



## Привязка данных в AngularJS

Последнее обновление: 1.11.2015



G+1



Изучай английский по Skype!

Привязка данных раскрывается через ряд директив:

- **ng-bind**: осуществляет привязку к свойству `innerText` html-элемента
- **ng-bind-html**: осуществляет привязку к свойству `innerHTML` html-элемента
- **ng-bind-template**: аналогична `ng-bind` за тем исключением, что позволяет установить привязку сразу к нескольким выражениям
- **ng-model**: создает двустороннюю привязку
- **ng-non-bindable**: определяет участок html-кода, в котором привязка не будет использоваться

Рассмотрим директивы на примере того же контроллера `phonesController`:

```
var myApp=angular.module('myApp', []);
myApp.controller('phoneController', function($scope) {

    $scope.phones = [{
        name: 'Nokia Lumia 630',
        year: 2014,
        price: 200,
        company: {
            name: 'Nokia',
            country: 'Финляндия'
        }
    }],{
        name: 'Samsung Galaxy S 4',
        year: 2014,
        price: 400,
        company: {
            name: 'Samsung',
            country: 'Республика Корея'
        }
    }
});
```

```
    }  
  }, {  
    name: 'Mi 5',  
    year: 2015,  
    price: 300,  
    company: {  
      name: 'Xiaomi',  
      country: 'Китай'  
    }  
  }  
}  
});
```

## ng-bind

Директива `ng-bind` используется для создания односторонней привязки одного элемента html к какому-нибудь выражению. И если данное выражение изменится, то вместе с ним обновится и содержимое связанного элемента html. Например, в прошлом разделе мы использовали массив моделей `phone`. Свяжем элемент с одним из таких объектов:

```
<div ng-controller="phoneController">  
  <p>Количество элементов: <span ng-bind="phones.length"></span></p>  
  <p>Первый элемент: <span ng-bind="phones[0].name"></span></p>  
</div>
```

Здесь идет привязка к длине массива `phones` и к первому его элементу. Если мы добавим в массив новый элемент или изменим название первого элемента, то соответственно изменится и текст в привязанных элементах `span`.

## ng-model

Директива `ng-model` позволяет выполнять двустороннюю привязку как к свойствам объекта, так и к вложенным свойствам:

```
<input type="text" ng-model="property" />  
<input type="text" ng-model="object.property" />  
<input type="text" ng-model="object.property.property" />
```

Директиву `ng-model` можно использовать для привязки к элементам `input`, `textarea` и `select`:

```
<div ng-controller="phoneController">  
  <input type="text" ng-model="phones[0].name" />  
  <p ng-bind="phones[0].name"></p>  
</div>
```

При изменении текста в поле ввода синхронно будет изменяться и привязанный к `phones[0].name` элемент `p`.

## ng-bind-template

Директива `ng-bind-template` позволяет задать шаблон привязки с использованием

выражений. Например:

```
<div ng-controller="phoneController">
<p ng-bind-template="{{phones[0].name}}: {{phones[0].price}} $"></p>
</div>
```

## ng-non-bindable

Но бывает, что нам надо, чтобы AngularJS не обрабатывал какие-то выражения и не использовал привязку. В этом случае мы можем применить директиву `ng-non-bindable`:

```
<p>{{1 + 2}}</p>
<p ng-non-bindable>{{1 + 2}}</p>
```

Если в первом случае AngularJS обработает выражение и выведет на страницу его результат, то во втором случае выражение будет выведено как есть.

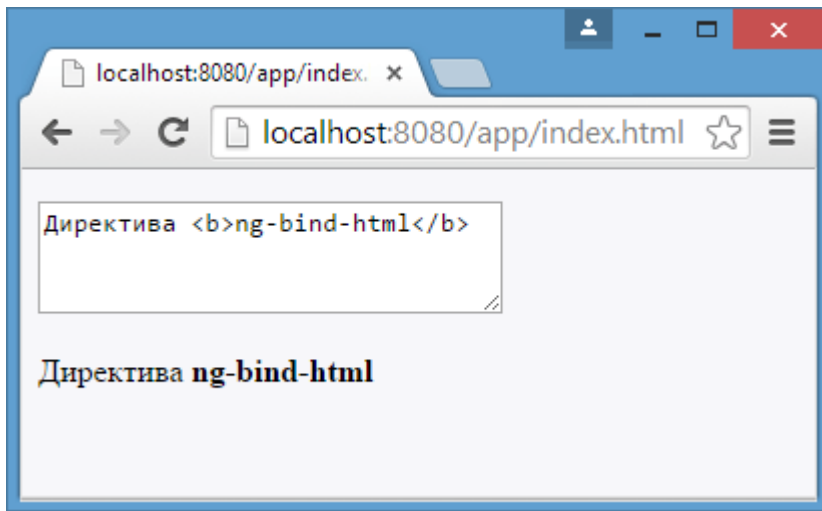
## ng-bind-html

Директива `ng-bind-html` позволяет выводить на страницу html-содержимое со всеми тегами. При использовании этой директивы необходимо помнить, результатом вывода может быть потенциально небезопасный код, например, содержащий javascript, поэтому данный аспект надо учитывать.

Чтобы осуществить привязку к коду html, надо подключить дополнительный файл. В архиве библиотеки, который можно найти на официальном сайте, имеется файл *angular-sanitize.js*, а также его минимизированный аналог. Положим этот файл в одну папку с главной библиотекой *angularjs* и изменим код страницы следующим образом:

```
<!doctype html>
<html ng-app="myApp">
<head>
<meta charset="utf-8">
</head>
<body>
<div ng-controller="helloController">
<p><textarea ng-model="htmlcode"></textarea></p>
<p ng-bind-html="htmlcode"></p>
</div>
<script src="js/lib/angular.min.js"></script>
<script src="js/lib/angular-sanitize.min.js"></script>
<script type="text/javascript">
var myApp = angular.module('myApp', ['ngSanitize']);
myApp.controller('helloController', function($scope) {

    $scope.htmlcode="Директива <b>ng-bind-html</b>";
});
</script>
</body>
</html>
```



Чтобы задействовать модуль `angular-sanitize`, надо при создании модуля установить зависимость от `angular-sanitize`: `var myApp = angular.module('myApp', ["ngSanitize"]);`. После этого мы сможем изменять код в текстовом поле, в том числе вносить код `html`, и все изменения автоматически будут применяться к веб-странице.

## Galaxy S7 till superpris

Samsung Galaxy S7 med Fastpris. Allt ingår för 335 kr/mån!



[Назад](#) [Содержание](#) [Вперед](#)



12 Комментариев

metanit.com

 Войти Рекомендовать Поделиться

Лучшее в начале



Присоединиться к обсуждению...

**Егор** • 6 месяцев назад

Почему может не работать привязка данных? Т.е. есть модель, есть представление. При начальной загрузке все привязки работают, данные модели отображаются. Но при изменении данных модели, данные меняются (видно в debug), но представление по-прежнему отображает старые данные, указанные при инициализации. Такая-же история и с двусторонней привязкой через input. В чем может быть проблема ?

  • Ответить • Поделиться ›**keeprock** → Егор • 20 дней назад

Есть еще такой вариант что у вас нет тот scope. Это надо смотреть через ng-inspector или Batarang. Допустим, если поставить ng-if, то внутри этого блока будет свой собственный \$scope, который может отличаться от \$scope контроллера в котором ng-if объявлен.

Поставьте ng-inspector на Chrome - так будет веселее работать. И почитайте про «Controller As», \$rootScope и parent.

  • Ответить • Поделиться ›**Shepot** → Егор • 4 месяца назад

Возможно не правильный порядок подключенных файлов.

  • Ответить • Поделиться ›**TheBububo** . • 9 месяцев назад

чем отличается `span ng-bind="phones[0].name" /span` от `span {{phones[0].name}} /span` ?  
Что в каких случаях используется?

  • Ответить • Поделиться ›**Lime** → TheBububo . • 7 месяцев назад

Разница в том, что в случае ошибки атрибут не будет отображен. Попробуй вывести выражением значение отсутствующее в scope.

  • Ответить • Поделиться ›**Metanit** Модератор → TheBububo . • 9 месяцев назад

в данном случае разницы большой нет  
но если обратиться к документации, то там сказано  
"Typically, you don't use ngBind directly, but instead you use the double curly markup like `{{ expression }}` which is similar but less verbose.

It is preferable to use ngBind instead of `{{ expression }}` if a template is momentarily displayed by the browser in its raw state before Angular compiles it."

Перевод

---

[r](#) | [Google+](#) | [Канал сайта на youtube](#) | [Помощь сайту](#)

✉: metanit22@mail.ru

t.com, 2012-2016. Все права защищены.