换成我们的控制接管

<!--index.html-->

...

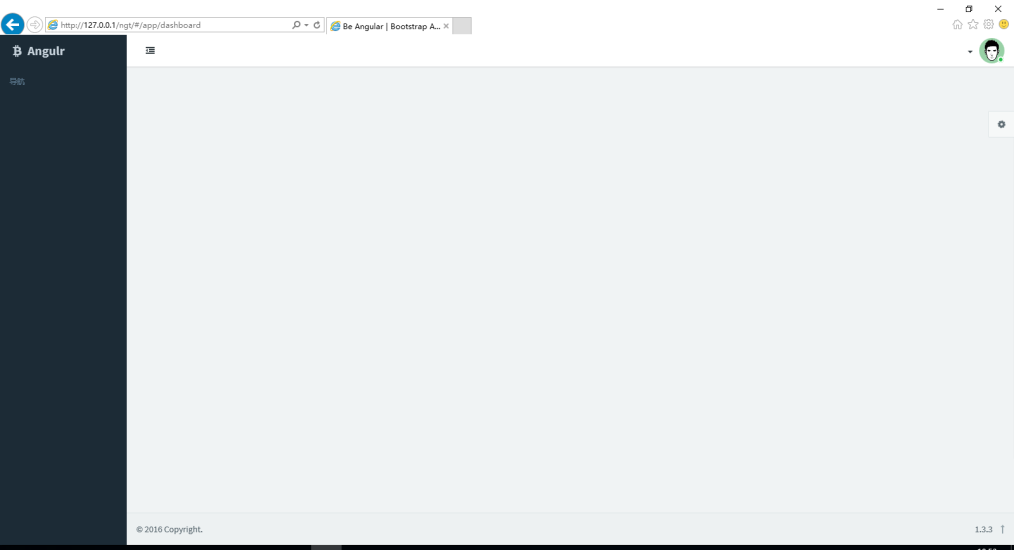
<!--<script src="js/config.router.js"></script>-->

<!--<script src="js/main.js"></script>-->

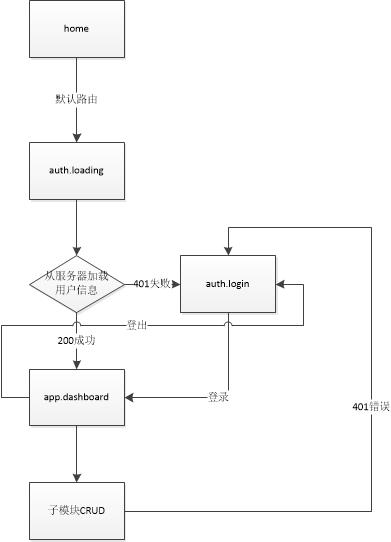
<script src="admin/router.js"></script>

<script src="admin/main.js"></script>

...

  
到此我们就得到了一套可自由扩展的前端框架

3) 接下来我们基于BasicAuth加入系统的用户验证功能。

   
这里我们采用按功能模块来建立子目录，区别于原模版框架（原框架是按文件类型区分子目录，例如脚本放在js/里面，模版放在tpl里面）。

首先启动Restful服务器，然后我们为服务器配置一个全局变量，**代码里面的host需要修改成服务器真实地址**

// admin/main.js

...

$scope.app = {

host: "http://172.17.9.92:8000",

name: 'Angulr',

...

创建认证功能模块目录auth

> mkdir admin\auth\

* 1

编写controller(控制器)

// admin/auth/ctrl.js

app.controller('LoadingController',function($scope,$resource,$state){

var $com = $resource($scope.app.host + "/auth/info/?");

$com.get(function(){

$state.go('app.dashboard');

},function(){

$state.go('auth.login');

})

});

app.controller('LoginController',function($scope,$state,$http,$resource,Base64){

$scope.login = function(){

$scope.authError = ""

var authdata = Base64.encode($scope.user.username + ':' + $scope.user.password);

$http.defaults.headers.common['Authorization'] = 'Basic ' + authdata;

var $com = $resource($scope.app.host + "/auth/info/?");

$com.get(function(){

$state.go('app.dashboard');

},function(){

$scope.authError = "服务器登录错误"

})

}

});

app.factory('Base64',function(){

var keyStr = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=';

return {

encode: function (input) {

var output = "";

var chr1, chr2, chr3 = "";

var enc1, enc2, enc3, enc4 = "";

var i = 0;

do {

chr1 = input.charCodeAt(i++);

chr2 = input.charCodeAt(i++);

chr3 = input.charCodeAt(i++);

enc1 = chr1 >> 2;

enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);

enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);

enc4 = chr3 & 63;

if (isNaN(chr2)) {

enc3 = enc4 = 64;

} else if (isNaN(chr3)) {

enc4 = 64;

}

output = output +

keyStr.charAt(enc1) +

keyStr.charAt(enc2) +

keyStr.charAt(enc3) +

keyStr.charAt(enc4);

chr1 = chr2 = chr3 = "";

enc1 = enc2 = enc3 = enc4 = "";

} while (i < input.length);

return output;

},

decode: function (input) {

var output = "";

var chr1, chr2, chr3 = "";

var enc1, enc2, enc3, enc4 = "";

var i = 0;

// remove all characters that are not A-Z, a-z, 0-9, +, /, or =

var base64test = /[^A-Za-z0-9\+\/\=]/g;

if (base64test.exec(input)) {

window.alert("There were invalid base64 characters in the input text.\n" +

"Valid base64 characters are A-Z, a-z, 0-9, '+', '/',and '='\n" +

"Expect errors in decoding.");

}

input = input.replace(/[^A-Za-z0-9\+\/\=]/g, "");

do {

enc1 = keyStr.indexOf(input.charAt(i++));

enc2 = keyStr.indexOf(input.charAt(i++));

enc3 = keyStr.indexOf(input.charAt(i++));

enc4 = keyStr.indexOf(input.charAt(i++));

chr1 = (enc1 << 2) | (enc2 >> 4);

chr2 = ((enc2 & 15) << 4) | (enc3 >> 2);

chr3 = ((enc3 & 3) << 6) | enc4;

output = output + String.fromCharCode(chr1);

if (enc3 != 64) {

output = output + String.fromCharCode(chr2);

}

if (enc4 != 64) {

output = output + String.fromCharCode(chr3);

}

chr1 = chr2 = chr3 = "";

enc1 = enc2 = enc3 = enc4 = "";

} while (i < input.length);

return output;

}

};

})

编写template(模版)

<!--admin/auth/loading.html-->

<div class="form-group" ng-controller="LoadingController">

<div class="col-md-12 text-center">

<span class="glyphicon glyphicon-refresh glyphicon-refresh-animate"></span>&nbsp;&nbsp;loading...

</div>

</div>

<!--admin/auth/login.html-->

<div class="container w-xxl w-auto-xs" ng-controller="LoginController">

<a href class="navbar-brand block m-t">{{app.name}}</a>

<div class="m-b-lg">

<div class="wrapper text-center">

<strong>Sign in to get in touch</strong>

</div>

<form name="form" class="form-validation">

<div class="list-group list-group-sm">

<div class="list-group-item">

<input type="text" placeholder="username" class="form-control no-border" ng-model="user.username" required>

</div>

<div class="list-group-item">

<input type="password" placeholder="password" class="form-control no-border" ng-model="user.password" required>

</div>

</div>

<button type="submit" class="btn btn-lg btn-primary btn-block" ng-click="login()" ng-disabled='form.$invalid'>Log in</button>

<div class="line line-dashed"></div>

<div class="text-danger wrapper text-center" ng-show="authError">

{{authError}}

</div>

</form>

</div>

</div>

改写路由

// admin/router.js

...

$urlRouterProvider

.otherwise('/auth/loading');

$stateProvider

.state('auth',{

abstract: true,

url:'/auth',

template: '<div ui-view class="fade-in"></div>',

resolve: {

deps: ['$ocLazyLoad',

function( $ocLazyLoad ){

return $ocLazyLoad.load('admin/auth/ctrl.js');

}]

}

})

.state('auth.loading',{

url:'/loading',

templateUrl:'admin/auth/loading.html',

})

.state('auth.login',{

url:'/login',

templateUrl:'admin/auth/login.html',

})

...

* 2代码挺多，就不逐行解释了，最核心的就是:   
  用Base64加密[用户名:密码]，在请求头加入 Authorization: Basic [加密串]

var authdata = Base64.encode(username + ":" + password); $http.defaults.headers.common['Authorization'] = 'Basic ' + authdata;

* 1

改完以后重新访问，将实现流程图中的home->loading->login->dashboard。   
事情还没完，我们再重新访问，又会重复这个流程，并不会自动登录，这是因为BasicAuth的特性是在每一次web请求的时候都需要加入Authorization请求头才行。所以我们还要做点工作：1.登录成功以后将authdata存在本地，2.给全局http请求统一加入这个请求头

// admin/auth/ctrl.js

...

app.controller('LoginController',function($scope,$state,$http,$resource,Base64,$localStorage)

...

$com.get(function(){

$localStorage.auth = authdata;

$state.go('app.dashboard');

},function(){

$scope.authError = "服务器登录错误"

})

...

* 1// admin/router.js

app

.run(

function ($rootScope, $state, $stateParams,$localStorage,$http) {

$http.defaults.headers.common['Authorization'] = 'Basic ' + $localStorage.auth;

$rootScope.$state = $state;

$rootScope.$stateParams = $stateParams;

}

)

...

重新刷新首页，页面将实现自动登录了，但是事情还没完，进入系统以后，虽然每次Web请求我们都加入了BasicAuth的请求头，但是如果服务器端做了帐号修改，一样会产生401的错误，产生的结果就是客户端点什么操作都不会有反应，我们应该在全局来拦截401，引导客户端跳转到重新登录的界面：

// admin/router.js

...

app.config(function ($httpProvider) {

$httpProvider.interceptors.push('AuthInterceptor');

})

app.factory('AuthInterceptor', function ($rootScope, $q,$location) {

return {

responseError: function (response) {

if(response.status==401)

{

$location.url('/auth/login');

}

return $q.reject(response);

}

};

})

大功即将告成，还有最后一步，有了登录必然有登出，BasicAuth协议本身是没有登出概念的，我们这里做的登出，就是删除本地那个保存的authdata。

// admin/main.js

angular.module('app')

.controller('AppCtrl', ['$scope', '$translate', '$localStorage', '$window', '$state','$http',

function( $scope, $translate, $localStorage, $window ,$state,$http) {

...

$scope.logout = function(){

$localStorage.auth = null;

$http.defaults.headers.common['Authorization'] = "Basic";

$state.go("auth.login");

}

function isSmartDevice( $window ){...}

* 1<!--admin/blocks/header.html-->

...

<!-- dropdown -->

<ul class="dropdown-menu animated fadeInRight w">

<li>

<a ng-click="logout()">Logout</a>

</li>

</ul>

<!-- / dropdown -->

...

锦上添花，标准的后台系统一般会在页面右上角显示登录用户的帐号信息，我们定义的协议/auth/info/是会把这些信息带下来的，我们来补全一下这个功能:

// admin/auth/ctrl.js

app.controller('LoadingController',function($scope,$resource,$state,$localStorage){

var $com = $resource($scope.app.host + "/auth/info/?");

$com.get(function(data){//引入data

$scope.session\_user = $localStorage.user = data; //保存用户信息

$state.go('app.dashboard');

})

});

app.controller('LoginController',function($scope,$state,$http,$resource,Base64,$localStorage){

$scope.login = function(){

$scope.authError = ""

var authdata = Base64.encode($scope.user.username + ':' + $scope.user.password);

$http.defaults.headers.common['Authorization'] = 'Basic ' + authdata;

var $com = $resource($scope.app.host + "/auth/info/?");

$com.get(function(data){//引入data

$scope.session\_user = $localStorage.user = data; //保存用户信息

$localStorage.auth = authdata;

$state.go('app.dashboard');

},function(){

$scope.authError = "服务器登录错误"

})

}

});

...

// admin/main.js

...

$scope.session\_user = $localStorage.user;

$scope.logout = function(){...}

<!--admin/blocks/header.html-->

...

<span class="hidden-sm hidden-md">{{session\_user.username}}</span> <b class="caret"></b>

...

4) CRUD

创建news目录

> mkdir admin\news\

* 1

增加news导航

<!-- admin/blocks/nav.html -->

<!-- list -->

<ul class="nav">

<li class="hidden-folded padder m-t m-b-sm text-muted text-xs">

<span translate="导航">导航</span>

</li>

<li ui-sref-active="active">

<a ui-sref="app.news.list">

<i class="glyphicon glyphicon-book icon text-info-dker"></i>

<span class="font-bold">新闻管理</span>

</a>

</li>

</ul>

<!-- / list -->

* 1书写控制器

// admin/news/ctrl.js

'use strict';

app.controller('ListController', function($scope, $resource,$stateParams,$modal,$state) {

//查询

$scope.query = function(page,filter){

var $com = $resource($scope.app.host + "/news/?page=:page&search=:filter",{page:'@page',filter:'@filter'});

if(!page){

page=1;

}else{

page=parseInt(page);

}

$com.get({page:page,filter:filter},function(data){

//扩展分页数据，显示页签，最终效果为 < 1 2 3 4 5 >

data.page\_index = page;

data.pages = []; //页签表

var N = 5; //每次显示5个页签

var s = Math.floor(page/N)\*N;

if(s==page)s-=N;

s += 1;

var e = Math.min(data.page\_count,s+N-1)

for(var i=s;i<=e;i++)

data.pages.push(i)

$scope.data = data;

$scope.search\_context = filter;

});

}

//搜索跳转

$scope.search = function(){

$state.go('app.news.list',{search:$scope.search\_context});

}

//全选

var selected = false;

$scope.selectAll = function(){

selected = !selected;

angular.forEach($scope.data.results,function(item){

item.selected = selected;

});

}

//自定义操作处理，其中1为删除所选记录

$scope.exec = function(){

if($scope.operate=="1"){

var ids = [];

angular.forEach($scope.data.results,function(item){

if(item.selected){

ids.push(item.id);

}

});

if(ids.length>0){

//弹出删除确认

var modalInstance = $modal.open({

templateUrl: 'admin/confirm.html',

controller: 'ConfirmController',

size:'sm',

});

modalInstance.result.then(function () {

var $com = $resource($scope.app.host + "/news/deletes/?");

$com.delete({'ids':ids.join(',')},function(){

$state.go('app.news.list');

});

});

}

}

}

//根据url参数（分页、搜索关键字）查询数据

$scope.query($stateParams.page,$stateParams.search);

});

app.controller('ConfirmController', ['$scope', '$modalInstance', function($scope, $modalInstance){

$scope.ok = function () {

$modalInstance.close();

};

$scope.cancel = function () {

$modalInstance.dismiss('cancel');

};

}]);

app.controller('DetailController', function($rootScope,$scope, $resource, $stateParams,$state) {

$scope.edit\_mode = !!$stateParams.id;

if($scope.edit\_mode){

var $com = $resource($scope.app.host + "/news/:id/?",{id:'@id'});

var resp = $com.get({id:$stateParams.id},function(data){

$scope.data = resp;

});

}

else{

$scope.data = {};

}

$scope.submit = function(){

if($scope.edit\_mode){

var $com = $resource($scope.app.host + "/news/:id/?",{id:'@id'},{

'update': { method:'PUT' },

});

$com.update({id:$stateParams.id},$scope.data,function(data){

$state.go($rootScope.previousState,$rootScope.previousStateParams);

});

}

else{

var $com = $resource($scope.app.host + "/news/?");

$com.save($scope.data,function(data){

$state.go('app.news.list');

});

}

};

$scope.delete = function(){

var $com = $resource($scope.app.host + "/news/:id/?",{id:'@id'});

$com.delete({id:$stateParams.id},function(){

$state.go('app.news.list');

})

}

});

书写列表模版

<!-- admin/news/list.html -->

<div class="wrapper-md" ng-controller="ListController">

<div class="panel panel-default">

<div class="panel-heading">

<ul class="nav nav-pills pull-right">

<li style=" padding-top:4px; padding-right:4px"><button class="btn m-b-xs btn-sm btn-primary btn-addon" ui-sref="app.news.create()"><i class="fa fa-plus"></i>新增</button></li>

</ul>

新闻列表

</div>

<div class="row wrapper">

<div class="col-sm-5 m-b-xs">

<select class="input-sm form-control w-sm inline v-middle" ng-model="operate" ng-init="operate=0">

<option value="0">---</option>

<option value="1">删除所选记录</option>

</select>

<button class="btn btn-sm btn-default" ng-click="exec()">执行</button>

</div>

<div class="col-sm-4">

</div>

<div class="col-sm-3">

<div class="input-group">

<input type="text" class="input-sm form-control" placeholder="Search" ng-model="search\_context">

<span class="input-group-btn">

<button class="btn btn-sm btn-default" ng-click="search()" type="button">Go!</button>

</span>

</div>

</div>

</div>

<div class="table-responsive" ng-if="data.total\_count>0">

<table class="table table-striped b-t b-light">

<thead>

<tr>

<th style="width:20px;">

<label class="i-checks m-b-none">

<input type="checkbox" ng-click="selectAll()"><i></i>

</label>

</th>

<th>标题</th>

<th>创建时间</th>

<th style="width:30px;"></th>

</tr>

</thead>

<tbody>

<tr ng-repeat="data in data.results">

<td><label class="i-checks m-b-none"><input type="checkbox" ng-model="data.selected"><i></i></label></td>

<td>{{data.title}}</td>

<td>{{data.create\_time|date:'yyyy-MM-dd HH:mm:ss Z'}}</td>

<td>

<a ui-sref="app.news.detail({id:data.id})" class="active"><i class="fa fa-edit"></i></a>

</td>

</tr>

</tbody>

</table>

</div>

<footer class="panel-footer">

<div class="row">

<div class="col-sm-8 text-left">

<small class="text-muted inline m-t-sm m-b-sm">{{data.total\_count}}条记录</small>

</div>

<div ng-if="data.page\_count>1" class="col-sm-4 text-right text-center-xs">

<ul class="pagination pagination-sm m-t-none m-b-none">

<li ng-class="{disabled:!data.previous}"><a ui-sref="app.news.list({page:data.page\_index-1,search:search\_context})"><i class="fa fa-chevron-left"></i></a></li>

<li ng-repeat="page in data.pages" ng-class="{active:page==data.page\_index}"><a ui-sref="app.news.list({page:page,search:search\_context})">{{page}}</a></li>

<li ng-class="{disabled:!data.next}"><a ui-sref="app.news.list({page:data.page\_index+1,search:search\_context})"><i class="fa fa-chevron-right"></i></a></li>

</ul>

</div>

</div>

</footer>

</div>

</div>

<!-- admin/news/detail.html -->

<div ng-controller="DetailController">

<div class="wrapper-md" >

<div class="panel panel-default">

<form class="form-horizontal ng-pristine ng-valid ng-valid-date ng-valid-required ng-valid-parse ng-valid-date-disabled" ng-submit="submit()">

<div class="panel-body">

<div class="form-group">

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

<div class="col-sm-10">

<input type="text" class="form-control" ng-model="data.title" required>

</div>

</div>

<div class="line line-dashed b-b line-lg pull-in"></div>

<div class="form-group">

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

<div class="col-sm-10">

<textarea class="form-control" rows="6" ng-model="data.content"></textarea>

</div>

</div>

</div>

<footer class="panel-footer text-right bg-light lter">

<input type="button" ng-click="delete()" ng-if="edit\_mode" class="btn btn-danger pull-left" value="删除"/>

<input type="submit" class="btn btn-success" value="提交"/>

</footer>

</form>

</div>

</div>

</div>

书写删除确认框模版

<!-- admin/confirm.html -->

<div class="modal-header">

<h3>确认删除?</h3>

</div>

<div class="modal-footer">

<button class="btn btn-default" ng-click="cancel()">Cancel</button>

<button class="btn btn-primary" ng-click="ok()">OK</button>

</div>

添加路由，注意run里面增加了事件监听，后文详细说

// admin/router.js

app

.run(

function (...) {

...

$rootScope.$on('$stateChangeSuccess', function(event, to, toParams, from, fromParams) {

$rootScope.previousState = from;

$rootScope.previousStateParams = fromParams;

});

}

)

...

.state('app.news', {

abstract: true,

url: '/news',

template: '<div ui-view class="fade-in"></div>',

resolve: {

deps: ['$ocLazyLoad',

function( $ocLazyLoad ){

return $ocLazyLoad.load('admin/news/ctrl.js');

}]

}

})

.state('app.news.list', {

url: '/list?page&search',

templateUrl: 'admin/news/list.html',

})

.state('app.news.detail', {

url: '/detail/{id}',

templateUrl: 'admin/news/detail.html',

})

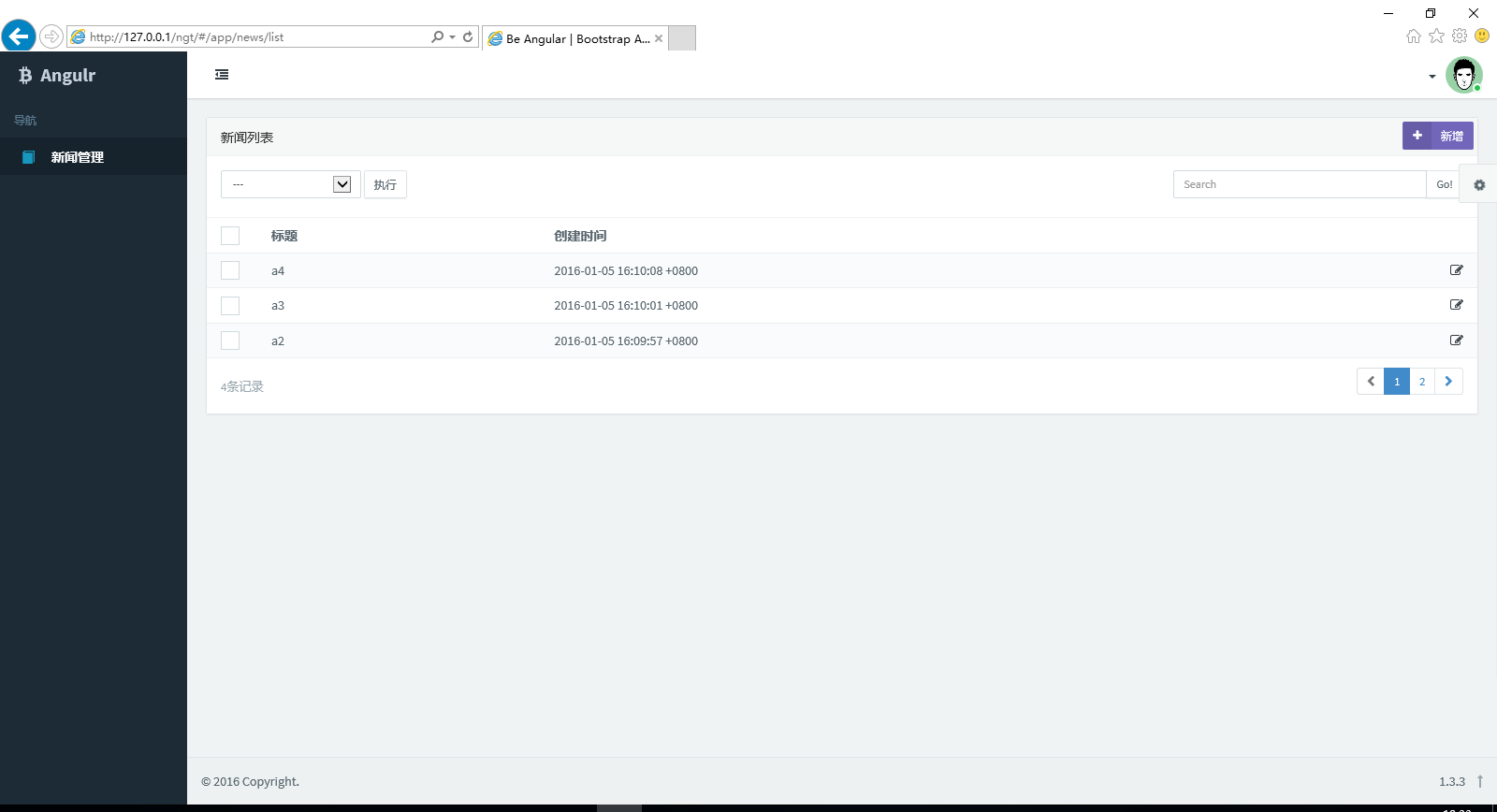
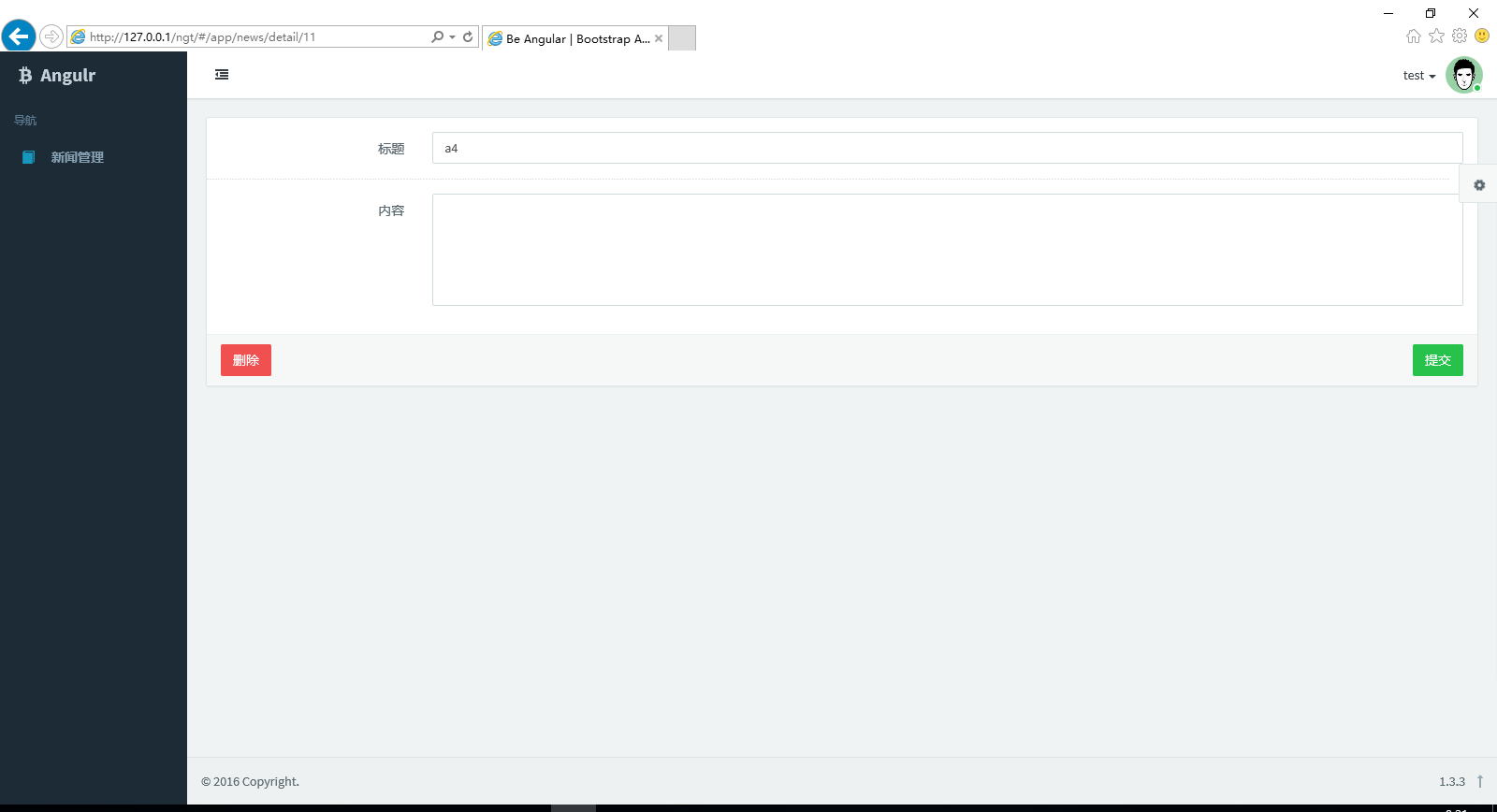
.state('app.news.create', {

url: '/create',

templateUrl: 'admin/news/detail.html',

})

...

成果如下   
  


我们这里搜索和分页都是采用URL跳转的方式(#/news/list/?search=str&page=int)，这样能保证刷新页面的时候能停留在之前的页面结果上，ng默认的页面跳转是不保留前一个页面状态的(链接和参数)，如果我们跳转到第2页，编辑，再返回，是会回到第1页去，为了比较好的用户体验所以我们有了如下代码

监听全局页面跳转信号($statChangeSuccess)，将参数保存下来

// admin/router.js

app.run(

...

$rootScope.$on('$stateChangeSuccess', function(event, to, toParams, from, fromParams) {

$rootScope.previousState = from;

$rootScope.previousStateParams = fromParams;

});

...

)

详情页编辑完成返回时读取参数跳转

// admin/news/ctrls.js

app.controller('DetailController',...){

...

$com.update({id:$stateParams.id},$scope.data,function(data){

$state.go($rootScope.previousState,$rootScope.previousStateParams);

});

...

});

完整的后台少不了导航   
   
我们这里选用github上面一个写得挺棒的ng导航插件[angular-breadcrumb](http://blog.csdn.net/iamnieo/article/details/angular-breadcrumb)，求简，我们直接下载<https://raw.github.com/ncuillery/angular-breadcrumb/master/dist/angular-breadcrumb.min.js>，由于我们是给ng装插件，所以建议放到vendor/里面去，接下来的改动也是针对原模版框架。

注入JS

<!-- index.html -->

...

<script src="vendor/angular/angular-breadcrumb/angular-breadcrumb.min.js"></script>

...

引入模块

// js/app.js

angular.module('app', [

...

'ncy-angular-breadcrumb',

]);

配置扩展，让导航能支持HTML(具体就是能显示回首页的图标)   
这里写图片描述

// js/config.js

...

app.config(function($breadcrumbProvider) {

$breadcrumbProvider.setOptions({

templateUrl: 'tpl/blocks/breadcrumb.html'

});

});

扩展的导航模版，写法参考插件官方文档

<!-- tpl/blocks/breadcrumb.html -->

<ul class="breadcrumb bg-white b-a">

<li ng-repeat="step in steps | limitTo:(steps.length-1)">

<a href="{{step.ncyBreadcrumbLink}}" ng-bind-html="step.ncyBreadcrumbLabel"></a>

</li>

<li ng-repeat="step in steps | limitTo:-1" class="active">

<span ng-bind-html="step.ncyBreadcrumbLabel"></span>

</li>

</ul>

配置路由，加入导航

// admin/router.js

...

.state('app.dashboard', {

...

ncyBreadcrumb: {

label: '<i class="fa fa-home"></i> 首页'

}

})

...

.state('app.news.list', {

...

ncyBreadcrumb: {

parent:'app.dashboard',

label: '新闻列表',

}

})

.state('app.news.detail', {

...

ncyBreadcrumb: {

parent:'app.news.list',

label: '编辑',

}

})

.state('app.news.create', {

...

ncyBreadcrumb: {

parent:'app.news.list',

label: '新增',

}

})

...

