

# AngularJS ng-model 指令

ng-model 指令用于绑定应用程序数据到 HTML 控制器(input, select, textarea)的值。

## ng-model 指令

ng-model 指令可以将输入域的值与 AngularJS 创建的变量绑定。

### AngularJS 实例

```
<div ng-app="myApp" ng-controller="myCtrl">
  名字: <input ng-model="name">
</div>

<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
  $scope.name = "John Doe";
});
</script>
```

[尝试一下 »](#)

## 双向绑定

双向绑定，在修改输入域的值时，AngularJS 属性的值也将修改：

### AngularJS 实例

```
<div ng-app="myApp" ng-controller="myCtrl">
  名字: <input ng-model="name">
  <h1>你输入了: {{name}}</h1>
</div>
```

[尝试一下 »](#)

## 验证用户输入

### AngularJS 实例

```
<form ng-app="" name="myForm">
  Email:
  <input type="email" name="myAddress" ng-model="text">
  <span ng-show="myForm.myAddress.$error.email">不是一个合法的邮箱地址</span>
</form>
```

[尝试一下 »](#)

以上实例中，提示信息会在 `ng-show` 属性返回 `true` 的情况下显示。

## 应用状态

`ng-model` 指令可以为应用数据提供状态值(`invalid`, `dirty`, `touched`, `error`):

### AngularJS 实例

```
<form ng-app="" name="myForm" ng-init="myText = 'test@runoob.com'">
  Email:
  <input type="email" name="myAddress" ng-model="myText" required></p>
  <h1>状态</h1>
  {{myForm.myAddress.$valid}}
  {{myForm.myAddress.$dirty}}
  {{myForm.myAddress.$touched}}
</form>
```

尝试一下 »

## CSS 类

`ng-model` 指令基于它们的状态为 HTML 元素提供了 CSS 类：

### AngularJS 实例

```
<style>
input.ng-invalid {
  background-color: lightblue;
}
</style>
<body>

<form ng-app="" name="myForm">
  输入你的名字:
  <input name="myAddress" ng-model="text" required>
</form>
```

尝试一下 »

`ng-model` 指令根据表单域的状态添加/移除以下类：

- `ng-empty`
- `ng-not-empty`
- `ng-touched`
- `ng-untouched`
- `ng-valid`
- `ng-invalid`
- `ng-dirty`

- ng-pending
- ng-pristine

← AngularJS API

AngularJS Scope(作用域) →



3 篇笔记

 写笔记