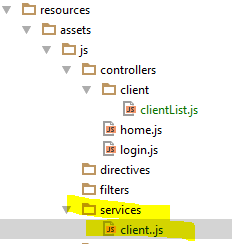
Vamos trabalhar com resources o que vão se comunicar com nossa api.

Resource encapsula tudo numa entidade

Criar módulo separado só para os serviços

**No App.js**

**var *app*** = **angular**.module(**'app'**, [**'ngRoute'**, **'angular-oauth2'**, **'app.controllers'**, **'app.services'**]);  
  
**angular**.module(**'app.controllers'**, [**'ngMessages'**, **'angular-oauth2'**]);  
**angular**.module(**'app.services'**, [**'ngResource'**]);  
  
***app***.**provider**(**'appConfig'**, **function**(){  
 **var** config = {  
 **baseUrl**: **'http://localhost:8000'** };  
  
 **return** {  
 **config**: config,  
 $get: **function**() {  
 **return** config;  
 }  
 }  
});  
  
*//rotas****app***.config([  
 **'$routeProvider'**, **'OAuthProvider'**, **'OAuthTokenProvider'**, **'appConfigProvider'**,  
 **function**($routeProvider, OAuthProvider, OAuthTokenProvider, appConfigProvider){  
 $routeProvider  
 .when(**'/login'**, {  
 **templateUrl**: **'build/views/login.html'**,  
 **controller**:**'LoginController'** })  
  
 .*when*(**'/home'**, {  
 **templateUrl**: **'build/views/home.html'**,  
 **controller**:**'HomeController'** })  
  
 .**when**(**'/clients'**, {  
 **templateUrl**: **'build/views/client/list.html'**,  
 **controller**:**'ClientListController'** });  
  
 *//Autenticar* OAuthProvider.configure({  
 **baseUrl**: appConfigProvider.config.**baseUrl**,  
 **clientId**: **'appid1'**,  
 **clientSecret**: **'secret'**, *// optional* **grantPath**: **'oauth/access\_token'**,  
 });  
  
 *//Deixar só com HTTP* OAuthTokenProvider.configure({  
 **name**: **'token'**,  
 **options**: {  
 **secure**: **false** }  
 });  
}]);  
  
*//depois que o angular é carregado isso é executado  
//esta adicionando um evento OAuth error para se for invalido não retornanr se for um token  
// invalido retoranar uma atualizacao o token****app***.run([**'$rootScope'**, **'$window'**, **'OAuth'**, **function**($rootScope, $window, OAuth) {  
 $rootScope.$on(**'oauth:error'**, **function**(event, rejection) {  
 *// Ignore `invalid\_grant` error - should be catched on `LoginController`.* **if** (**'invalid\_grant'** === rejection.data.error) {  
 **return**;  
 }  
  
 *// Refresh token when a `invalid\_token` error occurs.* **if** (**'invalid\_token'** === rejection.data.error) {  
 **return** OAuth.getRefreshToken();  
 }  
  
 *// Redirect to `/login` with the `error\_reason`.* **return** $window.**location**.**href** = **'/login?error\_reason='** + rejection.data.error;  
 });  
}]);



**Client.js**

**angular**.module(**'app.services'**)  
 .service(**'Client'**, [**'$resource'**, **'appConfig'**, **function** ($resource, appConfig) {  
 **return** $resource(appConfig.**baseUrl** + **'/client/:id'**, {**id**: **'@id'**}, {  
 **query**: {  
 **isArray**: **true**,  
 **method**: **'GET'**,  
 transformResponse: **function**(data,header){  
 **var** dataJson = ***JSON***.parse(data);  
 dataJson = dataJson.data;  
 **return** dataJson;  
 }  
 }  
 });  
 }]);

**Controller clientList.js**

**angular**.module(**'app.controllers'**)  
 .controller(**'ClientListController'**, [**'$scope'**, **'Client'**, **function** ($scope, Client) {  
 $scope.**clients** = Client.query();  
 }  
 ]);

**View list.html**

<**div class="container-fluid"**>  
 <**div class="row"**>  
 <**div class="col-md-8 col-md-offset-2"**>  
 <**div class="panel panel-default"**>  
 <**div class="panel-heading"**>Listagem de Clientes</**div**>  
 <**div class="panel-body"**>  
 <**table class="table"**>  
 <**thead**>  
 <**tr**>  
 <**th**>#</**th**>  
 <**th**>Name</**th**>  
 <**th**>Responsible</**th**>  
 <**th**>E-mail</**th**>  
 <**th**>Phone</**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 <**tr ng-repeat="o in clients"**>  
 <**td**>{{o.id}}</**td**>  
 <**td**>{{o.name}}</**td**>  
 <**td**>{{o.responsible}}</**td**>  
 <**td**>{{o.email}}</**td**>  
 <**td**>{{o.phone}}</**td**>  
 </**tr**>  
 </**tbody**>  
 </**table**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
</**div**>