

Что такое AngularJS и с чем его едят?

AngularJS — JavaScript-фреймворк с открытым исходным кодом. Предназначен для разработки одностраничных приложений.

AngularJS разработан в 2009 году сотрудником Google Мишко Хавери. В 2012 году появился AngularJS 1.0. Идея оказалась удачной и проект теперь официально поддерживается Google.

AngularJS – это JavaScript-фреймворк

Фреймворк подразумевает под собой то, что вместо того, чтобы писать весь код самому разработчик “играет по заранее оговоренным правилам”. Эти правила – это стандарты и инструменты выбранного фреймворка. Это библиотека, написанная на JavaScript. AngularJS добавляется на веб-страницу с помощью тега `<script>`:

AngularJS приложения

Основные части AngularJS приложения - это модули и контроллеры. Рассмотрим это на примере следующего приложения.

AngularJS модули определяют AngularJS приложение. Они содержат различные части приложения. Чтобы создать модуль, нужно использовать AngularJS функцию `angular.module`:

```
<div ng-app="myApp">...</div>
<script>
var app = angular.module("myApp", []);
</script>
```

Параметр «myApp» обращается к HTML-элементу, в котором запущено приложение. Директива `ng-app` используется для инициализации приложения.

Контроллеры управляют работой приложения. Добавление контроллера из примера, что был сверху:

```
<div ng-app="myApp" ng-controller="myCtrl">
{{ firstName + " " + lastName }}
</div>
<script>
var app = angular.module("myApp", []);
app.controller("myCtrl", function($scope) {
    $scope.firstName = "John";
    $scope.lastName = "Doe";
});
</script>
```

Директива `ng-controller` используется для обращения к контроллеру.

AngularJS директивы

AngularJS позволяет расширить возможности HTML с помощью новых атрибутов - Директив. Возможно использовать как встроенные директивы, так и написать свои собственные. Директивы начинаются с префикса –ng. В примере выше были использованы:

- ng-app инициализирует приложение,
- ng-controller определяет объект контроллер для приложения,
- ng-model связывает значения из HTML элементов контроля с приложением,

AngularJS выражения

Выражения AngularJS – это фрагмент кода, который прописывается внутри двойных фигурных скобок: {{ expression }} или внутри директивы: ng-bind="expression".

Первый пример – выражение в двойных скобках (есть на сайте), а второй – выражение в директиве.

AngularJS разрешит выражение и вернет результат именно там, где написано данное выражение.

Двухстороннее связывание данных в AngularJS

Двухстороннее связывание данных в AngularJS – синхронизация между моделью и представлением. Рассмотрим это свойство на следующем примере.

Модель данных – набор доступных приложению данных. Например:

```
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.firstname = "John";
    $scope.lastname = "Doe";
});
```

Представление – HTML-контейнер, где отображается приложение AngularJS. Оно имеет доступ к модели. Для отображения модели в представлении можно использовать директиву ng-model:

```
<input ng-model="firstname">
```

Двухсторонняя привязка данных означает, что когда меняется модель, представление также изменяется. Это происходит незамедлительно и автоматически, и позволяет быть уверенным, что модель и представления всегда обновлены.

Заключение

Таким образом, при изучении AngularJS были выделены следующие достоинства.

1. Двухстороннее связывание данных.
2. Выражения. Чтобы что-то вывести в содержимое какого-либо объекта, не требуется писать в шаблоне громоздкие конструкции. Достаточно заключить выражение в двойные фигурные скобки, и все, данные будут обработаны.
3. Директивы. Директивы расширяют синтаксис HTML.

Несмотря на вышеупомянутые особенности AngularJS, у него есть и несколько недостатков:

1. Сложность освоения. Трудности обычно возникают у тех, кто раньше использовал библиотеку jQuery, ведь в отличие от Angular.js она полагается на выполнение манипуляций с деревом DOM.
2. Отсутствие обратной совместимости со второй версией. Разумеется, вы можете начать подготовку своего кода к миграции уже сейчас, но гарантий, что она пройдет гладко, нет.