



## Контроллеры AngularJS

Последнее обновление: 1.11.2015

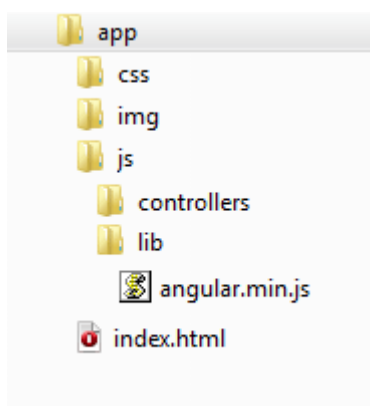


G+1



Изучай английский по Skype!

Прежде чем перейти к освоению основ AngularJS, сразу обозначим общую структуру приложения. Создадим в корневом каталоге веб-сервера (в качестве веб-сервера можно выбрать, например, NodeJS или Apache) папку *app*, в которой будут следующие подкаталоги:



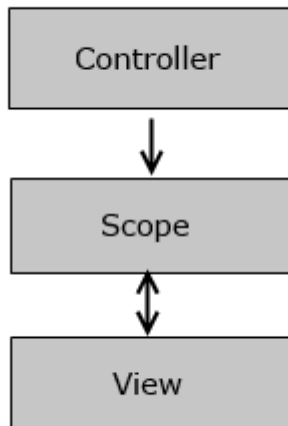
Папка *css* будет содержать используемые стили, *img* - изображения, *js* - скрипты. В папке *js* создадим два каталога: *controllers* - для контроллеров и *lib*, в который поместим собственно библиотеку *angular.min.js* и в который в дальнейшем будем класть и другие сопроводительные скрипты.

Также в папке *app* у нас будет находиться веб-страничка *index.html* - само представление.

Для создания контроллера используется метод **controller(name, constructor)**, который определен у объекта *Module*. Первый параметр передает имя контроллера, а второй - функцию, которая выполняет настройку контроллера. Эту функцию еще называют конструктором:

```
var myApp=angular.module('myApp', []);
myApp.controller("phonesController", function ($scope) {
    // настройка контроллера
});
```

AngularJS предоставляет ряд встроенных сервисов, название которых начинается с символа \$. В функции контроллера в качестве параметра передается сервис \$scope, через который контроллер передает данные в представление.



Теперь создадим и применим контроллер. Для этого перейдем к веб-странице index.html. Изменим ее код следующим образом:

```
<!doctype html>
<html ng-app="myApp">
<head>
<meta charset="utf-8">
</head>
<body>
<div ng-controller="phoneController">
<p>Название: {{phone.name}}</p>
<p>Цена: {{phone.price}} $</p>
<p>Производитель: {{phone.company.name}}</p>
</div>
<script src="js/lib/angular.min.js"></script>
<script type="text/javascript">
var myApp=angular.module('myApp', []);
myApp.controller('phoneController', function($scope) {

    $scope.phone = {

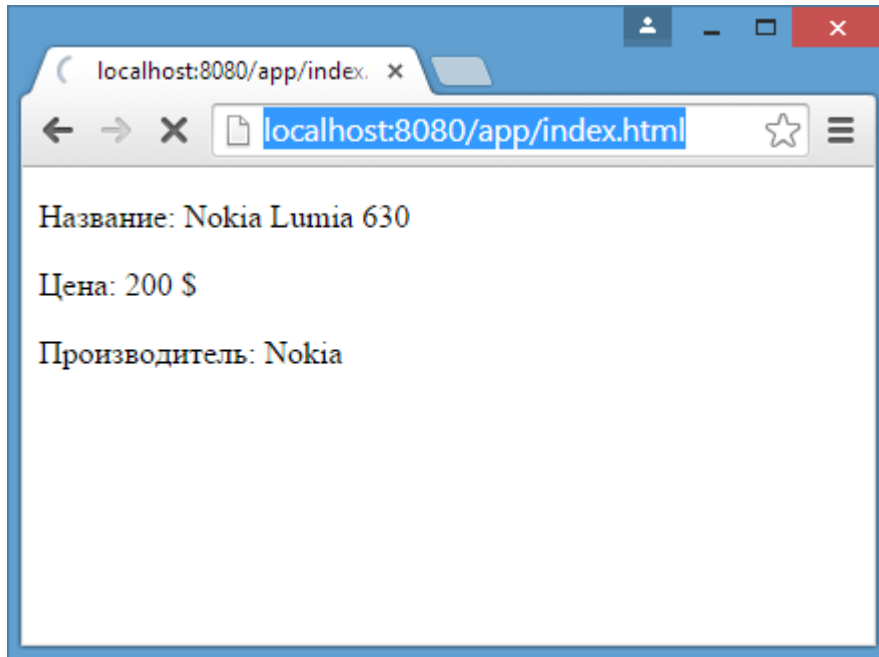
        name: 'Nokia Lumia 630',
        year: 2014,
        price: 200,
        company: {
            name: 'Nokia',
            country: 'Финляндия'
        }
    }
});
</script>
</body>
```

```
</html>
```

Здесь в контроллере через объект `$scope` передается некоторый объект `phone`, который содержит обычный javascript-объект с информацией о смартфоне.

Чтобы применить контроллер к определенному участку разметки html, надо использовать директиву **ng-controller**. После этого данный кусок разметки html будет обозначать представление контроллера.

Запустим в веб-браузере страницу:



С помощью пары двойных фигурных скобок мы создаем **выражения**. Выражения в AngularJS представляют вкрапления кода javascript в разметку html. В выражении не обязательно обращаться к свойствам переданного через `$scope` объекта. Например, мы можем передать математическое выражение: `<p>2+3={{2+3}}</p>`. И вместо данного выражения браузер выведет нам его результат.

Теперь отделим контроллер от представления. Для этого в ранее созданную папку `controllers` добавим следующий файл `phonesController.js`:

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

    $scope.phone = {

        name: 'Nokia Lumia 630',
        year: 2014,
        price: 200,
        company: {
            name: 'Nokia',
            country: 'Финляндия'
        }
    }
});
```

Обратите внимание, что здесь мы не создаем, а получаем ранее созданный модуль (`angular.module('myApp')`), так как метод `module` использует только один параметр.

И также изменим файл `index.html`:

```
<!doctype html>
<html ng-app="myApp">
<head>
<meta charset="utf-8">
</head>
<body>
<div ng-controller="phoneController">
<p>Название: {{phone.name}}</p>
<p>Цена: {{phone.price}} $</p>
<p>Производитель: {{phone.company.name}}</p>
</div>
<script src="js/lib/angular.min.js"></script>
<script type="text/javascript">
angular.module('myApp', []);
</script>
<script src="js/controllers/phonesController.js"></script>
</body>
</html>
```

### [Загрузить проект](#)

При этом мы можем использовать несколько представлений для одного контроллера в разных местах `html`-страницы:

```
<!doctype html>
<html ng-app="myApp">
<head>
<meta charset="utf-8">
</head>
<body>
<div ng-controller="phoneController">
<p>Название: {{phone.name}}</p>
<p>Цена: {{phone.price}} $</p>
<p>Производитель: {{phone.company.name}}</p>
</div>
<hr>
<div ng-controller="phoneController">
<p>Производитель: {{phone.company.country}}</p>
</div>
<script src="js/lib/angular.min.js"></script>
<script type="text/javascript">
angular.module('myApp', []);
</script>
<script src="js/controllers/phonesController.js"></script>
</body>
</html>
```

Хотя тут используется один контроллер, но для каждого представления, использующего директиву `ng-controller`, будет создаваться свой отдельный объект контроллера и соответственно свой объект `$scope`.

## Наследование контроллеров

Для наследования контроллеров нам надо вложить один элемент с директивой `ng-controller` в другой, который также имеет подобную директиву:

```
<div ng-controller="parentController">
  <div ng-controller="firstChildController"></div>
  <div ng-controller="secondChildController"></div>
</div>
```

## Контроллеры без использования \$scope

Нам не обязательно использовать объект `$scope` для передачи данных из контроллера в представления. Важно понимать, что контроллер - это также javascript-объект, который работает также как и другие js-объекты:

```
<!doctype html>
<html ng-app="myApp">
<head>
<meta charset="utf-8">
</head>
<body>
<div ng-controller="myController as myCtrl">
  <h3>{{myCtrl.message}}</h3>
  <p>{{myCtrl.text}}</p>
</div>
<script src="js/lib/angular.min.js"></script>
<script type="text/javascript">
var myApp = angular.module('myApp', []);
myApp.controller('myController', function() {
  this.message="myController";
  this.text="Контроллер без $scope";
});
</script>
</body>
</html>
```

Определение свойств внутри контроллера также, как и в любом js-объекте, можно произвести с помощью ключевого слова `this`:

```
this.message="myController"
```

Затем при применении контроллера нам надо указать переменную, через которую мы будем взаимодействовать с контроллером:

```
<div ng-controller="myController as myCtrl">
```

В данном случае теперь мы сможем обращаться к контроллеру через `myCtrl`. И после этого мы также сможем использовать переданные значения в выражениях AngularJS.

# Galaxy S7 till superpris

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



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



26 Комментариев

metanit.com

 Войти Рекомендовать Поделиться

Лучшее в начале



Присоединиться к обсуждению...

**Михаил** • 2 месяца назад

Здравствуйте, вопрос такой:

Правильно понимать под представлением, область html: `<div ng-controller="someController">//TODO</div>`. Таким образом на одном документе может быть несколько представлений разных контроллеров? Из этого, представление привязано к отдельно взятому контроллеру, верно ли так рассуждать?

  • Ответить • Поделиться ›**Metanit** Модератор → Михаил • 2 месяца назад

скажу так, на странице может использоваться несколько контроллеров, и каждый по-своему может влиять на содержимое страницы

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

Первое что бы хотелось сказать автору, огромное спасибо! Все работает великолепно и у Вас огромный талант в объяснении сложных вещей - простым языком. За это низкий поклон Вам.

Ну чисто для себя, потестил наследование и нашел интересный момент, а именно: Если наследовать контроллер, то все свойства родителя объявлены через `this` НЕ наследуются. Я так понимаю так и должно быть? А свойства объявлены глобально в скоуп наследуются. Ну я так понимаю это идет из особенностей джаваскрипт объектов, что это свойство касается только текущего объекта.

  • Ответить • Поделиться ›**Pavel** • 5 месяцев назад

Скажите, в чем смысл подключать библиотеку ангуляра после `div`-а с контроллером? Почему ее не подключать в `head`? В вашем случае при загрузке страницы сначала отображается `{{phone.name}}`, а потом, через долю секунды - Nokia Lumia 630

  • Ответить • Поделиться ›**Metanit** Модератор → Pavel • 5 месяцев назад

как правило, файлы `js` подключаются перед закрывающим тегом `body` для ускорения загрузки страницы. Но если вам приносит неудобство такой медленный рендеринг, то можно и в `head` подключать

  • Ответить • Поделиться ›**Johnny Juarez** • 6 месяцев назад

со всем уважением Вам, и с признательностью за старания просвещать людей, но мне кажется, что здесь ошибка...

Если скопипастить содержимое от начала и до конца, не изменяя абсолютно ничего, то получим:

---

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

✉: metanit22@mail.ru

metanit.com, 2012-2016. Все права защищены.