

(GRADUALLY)

Upgrading From NG1 to NG2

From Babel to TypeScript

Sergio Cruz [@hashtagserg](#)

Developer & Instructor at Code School



Angular 1.x Application Getting Upgraded

What's the weather?

Los Angeles

United States

☀️ Get Weather

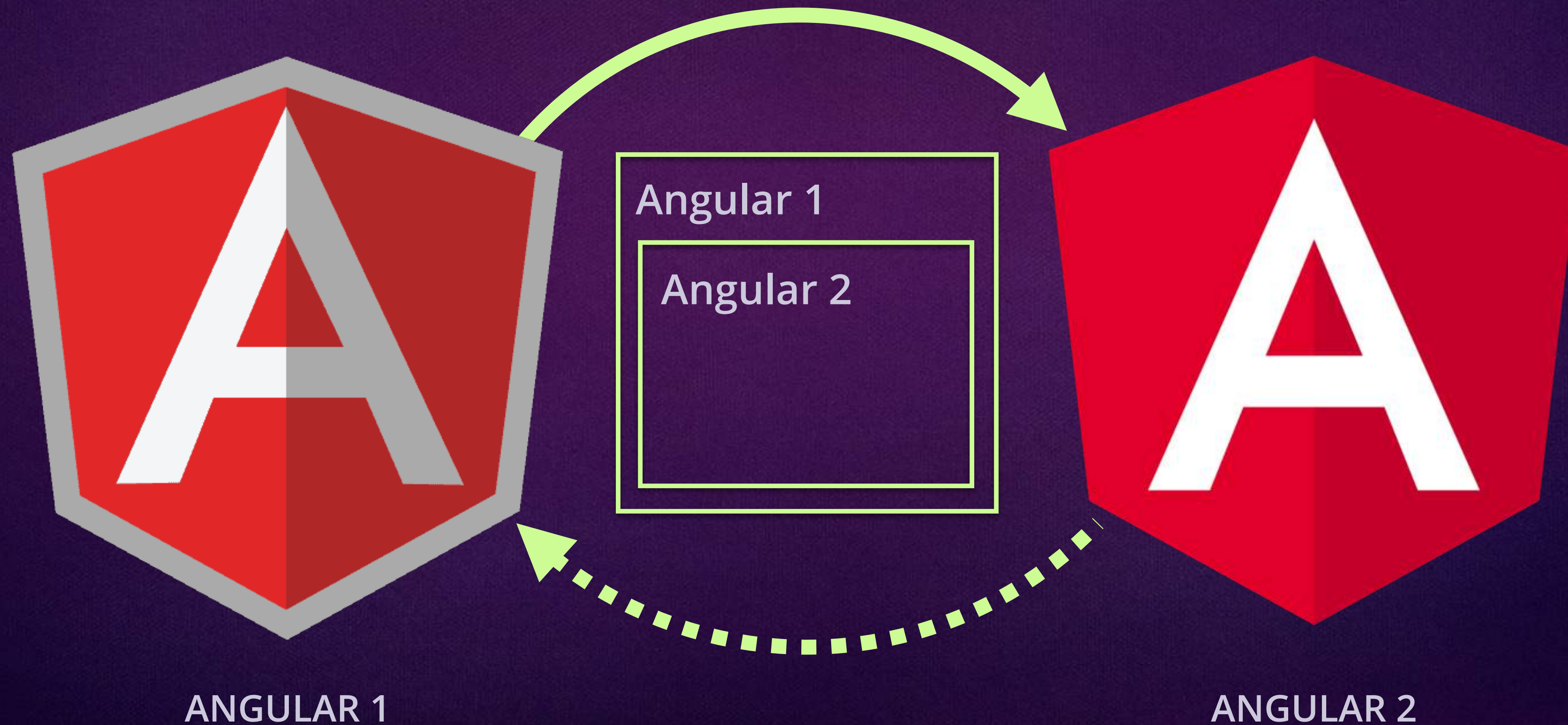


It's 69°F in *Los Angeles* right now.

directive getting upgraded

Running Both Versions of Angular

By the end of this talk our app will be running both versions of Angular



My Angular Story

How I've shipped Angular apps over the years

2013 - Manual Script Tags

Angular app inside a container with script tags on the page.

```
<script src="vendors/jquery.js"></script>  
<script src="vendors/angular.js"></script>  
<script src="vendors/angular-resource.js"></script>  
<script src="scripts/controllers/main.js"></script>  
<script src="scripts/controllers/about.js"></script>
```


2014 - The Year of Yeoman

Fully-fledged Single Page Application (SPA) with Angular generated with *generator-angular*.

```
<script src="scripts/vendor.js"></script>
<script src="scripts/scripts.js"></script>
```

```
ipt src="bower_components/
ipt src="bower_components/angular-sanitize/angu
ipt src="bower_components/angular-animate/angul
ipt src="bower_components/angular-touch/angular
ipt src="bower_components/angular-route/angular
endbower -->
endbuild -->
```

```
build:js({.tmp,app}) scripts/scripts.js -->
ipt src="scripts/app.js"></script>
ipt src="scripts/controllers/main.js"></script>
ipt src="scripts/controllers/about.js"></script>
```

Vendor and app files
were group and minified
in a build step.



2015 - Module Loaders such as Browserify

Node.js-like JavaScript code using NPM, CommonJS and Babel for ES2015 syntax support.

ES2015 Syntax



```
const angular = require('angular');
```

```
angular.module('app', [  
  require('angular-animate'),  
  require('angular-cookies'),  
  require('angular-resource'),  
  require('angular-sanitize'),  
  require('angular-ui-router'),  
])
```

require NPM modules

 No global variables

2016 - ?

Angular 2 had its first Release Candidate version, but now what?

TypeScript?

Migrate?

\\(ツ)//

SystemJS?

Rewrite?

Here's what I've been doing

I have been gradually upgrading my Angular 1.x Apps to Angular 2

IN BROWSERIFY, I AM TRANSPILING
WITH TYPESCRIPT INSTEAD OF BABEL

...BTW EASIER SAID THAN DONE

* SAME IS POSSIBLE WITH WEBPACK

How Sample App Was Built

Sample app uses similar architecture to what my larger apps look like.

Angular 1.5

NOT USING `.COMPONENT()`

UI Router

Gulp

Browserify

Babel

APPLICATION STRUCTURE

```
app/  
├── app.js  
└── weather  
    ├── index.js  
    ├── routes.js  
    ├── main.controller.js  
    ├── weather.controller.js  
    ├── weather.service.js  
    └── weather-preview.js
```


Demo #1

Steps for a Gradual Upgrade

How to get to the cutting edge while (trying to) playing safe

1. TRANSPILE WITH TYPESCRIPT
2. BRING ANGULAR 2 INTO CURRENT PROJECT
3. CONVERT DIRECTIVES TO ANGULAR 2 COMPONENTS

1. Transpiling with TypeScript

Replacing Babel with TypeScript

Steps to Transpile with TypeScript

What we need to do to move away from Babel

1 `npm install typescript --save-dev`

- a) created *tsconfig.json*
- b) renamed *app.js* to *app.ts*

2 `npm install typings --save-dev`

- a) installed all typings that were needed
- b) added *postinstall* script to install typings

3 `npm install tsify --save-dev`

- a) replaces *babelify* transform

4 update files

- a) fix imports to use `"* as"`
- b) use `require()` syntax for angular



Let's do these now
during live demo

Demo #2

2. Bring in Angular 2

Let's install Angular 2 and its dependencies

Packages Needed to Install Angular 2

No need to fear, but there are quite a few packages needed 😊💧

```
npm install --save-dev:
```

```
@angular/common  
@angular/compiler  
@angular/core  
@angular/platform-browser  
@angular/platform-browser-dynamic  
@angular/upgrade  
reflect-metadata  
rxjs  
zone.js
```

*USING VERSION 2.0.0-RC.5 FOR THIS APP

Remaining Steps

After installing dependencies we need to:

1 Get the UpgradeAdapter ready

- a) create new file *app/ng2-upgrade.ts*
- b) import dependencies
- c) export instance of *UpgradeAdapter*

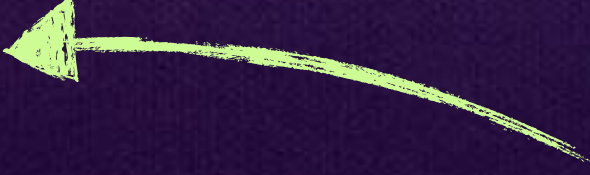
Important that app always uses
same instance of *UpgradeAdapter*



2 Manually bootstrap Angular 1.x

- a) remove *[ng-app]* from HTML
- b) add *angular.bootstrap(el, ['moduleName'])*
- c) import instance of *UpgradeAdapter*
- d) replace *angular.bootstrap* with upgrade adapter

Let's do these now
during live demo



Demo #3

3. First Angular 2 Component

Let's convert *<preview-weather>* into a Component

Converting Directive to an NG2 Component

Let's get the upgraded directive to render first:

1 Write *WeatherPreview* class

- a) rename to *weather-preview.ts*
- b) declare and export *WeatherPreview* class
- c) add *@Component* annotation with *selector & dummy template*

2 Downgrade Component

- a) in *weather/index.js*, import *UpgradeAdapter* instance at top of file
- b) in the directive declaration, wrap *WeatherPreview* with *upgradeAdapter.downgradeNg2Component()*

Live demo time
for all these



Demo #4.1

We Have Dummy Component, Now What?

Let's add some of the same features our previous directive had

Last live demo time



3 Accept `@Input()` from parent

- a) Import and add `@Input()` property to class
- b) Print out *weather* property with *json* pipe
- c) Update template to use `[]` binding syntax

4 Use previous directive template

- a) Update *ng-if* with **ngIf*, *class* with *[ngClass]*, etc
- b) Remove number pipe (doesn't exist in NG2)
- c) Don't use *ui-sref* as it is not supported

Demo #4.2

Pros and Cons of This Approach

Not an exhaustive list, but here are a couple of things that come to mind.

Pros

- Somewhat low risk
- No full rewrite needed
- Preserve a lot of current architecture

Cons

- Build process takes more time
- Bundle file size increases significantly

Next Steps

If you got your app to this point, what should you do next?

1. MIGRATE MORE DIRECTIVES
2. USE ANGULAR 1.X COMPONENT-BASED ARCHITECTURE TO EASE MIGRATION
3. MIGRATE THE REST OF YOUR APP (INCLUDING ROUTER, ETC)
4. USE A BETTER MODULE LOADER (BROWSERIFY IS A BIT DATED THESE DAYS)

Thank You 😊

<https://github.com/sergiocruz/upgrade-ng1-to-ng2>



Check out the "Accelerating Through Angular 2" course on [CodeSchool.com](https://www.codeschool.com)

Sergio Cruz [@hashtagserg](https://twitter.com/hashtagserg)

Developer & Instructor at Code School

