**Resolve（预载入）**

参考资料：

使用预载入功能，开发者可以预先载入一系列依赖或者数据，然后注入到控制器中。在ngRoute中resolve选项可以允许开发者在路由到达前载入数据保证（promises）。在使用这个选项时比使用angular-route有更大的自由度。

预载入选项需要一个对象，这个对象的key即要注入到控制器的依赖，这个对象的value为需要被载入的factory服务。

如果传入的时字符串，angular-route会试图匹配已经注册的服务。如果传入的是函数，该函数将会被注入，并且该函数返回的值便是控制器的依赖之一。如果该函数返回一个数据保证（promise），这个数据保证将在控制器被实例化前被预先载入并且数据会被注入到控制器中。

<body >

<div ng-app="myApp" >

<a ui-sref="index">show index</a>

<div ui-view></div>

</div>

</body>

<script type="text/javascript">

var app = angular.module('myApp', ['ui.router']);

app.config(["$stateProvider", function ($stateProvider) {

$stateProvider

.state("home", {

url: '/',

template:"<div>homePage</div>"

})

.state("index", {

url: '/index/{id}',

template:"<div>indexcontent</div>",

resolve: {

//这个函数的值会被直接返回，因为它不是数据保证

user: function() {

return {

name: "peter",

email: "audiogroup@qq.com"

}

},

//这个函数为数据保证, 因此它将在控制器被实例化之前载入。

detail: function($http) {

return $http({

method: 'JSONP',

url: '/current\_details'

});

},

//前一个数据保证也可作为依赖注入到其他数据保证中！（这个非常实用）

myId: function($http, detail) {

$http({

method: 'GET',

url: 'http://facebook.com/api/current\_user',

params: {

email: currentDetails.data.emails[0]

}

})

}

},

controller:function(user,detail,myId$scope){

alert(user.name)

alert(user.email)

console.log(detail)

}

})

}]);

</script>