



```
<span class="glyphicon glyphicon-user"></span> Create New User
</button>
<hr>

<h3 ng-show="edit">Create New User:</h3>
<h3 ng-hide="edit">Edit User:</h3>

<form class="form-horizontal">
<div class="form-group">
  <label class="col-sm-2 control-label">First Name:</label>
  <div class="col-sm-10">
    <input type="text" ng-model="fName" ng-disabled="!edit" placeholder="First Name">
  </div>
</div>
<div class="form-group">
  <label class="col-sm-2 control-label">Last Name:</label>
  <div class="col-sm-10">
    <input type="text" ng-model="lName" ng-disabled="!edit" placeholder="Last Name">
  </div>
</div>
<div class="form-group">
  <label class="col-sm-2 control-label">Password:</label>
  <div class="col-sm-10">
    <input type="password" ng-model="passw1" placeholder="Password">
  </div>
</div>
<div class="form-group">
  <label class="col-sm-2 control-label">Repeat:</label>
  <div class="col-sm-10">
    <input type="password" ng-model="passw2" placeholder="Repeat Password">
  </div>
</div>
</form>

<hr>
<button class="btn btn-success" ng-disabled="error || incomplete">
  <span class="glyphicon glyphicon-save"></span> Save Changes
</button>
</div>

<script src = "myUsers.js"></script>
</body>
</html>
```

[尝试一下 »](#)

## 指令解析

| AngularJS 指令 | 描述                     |
|--------------|------------------------|
| <html ng-app | 为 <html> 元素定义一个应用(未命名) |

|                     |   |
|---------------------|---|
| <body ng-controller | 为 <body> 元素定义一个控制器                        |
| <tr ng-repeat       | 循环 users 对象数组，每个 user 对象放在 <tr> 元素中。      |
| <button ng-click    | 当点击 <button> 元素时调用函数 editUser()           |
| <h3 ng-show         | 如果 edit = true 显示 <h3> 元素                 |
| <h3 ng-hide         | 如果 edit = true 隐藏 <h3> 元素                 |
| <input ng-model     | 为应用程序绑定 <input> 元素                        |
| <button ng-disabled | 如果发生错误或者 incomplete = true 禁用 <button> 元素 |

## Bootstrap 类解析

| 元素       | Bootstrap 类      | 定义       |
|----------|------------------|----------|
| <div>    | container        | 内容容器     |
| <table>  | table            | 表格       |
| <table>  | table-striped    | 带条纹背景的表格 |
| <button> | btn              | 按钮       |
| <button> | btn-success      | 成功按钮     |
| <span>   | glyphicon        | 字形图标     |
| <span>   | glyphicon-pencil | 铅笔图标     |
| <span>   | glyphicon-user   | 用户图标     |
| <span>   | glyphicon-save   | 保存图标     |
| <form>   | form-horizontal  | 水平表格     |
| <div>    | form-group       | 表单组      |
| <label>  | control-label    | 控制器标签    |
| <label>  | col-sm-2         | 跨越 2 列   |
| <div>    | col-sm-10        | 跨越 10 列  |

# JavaScript 代码

## myUsers.js

```
angular.module('myApp', []).controller('userCtrl', function($scope) {
  $scope.fName = '';
  $scope.lName = '';
  $scope.passw1 = '';
  $scope.passw2 = '';
  $scope.users = [
    {id:1, fName:'Hege', lName:"Pege" },
    {id:2, fName:'Kim', lName:"Pim" },
    {id:3, fName:'Sal', lName:"Smith" },
    {id:4, fName:'Jack', lName:"Jones" },
    {id:5, fName:'John', lName:"Doe" },
    {id:6, fName:'Peter',lName:"Pan" }
  ];
  $scope.edit = true;
  $scope.error = false;
  $scope.incomplete = false;

  $scope.editUser = function(id) {
    if (id == 'new') {
      $scope.edit = true;
      $scope.incomplete = true;
      $scope.fName = '';
      $scope.lName = '';
    } else {
      $scope.edit = false;
      $scope.fName = $scope.users[id-1].fName;
      $scope.lName = $scope.users[id-1].lName;
    }
  };

  $scope.$watch('passw1',function() {$scope.test();});
  $scope.$watch('passw2',function() {$scope.test();});
  $scope.$watch('fName', function() {$scope.test();});
  $scope.$watch('lName', function() {$scope.test();});

  $scope.test = function() {
    if ($scope.passw1 !== $scope.passw2) {
      $scope.error = true;
    } else {
      $scope.error = false;
    }
    $scope.incomplete = false;
    if ($scope.edit && (!$scope.fName.length ||
    !$scope.lName.length ||
    !$scope.passw1.length || !$scope.passw2.length)) {
      $scope.incomplete = true;
    }
  };

});
```

# JavaScript 代码解析

| Scope 属性           | 用途                            |
|--------------------|-------------------------------|
| \$scope.fName      | 模型变量 (用户名)                    |
| \$scope.lName      | 模型变量 (用户姓)                    |
| \$scope.passw1     | 模型变量 (用户密码 1)                 |
| \$scope.passw2     | 模型变量 (用户密码 2)                 |
| \$scope.users      | 模型变量 (用户的数组)                  |
| \$scope.edit       | 当用户点击创建用户时设置为true。            |
| \$scope.error      | 如果 passw1 不等于 passw2 设置为 true |
| \$scope.incomplete | 如果每个字段都为空(length = 0)设置为 true |
| \$scope.editUser   | 设置模型变量                        |
| \$scope.watch      | 监控模型变量                        |
| \$scope.test       | 验证模型变量的错误和完整性                 |

← AngularJS 输入验证

AngularJS 包含 →



1 篇笔记

 写笔记



那个 test() 的监听函数里，当如果是编辑状态时，密码和重复密码为空时，保存按钮也可以点，因为编辑状态时 `$scope.edit=false`，`$scope.incomplete=false`，因为没有判断密码和重复密码为空，所以 `$scope.error=false`;

所以这个有两种方法，一种是判断密码和重复密码为空时让 `$scope.error=true`;

第二种就是分开创建用户和编辑用户两种状态，创建用户时，所有都不能为空，编辑用户时，密码不能为空。

```
if ($scope.edit) {
    if (!$scope.fName.length || !$scope.lName.length || !$scope.passw1.length || !$scope.passw2.length) {
        $scope.incomplete=true;
    }
}
```

```
}else{  
    if (!$scope.passw1.length || !$scope.passw2.length) {  
        $scope.incomplete=true;  
    }  
}
```

心动随缘 12个月前 (03-23)