AngularJS Form 进阶：远程校验和自定义输入项

大漠穷秋

本节修订了官方提供的2个例子（因为官方的例子特么有Bug！）。实例一用来示范如何用地道的Angular代码进行远程表单校验；实例二示范如何自定义表单中的输入项。

表单远程校验

HTML代码：

```
<!doctype html>
<html ng-app="form-example1">
    <head>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <link href="../bootstrap/css/bootstrap.min.css" rel="stylesheet" media="screen">
        <script src="../angular-1.0.3/angular.js"></script>
        <script src="FormValidation.js"></script>
    </head>
    <body>
        <div>
            <form name="myForm" class="css-form" novalidate>
                <div>
                    整数(0-10):
                    <input type="number" ng-model="size" name="size" min="0" max="10" integ
                    {{size}}
                    <br/>
                    <span ng-show="myForm.size.$error.integer">不是合法的整数！</span>
                    <span ng-show="myForm.size.$error.min || myForm.size.$error.max">
                        数值必须位于0到10之间！
                    </span>
                </div>
                <div>
                    浮点数:
                    <input type="text" ng-model="length" name="length" smart-float />
                        {{length}}
                    <br/>
                    <span ng-show="myForm.length.$error.float">不是合法的浮点数！</span>
                </div>
                <div>
                    远程校验:
                    <input type="text" ng-model="remote" name="remote" remote-validation />
                        {{remote}}
                    <br/>
                    <span ng-show="myForm.remote.$error.remote">非法数据！</span>
```

```
                </div>
            </form>
        </div>
    </body>
</html>
```

JS代码：

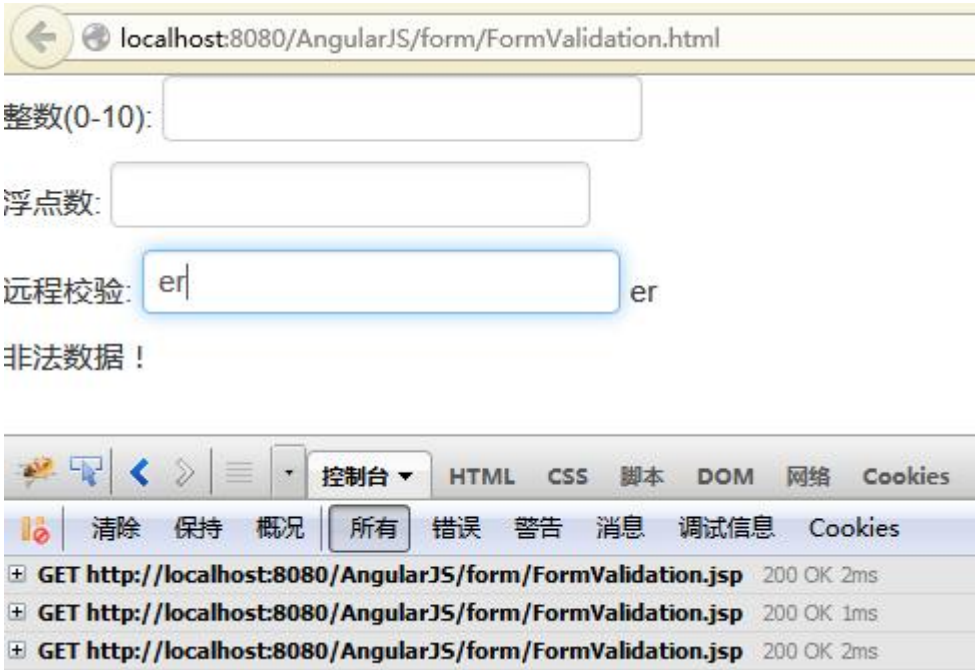
```
var app = angular.module('form-example1', []);
var INTEGER_REGEXP = /^\-?\d*$/;
app.directive('integer', function() {
    return {
        require : 'ngModel',
        link : function(scope, elm, attrs, ctrl) {
            ctrl.$parsers.unshift(function(viewValue) {
                if (INTEGER_REGEXP.test(viewValue)) {
                    ctrl.$setValidity('integer', true);
                    return viewValue;
                } else {
                    ctrl.$setValidity('integer', false);
                    return undefined;
                }
            });
        }
    };
});

var FLOAT_REGEXP = /^\-?\d+((\.|\,)\d+)?$/;
app.directive('smartFloat', function() {
    return {
        require : 'ngModel',
        link : function(scope, elm, attrs, ctrl) {
            ctrl.$parsers.unshift(function(viewValue) {
                if (FLOAT_REGEXP.test(viewValue)) {
                    ctrl.$setValidity('float', true);
                    return parseFloat(viewValue.replace(',','.'));
                } else {
                    ctrl.$setValidity('float', false);
                    return undefined;
                }
            });
        }
    };
});

app.directive('remoteValidation', function($http) {
    return {
        require : 'ngModel',
        link : function(scope, elm, attrs, ctrl) {
            elm.bind('keyup', function() {
```

```
                $http({method: 'GET', url: 'FormValidation.jsp'}).
                success(function(data, status, headers, config) {
                    if(parseInt(data)==0){
                        ctrl.$setValidity('remote',true);
                    }else{
                        ctrl.$setValidity('remote',false);
                    }
                }).
                error(function(data, status, headers, config) {
                    ctrl.$setValidity('remote', false);
                });
            });
        }
    };
});
```

后台JSP代码：

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<%
    //随机成功或者失败！
    double d=Math.random();
    if(d>0.5){
        response.getWriter().write("0");
    }else{
        response.getWriter().write("1");
    }
%>
```

运行效果：



第三个例子示范远程表单校验，代码比较简单，请自己仔细看（注意是如何注入$http服务的）。

自定义输入项

HTML代码：

```
<!doctype html>
<html ng-app="form-example2">
    <head>
        <link href="../bootstrap/css/bootstrap.min.css" rel="stylesheet" media="screen">
        <script src="../angular-1.0.3/angular.js"></script>
        <script src="FormCustom.js"></script>
        <style type="text/css">
            div[contentEditable] {
                cursor: pointer;
                background-color: #D0D0D0;
            }
        </style>
    </head>
    <body>
        <div>
            <div contentEditable="true" ng-model="content" title="Click to edit">Some</div>
            <pre>model = {{content}}</pre>
        </div>
    </body>
</html>
```

JS代码：

```
angular.module('form-example2', []).directive('contenteditable', function() {
    return {
        require : 'ngModel',
        link : function(scope, elm, attrs, ctrl) {
            // view -> model
            elm.bind('keyup', function() {
                scope.$apply(function() {
                    ctrl.$setViewValue(elm.text());
                });
            });

            // model -> view
            ctrl.$render = function() {
                elm.html(ctrl.$viewValue);
            };

            // load init value from DOM
            ctrl.$setViewValue(elm.html());
        }
```

```
    };
});
```

运行效果：

这个例子是从官方的文档修改而来。

使用这种方式可以用DIV来模拟input，从而可以定义出绚丽的表单效果。

版权申明