WIKIPEDIA

AngularJS

AngularJS is a JavaScript-based open-source front-end web framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications.

AngularJS is the frontend part of the MEAN stack, consisting of MongoDB database, Express.js web application server framework, Angular.js itself, and Node.js server runtime environment. Version 1.7.x is on Long Term Support until July 1st 2021. After that date AngularJS will no longer be updated and Angular (2.0+) is suggested instead.[3][4]

AngularJS

Developer(s)	Google
Initial release	October 20, 2010[1]
Stable release	1.7.9 / November 19, 2019[2]
Repository	AngularJS Repository (https://github.com/angular/angular.js)
Written in	JavaScript
Platform	JavaScript engine
Size	167 kB production 1.2 MB development
Type	Web framework
License	MIT License
Website	angularjs.org (https://angularjs.org)

Contents

Overview

The AngularJS framework works by first reading the Hypertext Markup Language (HTML) page, which has an additional custom HTML attributes embedded into it. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

AngularJS is built on the belief that declarative programming should be used to create user interfaces and connect software components, while imperative programming is better suited to defining an application's business logic.[5] The framework adapts and extends traditional HTML to present dynamic content through two-way data-binding that allows for the automatic synchronization of models and views. As a result, AngularJS de-emphasizes explicit Document Object Model (DOM) manipulation with the goal of improving testability and performance.

AngularJS's design goals include:

- to decouple DOM manipulation from application logic. The difficulty of this is dramatically affected by the way the code is structured.
- to decouple the client side of an application from the server-side. This allows development work to progress in parallel and allows for reuse of both sides.
- to provide structure for the journey of building an application: from designing the UI, through writing the business logic, to testing.

AngularJS implements the MVC pattern to separate presentation, data, and logic components.[6] Using dependency injection, Angular brings traditionally server-side services, such as view-dependent controllers, to client-side web applications. Consequently, much of the burden on the server can be reduced.

Scope

AngularJS uses the term "scope" in a manner akin to the fundamentals of computer science.

Scope in computer science describes when in the program a particular binding is valid. The ECMA-262 specification defines scope as: a lexical environment in which a Function object is executed in client-side web scripts;[7] akin to how scope is defined in lambda calculus.[8]

As a part of the "MVC" architecture, the scope forms the "Model", and all variables defined in the scope can be accessed by the "View" as well as the "Controller". The scope behaves as a glue and binds the "View" and the "Controller".

Bootstrap

The task performed by the AngularJS bootstrapper occur in three phases[9] after the DOM has been loaded:

1. Creation of a new Injector
2. Compilation of the directives that decorate the DOM
3. Linking of all directives to scope

AngularJS directives allow the developer to specify custom and reusable HTML-like elements and attributes that define data bindings and the behavior of presentation components. Some of the most commonly used directives are:

`ng-animate`
: A module provides support for JavaScript, CSS3 transition and CSS3 keyframe animation hooks within existing core and custom directives.

Since **`ng-*`** attributes are not valid in HTML specifications, **`data-ng-*`** can also be used as a prefix. For example, both **`ng-app`** and **`data-ng-app`** are valid in AngularJS.

`ng-app`
: Declares the root element of an AngularJS application, under which directives can be used to declare bindings and define behavior.

`ng-aria`
: A module for accessibility support of common ARIA attributes.

`ng-bind`
: Sets the text of a DOM element to the value of an expression. For example, `<span ng-bind="name"></span>` displays the value of ‘name’ inside the span element. Any change to the variable ‘name’ in the application's scope reflect instantly in the DOM.

`ng-class`
: Conditionally apply a class, depending on the value of a boolean expression.

`ng-controller`
: Specifies a JavaScript controller class that evaluates HTML expressions.

`ng-if`
: Basic if statement directive that instantiates the following element if the conditions are true. When the condition is false, the element is removed from the DOM. When true, a clone of the compiled element is re-inserted.

`ng-init`
: Called once when the element is initialized.

`ng-model`
: Similar to `ng-bind`, but establishes a two-way data binding between the view and the scope.

`ng-model-options`
: Provides tuning for how model updates are done.

`ng-repeat`
: Instantiate an element once per item from a collection.

`ng-show` & **`ng-hide`**
: Conditionally show or hide an element, depending on the value of a boolean expression. Show and hide is achieved by setting the CSS display style.

`ng-switch`
: Conditionally instantiate one template from a set of choices, depending on the value of a selection expression.

`ng-view`
: The base directive responsible for handling routes[10] that resolve JSON before rendering templates driven by specified controllers.

Two-way data binding

AngularJS two-way data binding is its most notable feature, largely relieving the server backend of templating responsibilities. Instead, templates are rendered in plain HTML according to data contained in a scope defined in the model. The `$scope` service in Angular detects changes to the model section and modifies HTML expressions in the view via a controller. Likewise, any alterations to the view are reflected in the model. This circumvents the need to actively manipulate the DOM and encourages bootstrapping and rapid prototyping of web applications.[11] AngularJS detects changes in models by comparing the current values with values stored earlier in a process of dirty-checking, unlike Ember.js and Backbone.js that trigger listeners when the model values are changed.[12]

`$watch`
: is an angular method used for dirty checking. Any variable or expression assigned in $scope automatically sets up a `$watchExpression` in angular. So assigning a variable to `$scope` or using directives like `ng-if, ng-show, ng-repeat` etc. all create watches in angular

scope automatically. Angular maintains a simple array of `$$watchers` in the `$scope` objects
Different ways of defining a watcher in AngularJS.

- explicitly $watch an attribute on `$scope`.

```
$scope.$watch('person.username', validateUnique);
```

- place an interpolation in your template (a watcher will be created for you on the current $scope).
- ask a directive such as `ng-model` to define the watcher for you.

```
<input ng-model="person.username" />
```

`$digest`
is angular method that is invoked internally by angularjs in frequent intervals. In `$digest` method, angular iterates over all `$watches` in its scope/child scopes.

`$apply`
is an angular method that internally invokes `$digest`. This method is used when you want to tell angular manually start dirty checking (execute all `$watches`)

`$destroy`
is both a method and event in angularjs. `$destroy()` method, removes a scope and all its children from dirty checking. `$destroy` event is called by angular whenever a $scope or $controller is destroyed.

Development history

AngularJS was originally developed in 2009 by Miško Hevery[13] at Brat Tech LLC[14] as the software behind an online JSON storage service, that would have been priced by the megabyte, for easy-to-make applications for the enterprise. This venture was located at the web domain "GetAngular.com",[14] and had a few subscribers, before the two decided to abandon the business idea and release Angular as an open-source library.

The 1.6 release added many of the concepts of Angular to AngularJS, including the concept of a component-based application architecture.[15] This release among others removed the Sandbox, which many developers believed provided additional security, despite numerous vulnerabilities that had been discovered that bypassed the sandbox.[16] The current (as of March 2020) stable release of AngularJS is 1.7.9[17]

In January 2018, a schedule was announced for phasing-out AngularJS: after releasing 1.7.0, the active development on AngularJS will continue till June 30, 2018. Afterwards, 1.7 will be supported till June 30, 2021 as long-term support.[18]

Legacy browser support

Versions 1.3 and later of AngularJS do not support Internet Explorer 8 or earlier. While AngularJS 1.2 supports IE8, its team does not.[19][20]

Angular and AngularDart

Subsequent versions of AngularJS are simply called Angular. Angular is an incompatible TypeScript-based rewrite of AngularJS. Angular 4 was announced on 13 December 2016, skipping 3 to avoid a confusion due to the misalignment of the router package's version which was already distributed as v3.3.0.[21]

AngularDart works on Dart, which is an object-oriented, class defined, single inheritance programming language using C style syntax, that is different from Angular JS (which uses JavaScript) and Angular 2/ Angular 4 (which uses TypeScript). Angular 4 released in March 2017, with the framework's version aligned with the version number of the router it used. Angular 5 was released on November 1, 2017.[22] Key improvements in Angular 5 include support for progressive Web apps, a build optimizer and improvements related to Material Design.[23] Angular 6 was released on 3rd May 2018[24], Angular 7 was released on 18th October 2018, and Angular 8 was released on May 28, 2019. Angular follows Semantic Versioning standards, with each major version number indicating potentially breaking changes. Angular has pledged to provide 6 months of active support for each major version followed by 12 months of long term support. Major releases are bi-yearly with 1 to 3 minor releases for every major release.[25]

Angular Universal

A normal Angular application executes in the browser, while Angular Universal generates static application pages on the server through server-side rendering (SSR).[26]

Libraries

Angular Material

Angular Material is a UI component library that implements Material Design in AngularJS.[27]

Chrome extension

In July 2012, the Angular team built an extension for the Google Chrome browser called Batarang,[28] that improves the debugging experience for web applications built with Angular. The extension aims to allow for easy detection of performance bottlenecks and offers a GUI for debugging applications.[29] For a time during late 2014 and early 2015, the extension was not compatible with recent releases (after v1.2.x) of Angular.[30] The last update made to this extension was on April 4, 2017.

Performance

AngularJS sets out the paradigm of a *digest cycle*. This cycle can be considered a loop, during which AngularJS checks if there is any change to all the variables watched by all the `$scopes`. If `$scope.myVar` is defined in a controller and this variable was marked for watching, Angular will monitor the changes on myVar in each loop iteration.

This approach potentially leads to slow rendering when AngularJS checks on too many variables in the `$scope` every cycle. Miško Hevery suggests keeping fewer than 2000 watchers on any page.[12]

See also

- React.js

- Vue.js
- Polymer (library)
- Comparison of JavaScript frameworks

References

1. Earliest known releases (https://github.com/angular/angular.js/releases?after=v0.9.4)
2. "Releases" (https://github.com/angular/angular.js/releases). *GitHub*.
3. https://docs.angularjs.org/misc/version-support-status
4. https://blog.angular.io/stable-angularjs-and-long-term-support-7e077635ee9c
5. "What Is Angular?" (https://docs.angularjs.org/guide/introduction). Retrieved 12 February 2013.
6. Understanding Components (https://docs.angularjs.org/guide/component)
7. "Annotated ECMAScript 5.1, Section 10.2 Lexical Environments" (https://es5.github.io/#x10.2). Retrieved 2015-01-03.
8. Barendregt, Henk; Barendsen, Erik (March 2000), *Introduction to Lambda Calculus* (ftp://ftp.cs.ru.nl/pub/CompMath.Found/lambda.pdf) (PDF)
9. "Writing Directives" (https://www.youtube.com/watch?v=WqmeI5fZcho&list=TLJUxRYO87UWA). angularjs.org. November 28, 2012. Retrieved 2013-07-21.
10. Component Router (https://docs.angularjs.org/guide/component-router)
11. "5 Awesome AngularJS Features" (http://net.tutsplus.com/tutorials/javascript-ajax/5-awesome-angularjs-features/). Retrieved 13 February 2013.
12. Misko Hevery. "Databinding in angularjs" (https://stackoverflow.com/a/9693933/146423). Retrieved 2014-03-09.
13. "Hello World, <angular/> is here" (http://misko.hevery.com/2009/09/28/hello-world-angular-is-here/). Retrieved 2014-10-12.
14. "GetAngular" (https://web.archive.org/web/20100413141437/http://getangular.com/). Angular / BRAT Tech. LLC. Archived from the original (http://getangular.com/) on 2010-04-13. Retrieved 2014-10-12.
15. "AngularJS: Developer Guide for v1.5.8: Components" (https://code.angularjs.org/1.5.8/docs/guide/component). Google. Retrieved 2017-09-26.
16. "angular.js/CHANGELOG.md" (https://github.com/angular/angular.js/blob/master/CHANGELOG.md). *GitHub*. Retrieved 2017-09-26.
17. "Github Release 1.7.9" (https://github.com/angular/angular.js/releases/tag/v1.7.9).
18. "Stable AngularJS and Long Term Support" (https://blog.angular.io/stable-angularjs-and-long-term-support-7e077635ee9c). *Angular Blog*. 2018-01-26. Retrieved 2018-03-16.
19. "Internet Explorer Compatibility" (https://docs.angularjs.org/guide/ie). *Angular JS 1.7.7 Developer Guide*. Google. Retrieved 12 Feb 2019. "AngularJS 1.3 has dropped support for IE8. Read more about it on our blog. AngularJS 1.2 will continue to support IE8, but the core team does not plan to spend time addressing issues specific to IE8 or earlier."
20. Minar, Igor. "AngularJS 1.3: a new release approaches" (http://angularjs.blogspot.com/2013/12/angularjs-13-new-release-approaches.html). *AngularJS Blog*. Retrieved 2014-10-12.
21. "Ok... let me explain: it's going to be Angular 4.0" (http://angularjs.blogspot.kr/2016/12/ok-let-me-explain-its-going-to-be.html). *angularjs.blogspot.kr*. Retrieved 2016-12-14.
22. https://blog.angular.io/version-5-0-0-of-angular-now-available-37e414935ced
23. "Angular 5 JavaScript framework delayed" (https://www.infoworld.com/article/3225511/javascript/angular-5-javascript-framework-delayed.html).

24. Fluin, Stephen (3 May 2018). "Version 6 of Angular Now Available – Angular Blog" (https://blog.angular.io/version-6-of-angular-now-available-cc56b0efa7a4). *Angular Blog*. Retrieved 8 June 2018.
25. "Angular versioning and releases" (https://angular.io/guide/releases#release-schedule). *angular.io*. Google. Retrieved 8 June 2018.
26. "Dynamic SSR & Static Pre-rendering" (https://medium.com/@MarkPieszak/angular-universal-server-side-rendering-deep-dive-dc442a6be7b7).
27. Kotaru, V. Keerti (2016-08-25). *Material Design Implementation with AngularJS: UI Component Framework* (https://books.google.co.uk/books?id=3gvpDAAAQBAJ). Apress. p. 4. ISBN 9781484221907.
28. "angular/angularjs-batarang (GitHub)" (https://github.com/angular/angularjs-batarang). Retrieved 2014-10-12.
29. Ford, Brian. "Introducing the AngularJS Batarang" (http://angularjs.blogspot.com/2012/07/introducing-angularjs-batarang.html). *AngularJS Blog*. Retrieved 2014-10-12.
30. "batarang Chrome extension for AngularJS appears broken" (https://stackoverflow.com/questions/23506526/batarang-chrome-extension-for-angularjs-appears-broken).

Further reading

- Green, Brad; Seshadri, Shyam (March 22, 2013). *AngularJS* (http://shop.oreilly.com/product/0636920028055.do) (1st ed.). O'Reilly Media. p. 150. ISBN 978-1449344856.
- Kozlowski, Pawel; Darwin, Peter Bacon (August 23, 2013). *Mastering Web Application Development with AngularJS* (https://www.packtpub.com/angularjs-web-application-development/book) (1st ed.). Packt Publishing. p. 372. ISBN 978-1782161820.
- Ruebbelke, Lukas (January 1, 2015). *AngularJS in Action* (1st ed.). Manning Publications. p. 325. ISBN 978-1617291333.

External links

- Official website (https://angularjs.org)