**简介**

[Bootstrap](http://getbootstrap.com/)是Twitter推出的一个用于前端开发的开源工具包，它由Twitter的设计师Mark Otto和Jacob Thornton合作开，是一个CSS/HTML框架。

[AngularJS](http://angularjs.org/)是Google 开源出来的一套 js 工具，为了克服HTML在构建应用上的不足而设计的，试图成为WEB应用中的一种端对端的解决方案，通过为开发者呈现一个更高层次的抽象来简化应用的开发，后面简称"ng"。

[Seajs](http://seajs.org/docs/)是一款优秀的模块开发插件，可以实现按需加载。

　　Bootstrap有自己的一套js库，但是为了防止与AngularJS产生兼容性的问题，我们不选择使用，只是使用它的css而已，因此这里我还要引入另外一个基于Bootstrap与AngularJS的UI组件--[UI-Bootstrap](http://angular-ui.github.io/bootstrap/)，虽然该库提供的控件并不多，但是基本够用，而且可以参考它的代码来自己实现复杂的控件(后面的文章再讲)。

**背景**

　　当前Web前端开发中，我们已经可以很熟练的使用jQuery在对数据交互的时候来操作关联Dom的一些内容、样式等状态上的变化，然而很多的这些变化是代码重复的，虽然它并不复杂，但是有些确实需要大量的代码来进行操作；虽然jQuery为我们提供了大量的操作方式来减少复杂性，但是每次我们都要从页面中找寻一些相关的元素改变状态是重复且难以避免的，那么怎么样才能从这种重复的操作中解脱呢？

　　MVVM便可以让我们从这种重复的操作中解脱出来，简单的说MVVM就是我们将需要操作的元素与ViewModel进行绑定，当我们对ViewModel进行操作导致其状态发生变化的时候，与之相关的元素会自动发生变化。

　　由于我也使用jQuery开发了较长的时间，虽然前几年就已经有前端MVVM框架出现，却没有能让我觉得比原始方式更简单的，但是技术总是在发展的并趋向成熟，现在就可以将以上提到的3种框架进行结合，大大简化Web前端的开发(虽然需要一段时间的学习和适应)，接下来就让我们一起来使用这3中框架进行开发吧(这里我不会对各个库有详细的介绍，有些地方仍然需要自己去学习)。

**实现**

　　首先我们从登录功能开始着手，登录界面比较简单，html代码如下：

1 <body **class**="container" ng-app="login">  
2     <form name="frmLogin" **class**="col-sm-offset-4 col-sm-4" role="form" ng-controller="HomeLoginController">  
3         <input name="txtName" ng-model="name" type="text" **class**="form-control input-lg" placeholder="请输入用户名" required autofocus />  
4         <input name="txtPwd" ng-model="pwd" type="password" **class**="form-control input-lg" placeholder="密码" required />  
5         <button **class**="btn-lg btn-primary btn-block" ng-click="submit()">登录</button>  
6         <alert ng-repeat="a in alerts" type="a.type" close="closeAlert($index)">{{a.msg}}</alert>  
7     </form>  
8 </body>

　　这里不使用ngSubmit、ngShow、ngHide是因为有一些注意点需要讲解，登录的流程是当点击登录按钮的时候，如果用户名、密码未填写的情况下，会在登录按钮下面增加2个Alert提醒框，js代码如下：

angular.**module**('login', ['ui.bootstrap']);  
**function** HomeLoginController($scope) {  
    $scope.alerts = [];  
  
    $scope.submit = **function** () {  
        **if** ($scope.frmLogin.txtName.$invalid)  
            $scope.alerts.push({ type: 'danger', msg: '用户名不能为空!' });  
        **if** ($scope.frmLogin.txtPwd.$invalid)  
            $scope.alerts.push({ type: 'danger', msg: '密码不能为空!' });  
  
        **if** ($scope.alerts.length)  
            **return**;  
  
        $scope.alerts = [{ type: 'success', msg: '登录成功' }];  
    };  
  
    $scope.closeAlert = **function** (index) {  
        $scope.alerts.splice(index, 1);  
    };  
};

　　当用户名密码未填写或者表单正确的情况下，点击提交按钮的时候会出现以下2种情况，如图：



　　当我们点击X的时候(多次点击的时候会出现多个提示，别在意这个小问题，^\_^)，会调用closeAlert方法，并根据$index删除alerts数组内对应的元素，这时候绑定的视图就会自动刷新，不需要手动去操作这些元素，这里要注意的主要有3个地方：1、ng-app="login"是不能缺少的，除非你要手动启动ng；2、开头的angular.module('login', ['ui.bootstrap']);的意思是注册login模块且该模块需要对ui-bootstrap的引用(否则无法显示alert的效果)；3、HomeLoginController方法的参数$scope是不能重命名成其他名字，否则ng会报错。

　　如果我们想要让弹出来的Alert自己消失怎么办呢？一开始的时候，大家可能会跟我一样，会尝试使用setTimeout函数去直接从$scope.alerts数组内将对应的元素移除掉，然后等待着视图将Alert移除掉，但是最后会发现一点作用都没有，因为视图数据是在ng内部进行控制的，需要使用它提供的机制去处理，视图才会跟着数据发生变化，这里需要使用$timeout去完成以上的操作，代码改动如下：

//其他代码略  
  
$scope.submit = **function** () {  
    **if** ($scope.frmLogin.txtName.$invalid)  
        addAlert({ type: 'danger', msg: '用户名不能为空!' });  
    **if** ($scope.frmLogin.txtPwd.$invalid)  
        addAlert({ type: 'danger', msg: '密码不能为空!' });  
  
    **if** ($scope.alerts.length)  
        **return**;  
  
    addAlert({ type: 'success', msg: '登录成功' });  
};  
  
**function** addAlert(alert) {  
    $scope.alerts.push(alert);  
    $timeout(**function** () {  
        angular.forEach($scope.alerts, **function** (a, i) {  
            **if** (alert != a)  
                **return**;  
            $scope.alerts.splice(i, 1);  
        });  
    }, 500);  
};

　　写到这里已经把大致该出现的效果都做出来了，现在的问题是这个脚本是直接写在页面上的，那么怎么样通过seajs来加载呢？这里就需要用到angular的手动加载模式了，首先需要对html进行修改要先将ng-app给移除掉，不然ng会自动加载，由于已经将脚本转移到js文件内了，无法找到HomeLoginController方法ng会抛出错误，使用seajs后的代码如下：

    //页面  
    seajs.**use**('home-login');  
  
    //脚本文件  
    **this**.define(**function** (**require**, exports, **module**) {  
        angular.**module**('home.login', ['ui.bootstrap']).controller('HomeLoginController', **function** ($scope, $timeout) {  
            //原本HomeLoginController方法内的代码，这里就不重复贴出了  
        });  
  
        angular.bootstrap($('[name=frmLogin]'), ['home.login']);  
    });

　　从代码中可以看到首先要注册模块home.login这个名字是可以任意的，但是必须跟bootstrap内引用的模块对应，然后我们创建页面上需要的Controller(名字一样得对应)，并使用ng的bootstrap来手动启动。

　　那么第一篇结合bootstrap + angularjs + seajs就到这里了，如果有任何错误请各位指出，谢谢，^\_^!