




HTML enhanced for web apps!

 View on GitHub

 Download (1.2.27 / 1.3.4)

Follow +AngularJS on 

 Follow @angularjs 53.1K followers

 Tweet 4,933



Learn Angular in your browser for free!

Why AngularJS?

Alternatives

Extensibility

HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop.

Other frameworks deal with HTML's shortcomings by either abstracting away HTML, CSS, and/or JavaScript or by providing an imperative way for manipulating the DOM. Neither of these address the root problem that HTML was not designed for dynamic views.

AngularJS is a toolset for building the framework most suited to your application development. It is fully extensible and works well with other libraries. Every feature can be modified or replaced to suit your unique development workflow and feature needs. Read on to find out how.



ng-europe 2014

ng-conf 2014



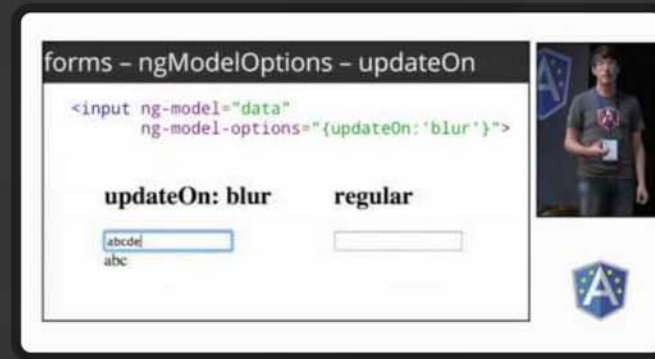
Keynote on the state of Angular by
Igor Minar, Brad Green and Judy
Tuan



Miško Hevery - Keynote on
AtScript



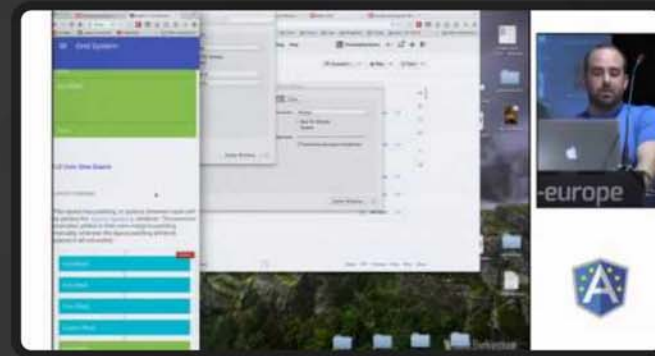
Angular 2.0 Core by Igor Minar & Tobias Bosch



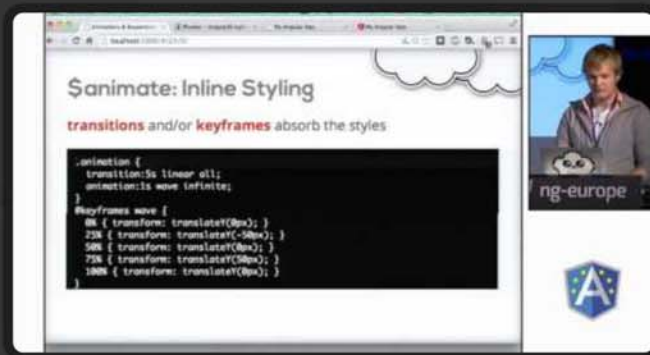
Angular 1.3 by Jeff Cross & Brian Ford



Q&A with Angular team on 2.0, AtScript & more



Material Design by Thomas Burleson & Max Lynch



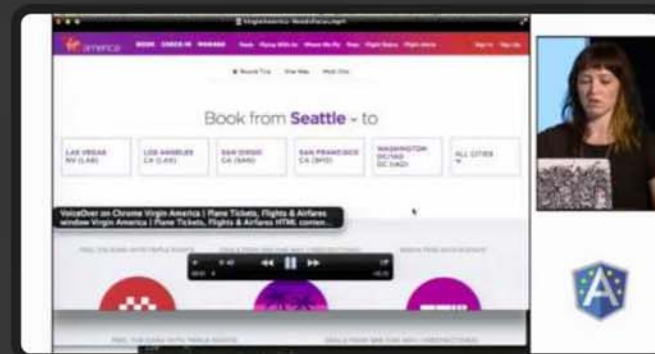
Animations (sequencer, web animations) by Matias Niemelä
aka yearofmoo



Protractor and the Testability API
by Julie Ralph & Chirayu
Krishnappa



The new Router for AngularJS by
Rob Eisenberg



AngularJS Accessibility by Marcy
Sutton



Tooling by Brian Ford

[view all videos](#)

www.ngeurope.org

The Basics

index.html

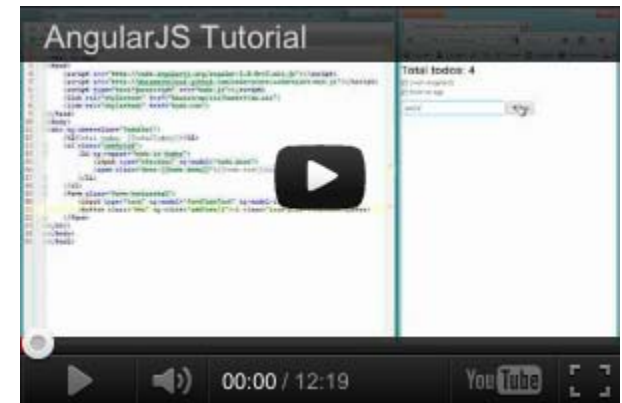
Hint: Click [me](#).

[Edit Me](#)

Name:

Hello !

Watch as we build this app



```
1. <!doctype html>
2. <html ng-app>
3.   <head>
4.     <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.4/
angular.min.js"></script>
5.   </head>
6.   <body>
7.     <div>
8.       <label>Name:</label>
9.       <input type="text" ng-model="yourName" placeholder="Enter a name here">
10.     <hr>
11.     <h1>Hello {{yourName}}!</h1>
12.   </div>
13. </body>
14. </html>
```

Add Some Control

Data Binding

Data-binding is an automatic way of updating the view whenever the model changes, as well as updating the model whenever the view changes. This is awesome because it eliminates DOM manipulation from the list of things you have to worry about.

Controller

Controllers are the behavior behind the DOM elements. AngularJS lets you express the behavior in a clean readable form without the usual boilerplate of updating the DOM, registering callbacks or watching model changes.

Plain JavaScript

Unlike other frameworks, there is no need to inherit from proprietary types in order to wrap the model in accessor methods. Angular models are plain old JavaScript objects. This makes your code easy to test, maintain, reuse, and again free from boilerplate.

[index.html](#)

[todo.js](#)

[todo.css](#)

Hint: Click [me](#) .

[Edit Me](#)


```

1. <!doctype html>
2. <html ng-app="todoApp">
3.   <head>
4.     <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.4/
angular.min.js"></script>
5.     <script src="todo.js"></script>
6.     <link rel="stylesheet" href="todo.css">
7.   </head>
8.   <body>
9.     <h2>Todo</h2>
10.    <div ng-controller="TodoController">
11.      <span>{{remaining()}} of {{todos.length}} remaining</span>
12.      [ <a href="" ng-click="archive()">archive</a> ]
13.      <ul class="unstyled">
14.        <li ng-repeat="todo in todos">
15.          <input type="checkbox" ng-model="todo.done">
16.          <span class="done-{{todo.done}}">{{todo.text}}</span>
17.        </li>
18.      </ul>
19.      <form ng-submit="addTodo()">
20.        <input type="text" ng-model="todoText" size="30"
21.          placeholder="add new todo here">
22.        <input class="btn-primary" type="submit" value="add">
23.      </form>
24.    </div>
25.  </body>
26. </html>

```

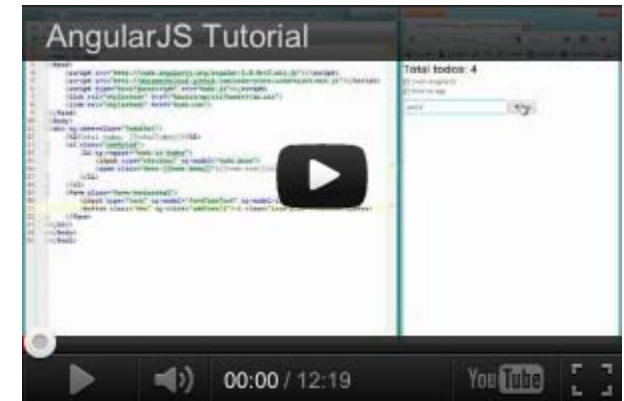
Todo

1 of 2 remaining [[archive](#)]

☒ learn angular

☐ build an angular app

Watch as we build this app



Wire up a Backend

Deep Linking

A deep link reflects where the user is in the app,

Form Validation

Client-side form validation is an important part of

Server Communication

AngularJS provides built-in services on top of XHR

this is useful so users can bookmark and email links to locations within apps. Round trip apps get this automatically, but AJAX apps by their nature do not. AngularJS combines the benefits of deep link with desktop app-like behavior.

great user experience. AngularJS lets you declare the validation rules of the form without having to write JavaScript code. Write less code, go have beer sooner.

as well as various other backends using third party libraries. Promises further simplify your code by handling asynchronous return of data. In this example, we use the AngularFire library to wire up a [Firebase](#) backend to a simple Angular app.

index.html bootstrap.css project.js list.html detail.html

Hint: Click [me](#).

[Edit Me](#)

```
1. <!doctype html>
2. <html ng-app="project">
3.   <head>
4.     <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.4/
angular.min.js"></script>
5.     <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.4/
angular-resource.min.js">
6.     </script>
7.     <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.4/
angular-route.min.js">
8.     </script>
9.     <script src="https://cdn.firebase.com/js/client/2.0.4/firebase.js
"></script>
10.    <script src="https://cdn.firebase.com/libs/angularfire/0.9.0/
angularfire.min.js"></script>
11.    <link rel="stylesheet" href="bootstrap.css">
12.    <script src="project.js"></script>
13.  </head>
14.  <body>
15.    <h2>JavaScript Projects</h2>
16.    <div ng-view></div>
17.  </body>
18. </html>
```

JavaScript Projects

Project	Description	+
backspace e.V.	hackerspace bamberg	✎
outro teste	Uma descrição	✎
pop	ddshghj	✎
Science	sdfsd	✎

Create Components

[index.html](#)[components.js](#)[app.js](#)[bootstrap.css](#)Hint: Click [me](#).[Edit Me](#)

```
1. <!doctype html>
2. <html ng-app="app">
3.   <head>
4.     <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.4
      /angular.min.js"></script>
5.     <script src="components.js"></script>
6.     <script src="app.js"></script>
7.     <link rel="stylesheet" href="bootstrap.css">
8.   </head>
9.   <body>
10.    <tabs>
11.      <pane title="Localization">
12.        Date: {{ '2012-04-01' | date:'fullDate' }} <br>
13.        Currency: {{ 123456 | currency }} <br>
14.        Number: {{ 98765.4321 | number }} <br>
15.      </pane>
16.      <pane title="Pluralization">
17.        <div ng-controller="BeerCounter">
18.          <div ng-repeat="beerCount in beers">
19.            <ng-pluralize count="beerCount" when="beerForms"></ng-pluralize>
20.          </div>
21.        </div>
22.      </pane>
23.    </tabs>
24.  </body>
25. </html>
```

Locale: US

[Localization](#)[Pluralization](#)

Date: Sunday, April 1, 2012

Currency: \$123,456.00

Number: 98,765.432

Locale: SK

[Localization](#)[Pluralization](#)

Date: nedela, 1. aprila 2012

Currency: 123 456,00 €

Number: 98 765,432

Embeddable

AngularJS works great with other technologies. Add as much or as little of AngularJS to an existing page as you like. Many other frameworks require full commitment. This page has multiple AngularJS applications embedded in it. Because AngularJS has no global state multiple apps can run on a single page without the use of `iframe`s. We encourage you to view-source and look around.

Injectable

The dependency injection in AngularJS allows you to declaratively describe how your application is wired. This means that your application needs no `main()` method which is usually an unmaintainable mess. Dependency injection is also a core to AngularJS. This means that any component which does not fit your needs can easily be replaced.

Testable

AngularJS was designed from ground up to be testable. It encourages behavior-view separation, comes pre-bundled with mocks, and takes full advantage of dependency injection. It also comes with end-to-end scenario runner which eliminates test flakiness by understanding the inner workings of AngularJS.