Angular

*ngThess (meetup1)

June 21, 2017

Nektarios Konstantinidis

www.nekkon.com

nekkon01@gmail.com

George Tzinos

geotzinos@gmail.com

Presentation topics

1. Introduction (+Latest news)
2. Typescript
3. IDE (Microsoft Visual Code)
4. Angular-cli
5. Testing Unit tests (Karma) / e2e tests (Protractor)
6. UI Components (Material2, PrimeNG, NG-Bootstrap, Kendo-UI, Admin Templates)
7. Flex-Layout
8. Routing (Lazy Loading)
9. Optimizations (Aot / Tree shaking)
10. Angular Universal (Server side rendering)
11. Nativescript / Ionic
12. Useful Resources
13. Future of Angular
14. Thank you!

1. Introduction

+ Latest news

1. Introduction

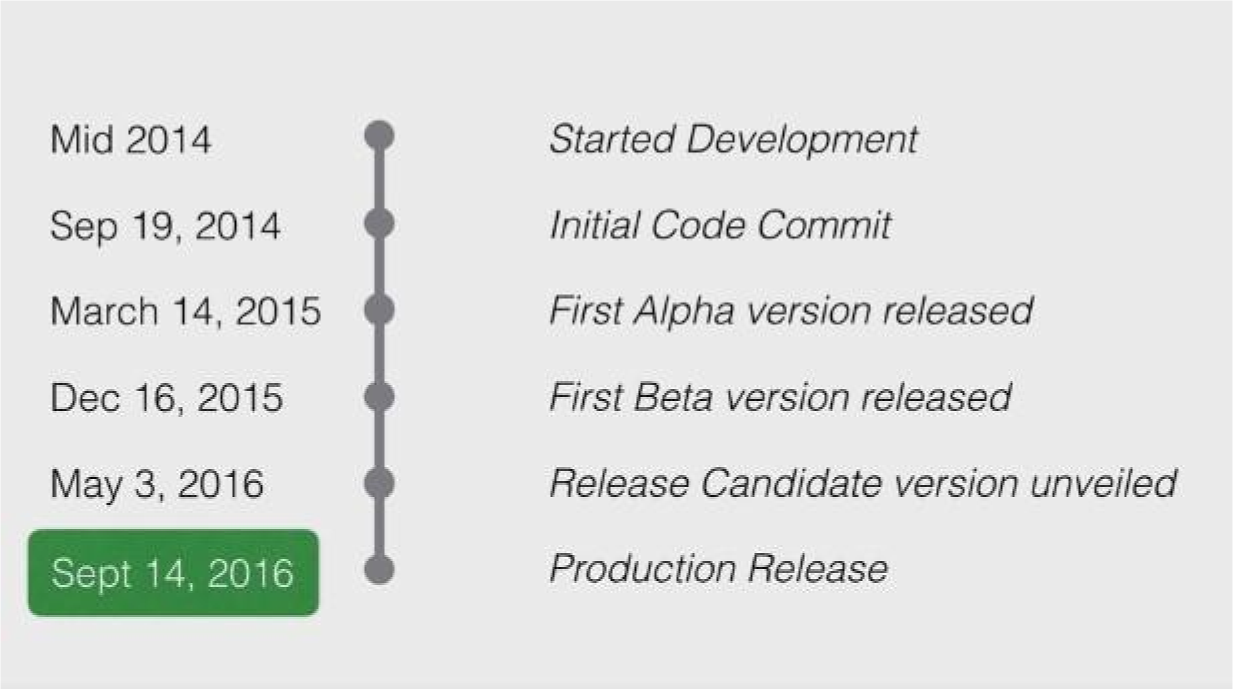
What is Angular?

Angular is a TypeScript-based open-source front-end web application platform led by the Angular Team at Google and by a community of individuals and corporations to address all of the parts of the developer's workflow while building complex web applications. Angular is a complete rewrite from the same team that built AngularJS.

AngularJS History

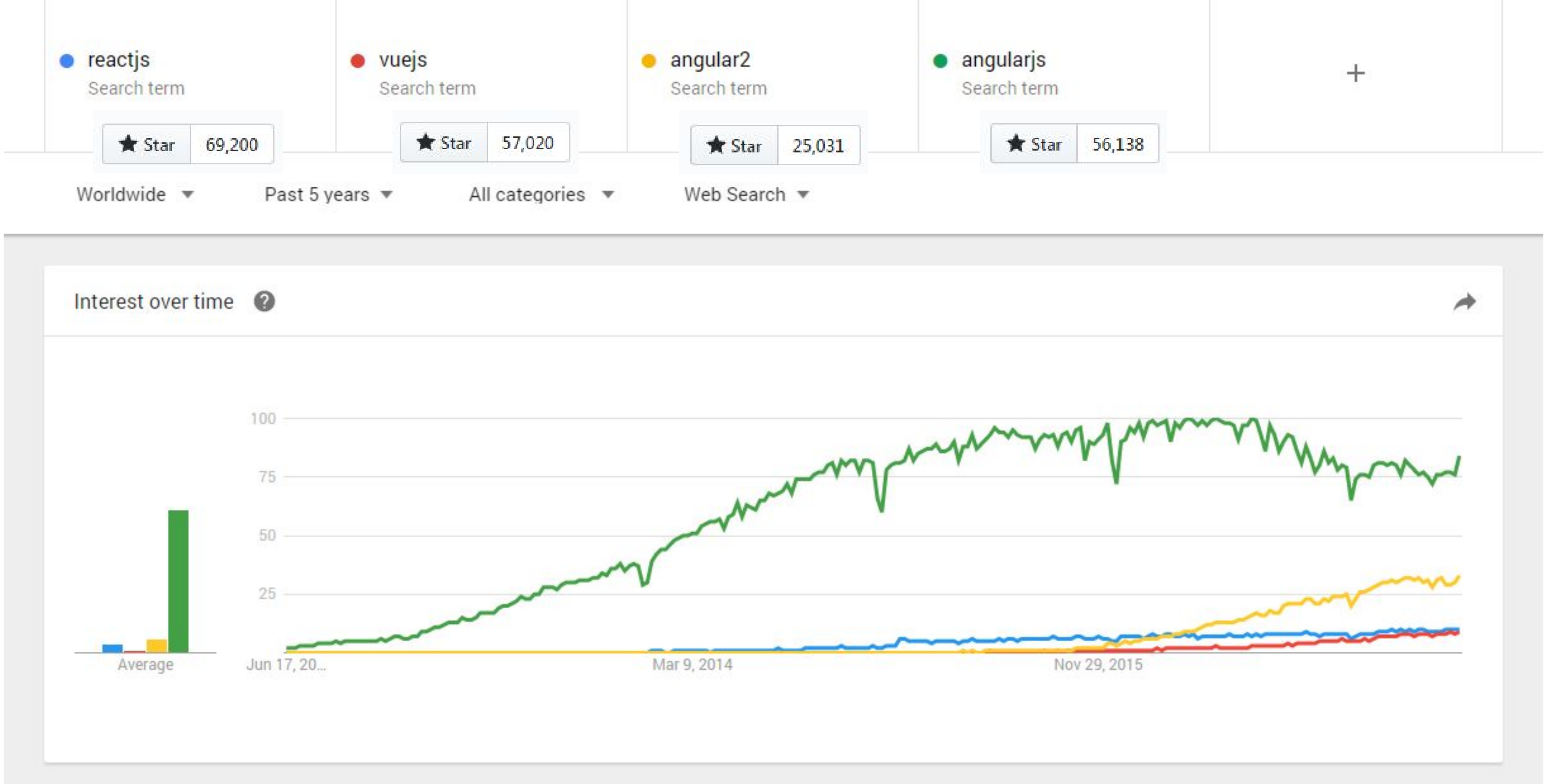
AngularJS version 1.0 was released in 2012.
Miško Hevery, a Google employee, started to work with AngularJS in 2009.
The idea turned out very well, and the project is now officially supported by Google.

1. Introduction - Angular2 Timeline



Mid 2014	*Started Development*
Sep 19, 2014	*Initial Code Commit*
March 14, 2015	*First Alpha version released*
Dec 16, 2015	*First Beta version released*
May 3, 2016	*Release Candidate version unveiled*
Sept 14, 2016	*Production Release*

1. Introduction - Popularity compared to other options



● reactjs
Search term
★ Star 69,200

● vuejs
Search term
★ Star 57,020

● angular2
Search term
★ Star 25,031

● angularjs
Search term
★ Star 56,138

Worldwide ▾ Past 5 years ▾ All categories ▾ Web Search ▾

Interest over time

1. Introduction - Sites made with AngularJS / Angular

1. The Guardian
2. Paypal
3. jetBlue
4. Lego
5. iStock Photo
6. Upwork
7. Netflix
8. Freelancer
9. Weather
10. Youtube

madewithANGULAR

www.madewithangular.com

http://angularexpo.com/

1. Latest news - Ng-conf 2017 (Salt Lake City Utah, April 5-7th)

- During the conference, Google mentioned that they have over 200 internal applications now running built on Angular.

- The conference focused on three main themes: Speed and Size, Smooth Updates, Simplicity

- Speed is critical to mobile applications and was the focus of several conference presentations. The Angular team shared some statistics from DoubleClick (a division of Google) that showed 53% of visits to mobile sites are abandoned if they take more than three seconds to load. 77 percent of current sites take more than ten seconds to load and the average load time is over 19 seconds on 3G connections. Angular aims to address this in two ways:
    - **Server side rendering**, or pre-rendering, allows all or a portion of an application to be generated on the server. The browser does not have to wait until all content has been downloaded and the Angular app has bootstrapped before content can be displayed. The Angular Universal project, which allowed Angular applications to be rendered server side, has now been adopted by the Angular team itself as part of the @angular/platform-server module.
    - Pre-rendering is not the only way to achieve an improvement in application load speed. One approach is to leverage PWA (**Progressive Web Application**) techniques using service workers as a means of loading and locally caching application content. By doing this, when a user makes a repeat visit to your site, the already pre-fetched content can be instantly loaded. This also permits the development of Angular applications that run offline, which is a necessity for some mobile applications. The mobile project that was first announced nearly a year ago has evolved into the @angular/service-worker module.

1. Latest news - Ng-conf 2017 (Salt Lake City, Utah)

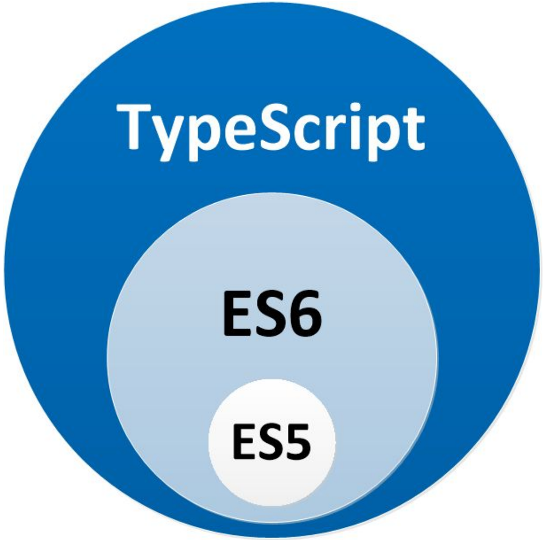
- Angular 4 was released about a week and a half before to the start of the conference. Unlike the jump from Angular 1 to Angular 2, this is a much more evolutionary release. The biggest enhancement is a reduction in application code size. In fact, AOT generated code from your custom components can be reduced by as much as 60 percent.

- The Angular team announced that Angular 4 would be an LTS (Long Term Support) release with support extended to October 2018. This is the first time any Javascript framework offers support to a stable release for as long as 18 months. This will appease corporate developers who cannot perform major version upgrades every six months. The transition from Angular 2 to Angular 4 has generally been trouble free and the Angular team has suggested that the upgrade path for Angular 5 in October 2017 should be similarly straightforward.

2. Typescript

2. Typescript

- Developed by Microsoft Team
- Superset of JavaScript (includes JavaScript)
- Compiles to JavaScript (ECMAScript 3, 5 or 6)
- Optional types! This allows to have autocomplete in IDE
- Easier refactoring of code
- Has features that are not yet in JavaScript (@Annotations, async/await)
- Compilation errors, instead of runtime errors in JavaScript (Static analysis of code)

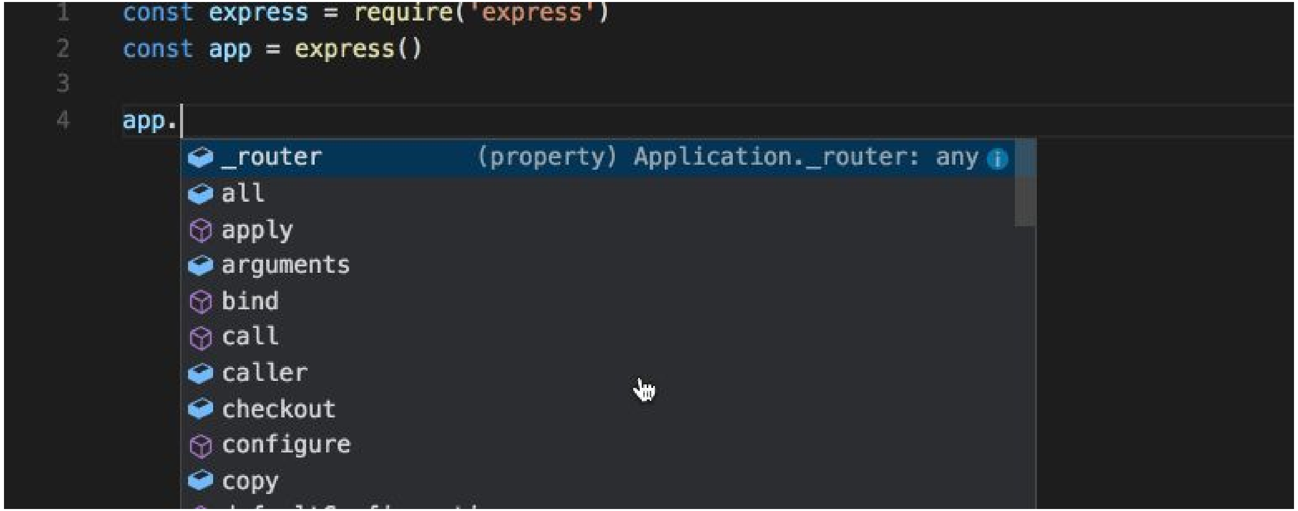


https://github.com/Microsoft/TypeScript/wiki/TypeScript-Design-Goals#goals

3. IDE (Microsoft Visual Code)

3. IDE - Microsoft Visual Studio Code

- Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux.
- It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Python, PHP, Go) and runtimes (such as .NET and Unity).



```
const express = require('express')
const app = express()

app.
```

3. IDE - Microsoft Visual Studio Code

- VS Code releases a new version each month with new features and important bug fixes.
- Based on Electron, a framework which is used to deploy Node.js applications.
- Debug code right from the editor. Launch or attach to your running apps and debug with breakpoints, call stacks, and an interactive console.
- Working with Git has never been easier. Review diffs, stage files, and make commits right from the editor. Push and pull from any hosted Git service.
- Extensible and customizable.

4. Angular-cli

4. Angular CLI

Initially, the entry barrier into the world of Angular development was pretty high because of the need to learn and manually configure multiple tools. Even to get started with a simple application, you need to know and use the TypeScript language, the TypeScript compiler, ES6 modules, SystemJS, npm, and a development web server. To work on a real-world project, you also need to learn how to test and bundle your app.

To jumpstart the development process, the Angular team created a tool called Angular CLI, which is a command-line tool that covers all the stages of the Angular application lifecycle, from scaffolding and generating an initial app to generating boilerplate for your components, modules, services, and so on. The generated code also includes pre-configured files for unit tests and bundling with Webpack.

Most common CLI commands: ng new, ng serve, ng generate, ng test, ng e2e, ng build

https://github.com/angular/angular-cli

5. Testing

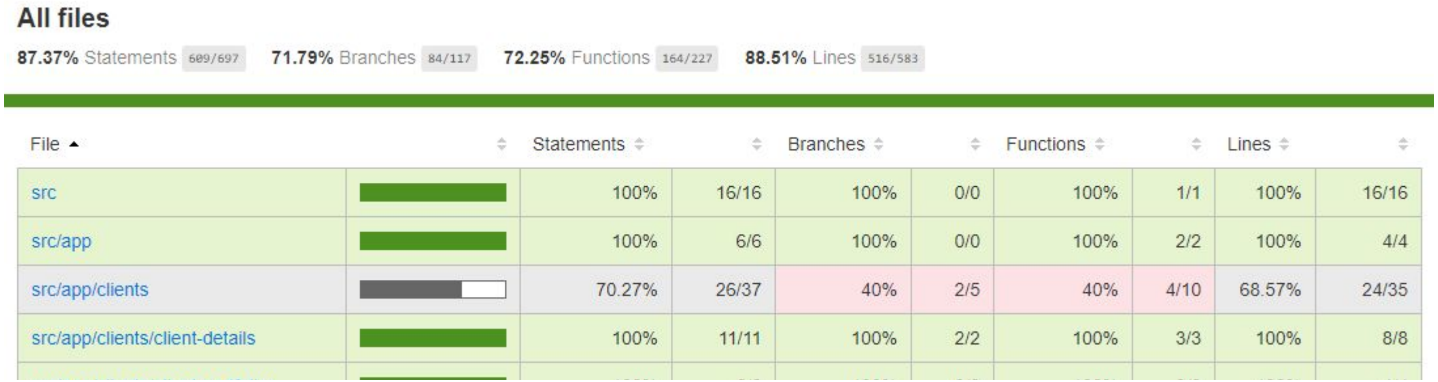
Unit tests (Karma) / e2e tests (Protractor)

5. Unit tests - Karma

Karma is a JavaScript test runner which means it will read code in a certain format (in our case written in Jasmine) and run it as a suite of tests.

https://github.com/onehungrymind/angular-applied-dashing

Angular-cli command: ng test -cc



All files

87.37% Statements 609/697 **71.79%** Branches 84/117 **72.25%** Functions 164/227 **88.51%** Lines 516/583

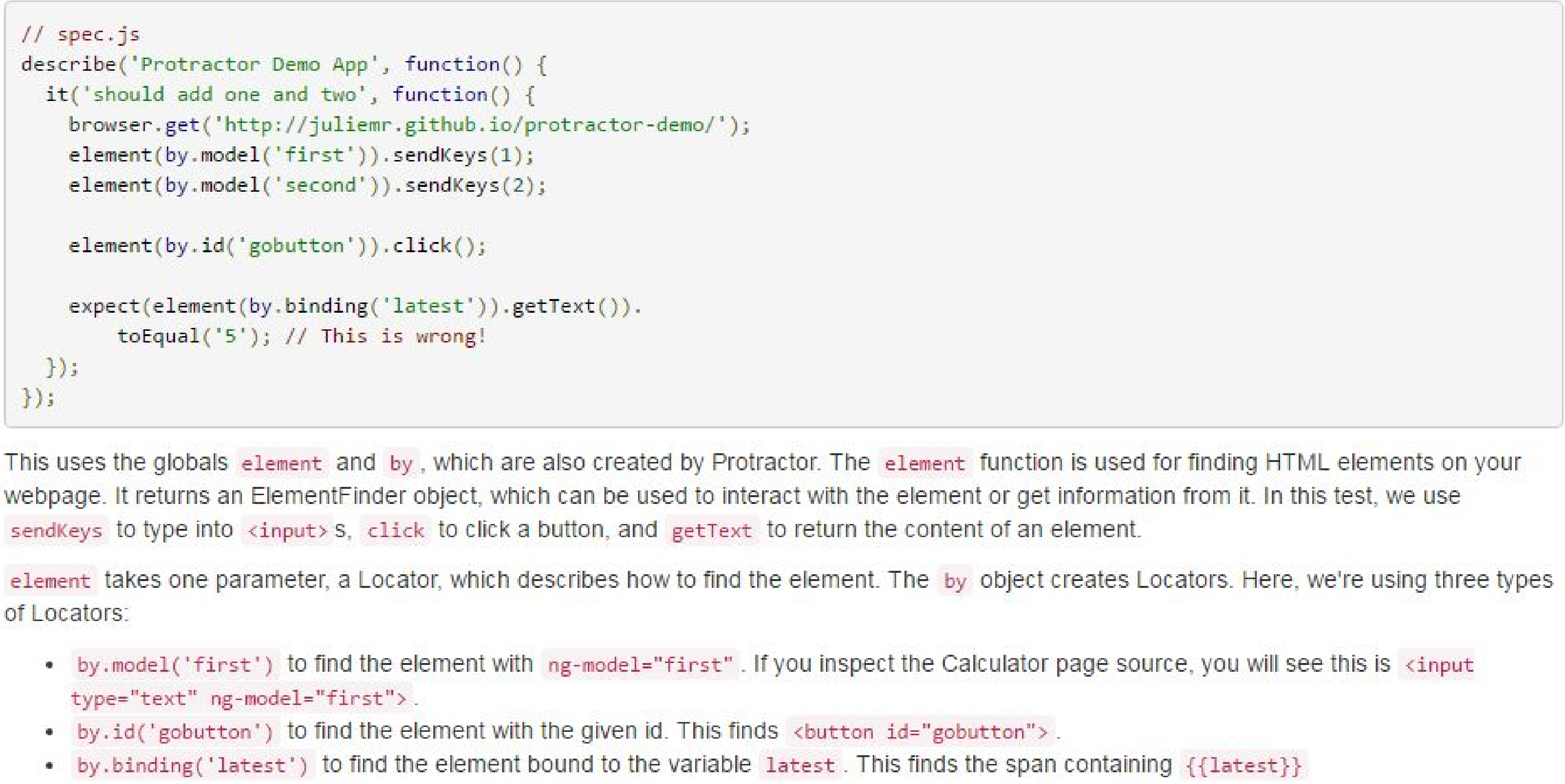
File ▲		Statements		Branches		Functions		Lines	
src		100%	16/16	100%	0/0	100%	1/1	100%	16/16
src/app		100%	6/6	100%	0/0	100%	2/2	100%	4/4
src/app/clients		70.27%	26/37	40%	2/5	40%	4/10	68.57%	24/35
src/app/clients/client-details		100%	11/11	100%	2/2	100%	3/3	100%	8/8

5. e2e tests (Protractor)

- Open source framework for end-to-end testing for Angular and AngularJS applications.
- Runs tests against your application running in a real browser, interacting with it as a user would.
- You no longer need to add waits and sleeps to your test.
- Run multiple tests automatically.
- By default, Protractor uses the Jasmine test framework for its testing interface.

http://www.protractortest.org/#/

5. e2e tests (Protractor)



```
// spec.js
describe('Protractor Demo App', function() {
  it('should add one and two', function() {
    browser.get('http://juliemr.github.io/protractor-demo/');
    element(by.model('first')).sendKeys(1);
    element(by.model('second')).sendKeys(2);

    element(by.id('gobutton')).click();

    expect(element(by.binding('latest')).getText()).
        toEqual('5'); // This is wrong!
  });
});
```

This uses the globals `element` and `by`, which are also created by Protractor. The `element` function is used for finding HTML elements on your webpage. It returns an ElementFinder object, which can be used to interact with the element or get information from it. In this test, we use `sendKeys` to type into `<input>`s, `click` to click a button, and `getText` to return the content of an element.

`element` takes one parameter, a Locator, which describes how to find the element. The `by` object creates Locators. Here, we're using three types of Locators:

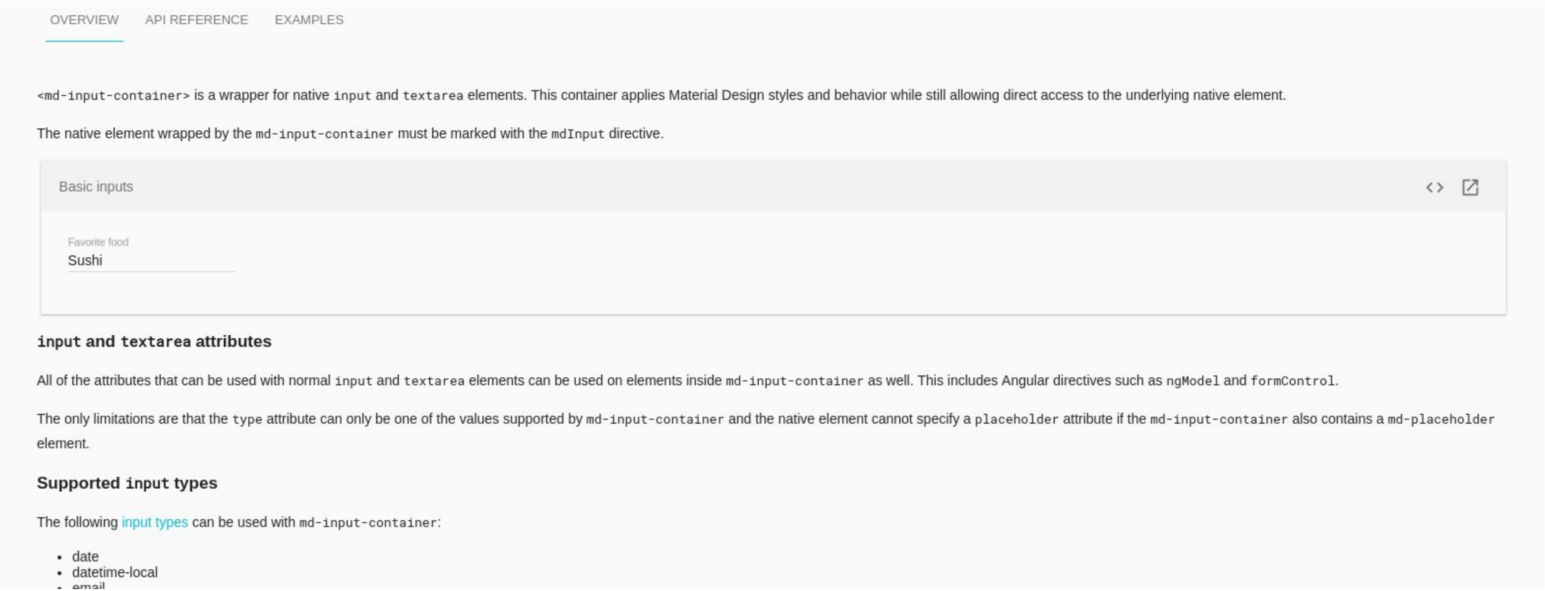
- `by.model('first')` to find the element with `ng-model="first"`. If you inspect the Calculator page source, you will see this is `<input type="text" ng-model="first">`.
- `by.id('gobutton')` to find the element with the given id. This finds `<button id="gobutton">`.
- `by.binding('latest')` to find the element bound to the variable `latest`. This finds the span containing `{{latest}}`

6. UI Components

Material2, PrimeNG, NG-Bootstrap, Kendo-UI

6. UI Components - Angular Material2

- Set of high-quality UI components built with Angular and TypeScript, following the Material Design spec.
- Modern UI.
- Fast and Consistent.
- Well-documented to serve as an example for Angular devs (Website).



`<md-input-container>` is a wrapper for native `input` and `textarea` elements. This container applies Material Design styles and behavior while still allowing direct access to the underlying native element.

The native element wrapped by the `md-input-container` must be marked with the `mdInput` directive.

Basic inputs

Favorite food

Sushi

`input` and `textarea` attributes

All of the attributes that can be used with normal `input` and `textarea` elements can be used on elements inside `md-input-container` as well. This includes Angular directives such as `ngModel` and `formControl`.

The only limitations are that the `type` attribute can only be one of the values supported by `md-input-container` and the native element cannot specify a `placeholder` attribute if the `md-input-container` also contains a `md-placeholder` element.

Supported `input` types

The following input types can be used with `md-input-container`:

- date
- datetime-local
- email

6. UI Components - PrimeNG

- Collection of rich UI components for Angular.
- All widgets are open source and free to use under MIT License.
- Developed by PrimeTek Informatics, a vendor with years of expertise in developing open source UI solutions.
- Doesn’t follow material design guidelines.
- Responsive & cross browser compatibility components.
- Multiple themes.
- Well documented with examples.

https://www.primefaces.org/primeng/#/

6. UI Components - NG-Bootstrap

- Built from scratch for angular.
- Based on bootstrap version 4.0.
- Supported browsers:
  - Chrome (45+)
  - Firefox (40+)
  - IE (10+)
  - Edge (20+)
  - Safari (7+)
- More on https://github.com/ng-bootstrap/ng-bootstrap
- Online Demo

6. UI Components - Kendo-UI

- Register to get a trial.
- More on http://www.telerik.com/kendo-angular-ui

6. Free Admin Templates

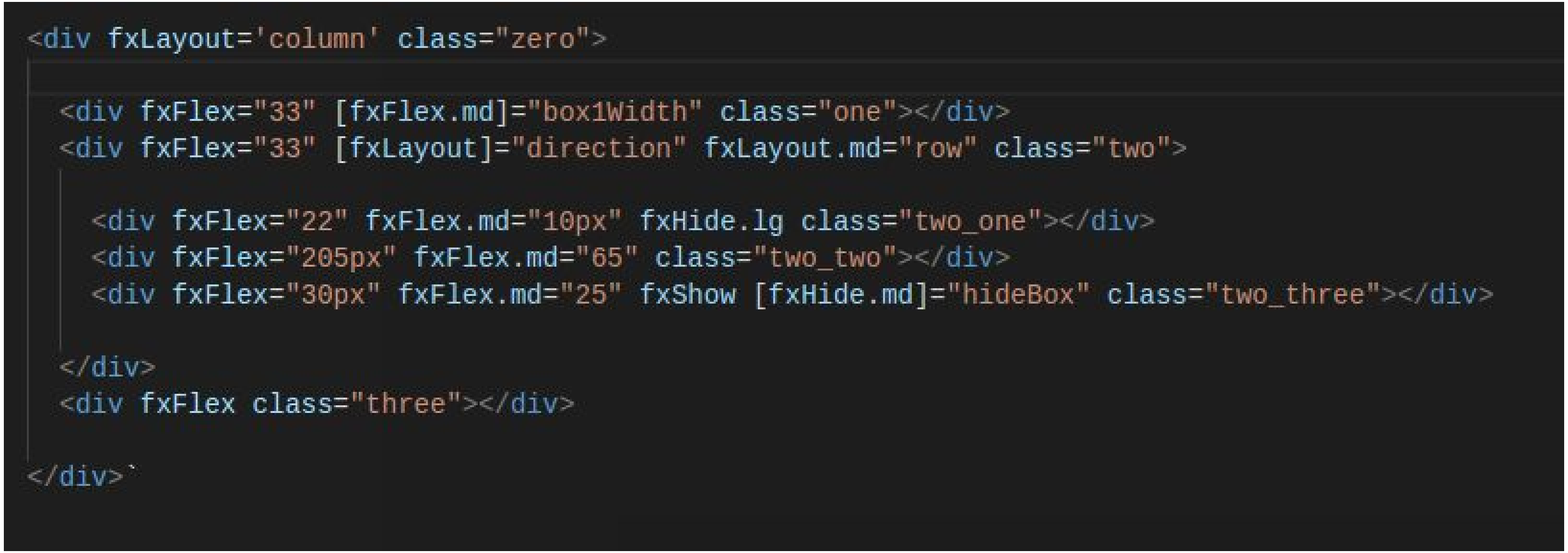
Free admin templates

- NG2-Admin (Source Code) (Demo)
- SB-Admin-BS4-Angular-4 (Source Code) (Demo)

7. Flex Layout

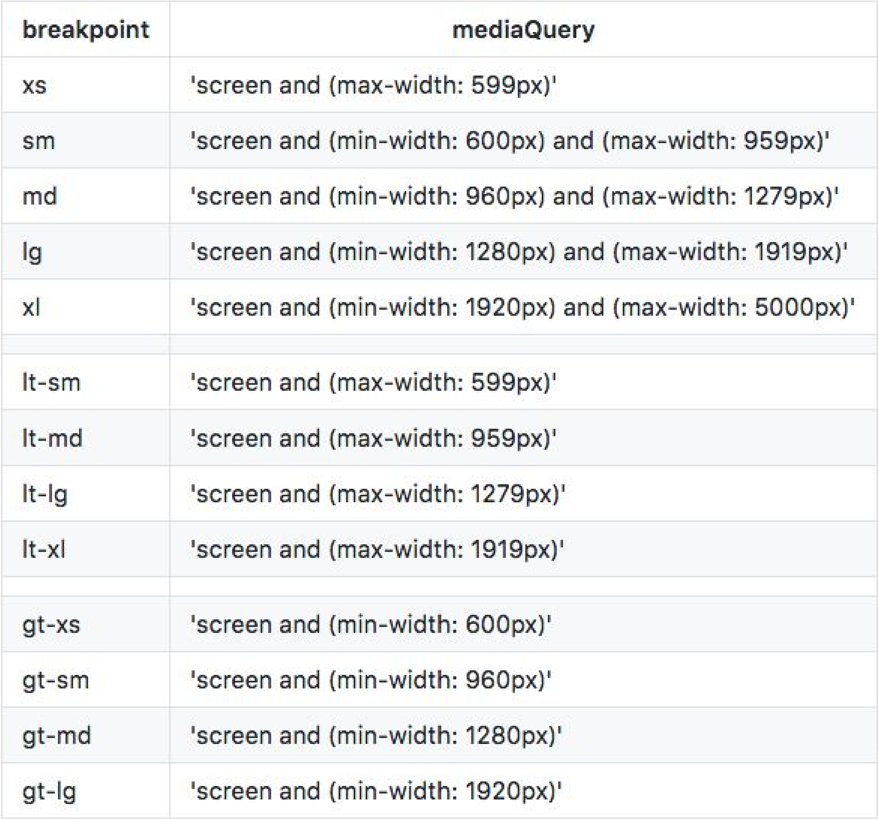
7. Flex-Layout

- Sophisticated layout API using Flexbox CSS + mediaQuery.
- For Angular (v4.1 and higher).
- Responsive & static layout.
- Breakpoints with mediaQuery definitions using breakpoint alias(es).



```
<div fxLayout='column' class="zero">

  <div fxFlex="33" [fxFlex.md]="box1Width" class="one"></div>
  <div fxFlex="33" [fxLayout]="direction" fxLayout.md="row" class="two">

    <div fxFlex="22" fxFlex.md="10px" fxHide.lg class="two_one"></div>
    <div fxFlex="205px" fxFlex.md="65" class="two_two"></div>
    <div fxFlex="30px" fxFlex.md="25" fxShow [fxHide.md]="hideBox" class="two_three"></div>

  </div>
  <div fxFlex class="three"></div>

</div>`
```

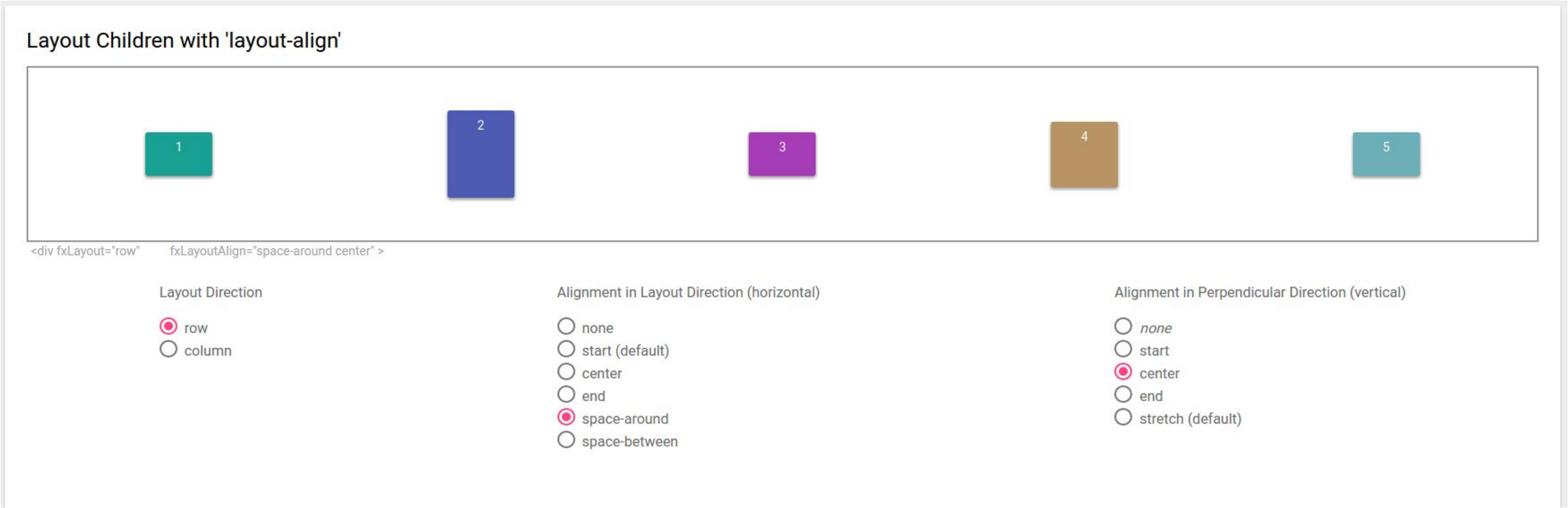
7. Flex-Layout



breakpoint	mediaQuery
xs	'screen and (max-width: 599px)'
sm	'screen and (min-width: 600px) and (max-width: 959px)'
md	'screen and (min-width: 960px) and (max-width: 1279px)'
lg	'screen and (min-width: 1280px) and (max-width: 1919px)'
xl	'screen and (min-width: 1920px) and (max-width: 5000px)'
lt-sm	'screen and (max-width: 599px)'
lt-md	'screen and (max-width: 959px)'
lt-lg	'screen and (max-width: 1279px)'
lt-xl	'screen and (max-width: 1919px)'
gt-xs	'screen and (min-width: 600px)'
gt-sm	'screen and (min-width: 960px)'
gt-md	'screen and (min-width: 1280px)'
gt-lg	'screen and (min-width: 1920px)'

7. Flex-Layout

Online demo
https://tburleson-layouts-demos.firebaseapp.com/#/docs



Layout Children with 'layout-align'

<div fxLayout="row" fxLayoutAlign="space-around center" >

Layout Direction

● row
○ column

Alignment in Layout Direction (horizontal)

○ none
○ start (default)
○ center
○ end
● space-around
○ space-between

Alignment in Perpendicular Direction (vertical)

○ *none*
○ start
● center
○ end
○ stretch (default)

8. Routing (Lazy loading)

8. Routing

https://angular.io/guide/router

Official Tutorial

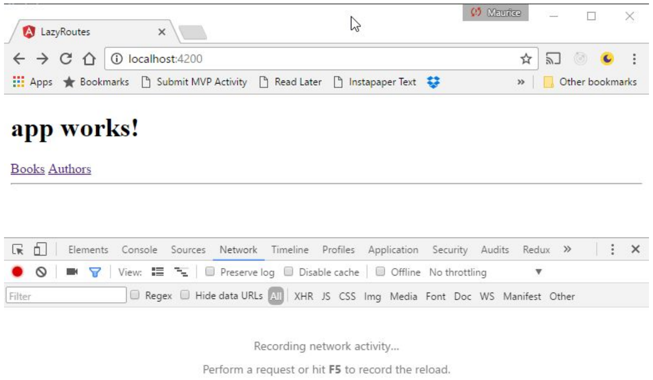
The tutorial describes development of a multi-page routed sample application. Along the way, it highlights design decisions and describes key features of the router such as:

- Organizing the application features into modules.
- Navigating to a component (Heroes link to "Heroes List").
- Including a route parameter (passing the Hero id while routing to the "Hero Detail").
- Child routes (the Crisis Center has its own routes).
- The CanActivate guard (checking route access).
- The CanActivateChild guard (checking child route access).
- The CanDeactivate guard (ask permission to discard unsaved changes).
- The Resolve guard (pre-fetching route data).
- Lazy loading feature modules.
- The CanLoad guard (check before loading feature module assets).

8. Routing - Lazy loading

One problem with creating a Single Page Application (SPA) is that you might load the entire application when the user first starts the application but most of it is never used. Sure navigation is nice and fast after the initial load but that slows down the initial load as that becomes larger. And that might be worth it if the user navigates to all, or most, routes loaded. But when most routes are never activated that is just a big waste.

Fortunately with Angular that is easy enough to fix with lazy loading. And lazy loading is really easy to set up in Angular as well.

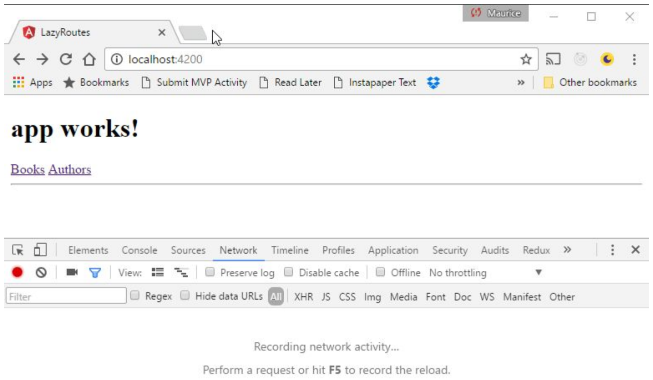


app works!

Books Authors

Recording network activity...

Perform a request or hit **F5** to record the reload.



app works!

Books Authors

Recording network activity...

Perform a request or hit **F5** to record the reload.

9. Optimizations

Aot / Tree shaking

9. Optimizations - AOT (Ahead of Time Compilation)

Why do AOT compilation?

- Faster rendering / Smaller Angular framework download size
  With AOT, the browser downloads a pre-compiled version of the application. The browser loads executable code so it can render the application immediately, without waiting to compile the app first.
  There's no need to download the Angular compiler if the app is already compiled. The compiler is roughly half of Angular itself, so omitting it dramatically reduces the application payload.
- Fewer asynchronous requests
  The compiler inlines external HTML templates and CSS style sheets within the application JavaScript, eliminating separate ajax requests for those source files.
- Detect template errors earlier
  The AOT compiler detects and reports template binding errors during the build step before users can see them.
- Better security
  AOT compiles HTML templates and components into JavaScript files long before they are served to the client. With no templates to read and no risky client-side HTML or JavaScript evaluation, there are fewer opportunities for injection attacks.

9. Optimizations - Tree shaking

AOT compilation sets the stage for further optimization through a process called tree shaking. A tree shaker walks the dependency graph, top to bottom, and shakes out unused code like dead leaves in a tree.

Tree shaking can greatly reduce the downloaded size of the application by removing unused portions of both source and library code. In fact, most of the reduction in small apps comes from removing unreferenced Angular features.

For example, this demo application doesn't use anything from the @angular/forms library. There is no reason to download forms-related Angular code and tree shaking ensures that you don't.

Tree shaking and AOT compilation are separate steps. Tree shaking can only target JavaScript code. AOT compilation converts more of the application to JavaScript, which in turn makes more of the application "tree shakable".

10. Angular Universal

Server-side Rendering

10. Angular Universal

The Angular Universal project consists of the base platform API and the surrounding tools that enables a developer to do server side rendering(or pre-rendering) of Angular applications. The platform API part has been merged into Angular core as of 4.0.

Why server side rendering?

- Prerender the application on the server
- Much faster time to first paint
- Enables better SEO
- Social media friendly.
- Enables content preview on social networks
- Fallback support for older browsers

https://github.com//cli-universal-demo

evertonrobertoauler

11. Ionic / NativeScript

Mobile Frameworks using Angular

11. Ionic

- Open source framework.
- Cross-platform:
  - Amazon Fire OS
  - Android
  - iOS
  - Windows 8 and 8.1
  - Windows Phone 7 and 8
  - Browser
- More than 120 device features.
- Well Documented.
- Uses webview.

https://ionicframework.com

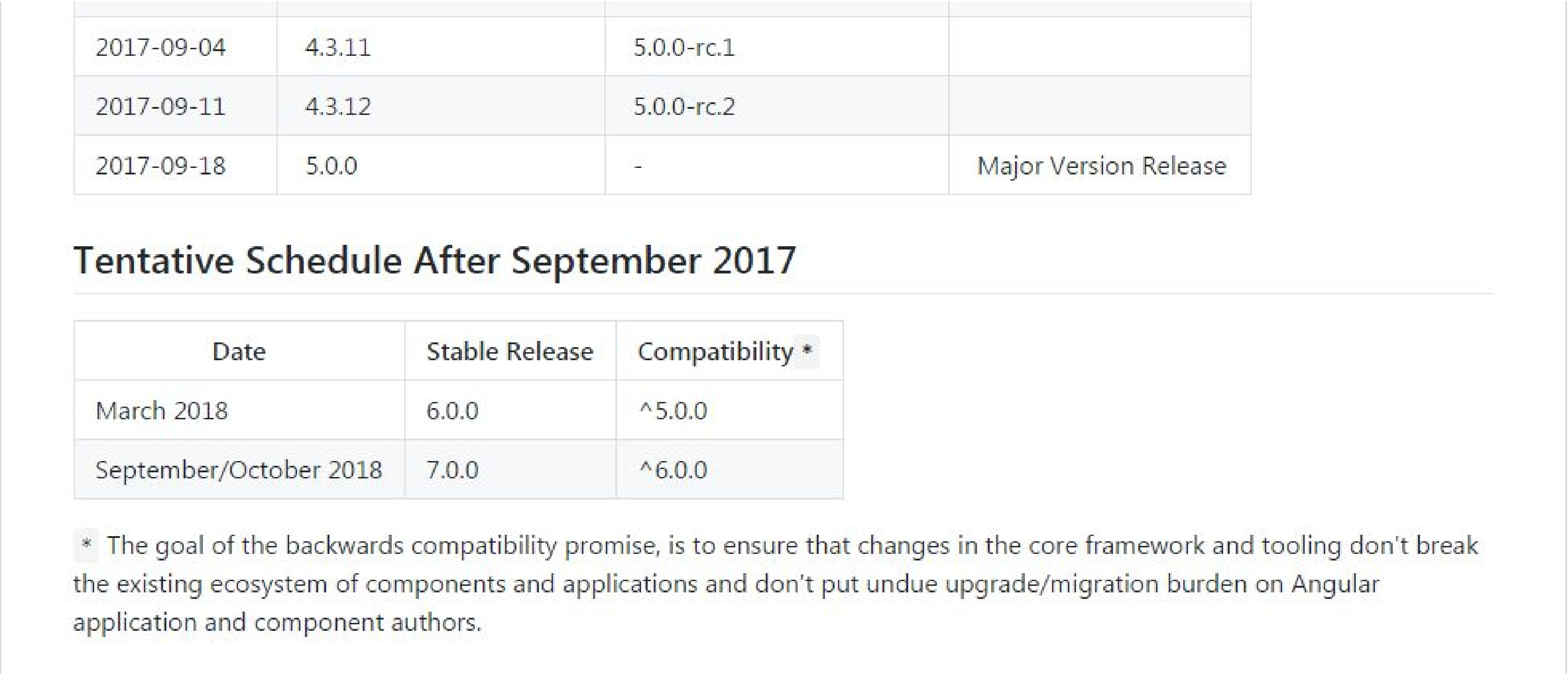
11. Nativescript

- Open source framework.
- Building truly native mobile apps (Without webview).
- Use of Angular, TypeScript or JavaScript.
- Build Android, iOS apps.
- Well documented with more examples.
- Official tutorials.
- Uses V8 on Android and JavaScriptCore on iOS.

https://www.nativescript.org/about

12. Future of Angular

12. Future of Angular



2017-09-04	4.3.11	5.0.0-rc.1	
2017-09-11	4.3.12	5.0.0-rc.2	
2017-09-18	5.0.0	-	Major Version Release

Tentative Schedule After September 2017

Date	Stable Release	Compatibility *
March 2018	6.0.0	^5.0.0
September/October 2018	7.0.0	^6.0.0

* The goal of the backwards compatibility promise, is to ensure that changes in the core framework and tooling don't break the existing ecosystem of components and applications and don't put undue upgrade/migration burden on Angular application and component authors.

https://github.com/angular/angular/blob/master/docs/RELEASE_SCHEDULE.md

13. Useful Resources

13. Useful resources

https://angular.io/ && https://angular.io/resources
Official Angular website

https://github.com/AngularClass/awesome-angular
A curated list of awesome Angular 2 and Angular 4 resources by @AngularClass

http://www.stefankrause.net/js-frameworks-benchmark6/webdriver-ts-results/table.html
Results for js web frameworks benchmark – round 6

https://github.com/angular-pakistan/ng-conf-2017/blob/master/README.md
Last angular conference

https://blog.angularjs.org/

Angular's official blog, great for latest news

14. Thank you!

There are polls available on meetup.com for you to vote:

1) Meetup topic
2) Meetup location
3) Meetup day/time

Feel free to contact us