

# Angular 2

Building a basic JavaScript App

# What is Angular?

Angular is structural framework for developing dynamic web applications using HTML-like syntax and JavaScript.

# MVC – Model View Controller

To understand an Angular application, it helps to first understand the MVC pattern

- Model – the data used by the application
- View – how that data is presented
- Controller – manages changes to the model

# The Hello Angular2 App

Clone the hello-angular2 repo from GitHub:

<https://github.com/htc-ccis2591/hello-angular2>

The repo includes an index.html file and app folder. The app folder includes two JS files that make up our simple intro application.

# Component

- Components manage a part of the view
- They do this by applying a view template to a CSS style selector
  - A template is just HTML that tells Angular how to show or render a view
  - The selector is a CSS selector for the HTML tag where the view should be rendered

# Sample Component

```
(function(app) {  
  app.AppComponent = ng.core.Component({  
    selector: 'hello-app',  
    template: '<h1>Hello Angular 2</h1>'  
  }).Class({  
    constructor: function() {}  
  });  
})(window.app || (window.app = {}));
```

# IIFE - used for scope

```
(function(app) {  
  app.AppComponent = ng.core.Component({  
    selector: 'hello-app',  
    template: '<h1>Hello Angular 2</h1>'  
  }).Class({  
    constructor: function() {}  
  });  
})(window.app || (window.app = {}));
```

# Export Component

```
(function(app) {  
  app.AppComponent = ng.core.Component({  
    selector: 'hello-app',  
    template: '<h1>Hello Angular 2</h1>'  
  }).Class({  
    constructor: function() {}  
  });  
})(window.app || (window.app = {}));
```



# Component Definition

```
(function(app) {  
  app.AppComponent = ng.core.Component({  
    selector: 'hello-app',  
    template: '<h1>Hello Angular 2</h1>'  
  }).Class({  
    constructor: function() {}  
  });  
})(window.app || (window.app = {}));
```

# Create Class

```
(function(app) {  
  app.AppComponent = ng.core.Component({  
    selector: 'hello-app',  
    template: '<h1>Hello Angular 2</h1>'  
  }).Class({  
    constructor: function() {}  
  });  
})(window.app || (window.app = {}));
```

# Bootstrap the app

To have Angular load a Component, we need to put it in the global app object then use Angular's bootstrap method.

# Sample Bootstrapping

```
(function(app) {  
  document.addEventListener('DOMContentLoaded', function() {  
    ng.platform.browser.bootstrap(app.AppComponent);  
  });  
})(window.app || (window.app = {}));
```

# HTML file

In our index HTML file, there are important libraries to be included in the head section.

- Polyfills - for IE and angular
- Reactive Extensions RxJS (for Ajax)
- Angular 2

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Hello Angular 2</title>
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <link rel="stylesheet" href="styles.css">
7
8   <!-- IE required polyfill -->
9   <script src="https://cdnjs.cloudflare.com/ajax/libs/es6-shim/0.35.0/es6-shim.min.js"></script>
10  <script src="https://npmcdn.com/angular2@2.0.0-beta.13/es6/dev/src/testing/shims_for_IE.js"></script>
11
12  <!-- Angular 2 beta 13 -->
13  <script src="https://code.angularjs.org/2.0.0-beta.13/angular2-polyfills.js"></script>
14  <script src="https://code.angularjs.org/2.0.0-beta.13/Rx.umd.js"></script>
15  <script src="https://code.angularjs.org/2.0.0-beta.13/angular2-all.umd.dev.js"></script>
16
17  <!-- Our Angular2 app Modules (order is important) -->
18  <script src='app/app.component.js'></script>
19  <script src='app/main.js'></script>
20
21 </head>
22
23 <body>
24   <!-- my-app is the selector used by the Component -->
25   <hello-app>Loading...</hello-app>
26 </body>
27
28 </html>
```

# HTML file

# HTML selector

Angular locates the selector <hello-app>...

```
<body>  
  <!-- my-app is the selector used by the Component -->  
  <hello-app>Loading...</hello-app>  
</body>
```

# HTML selector (rendered)

And puts the template HTML inside it.

```
<body>
  <!-- my-app is the selector used by the Component -->
  <hello-app>
    <h1>Hello Angular 2!</h1>
  </hello-app>
</body>
```



# TypeScript

Angular 2 is promoting the use of TypeScript, which is a language that compiles to JavaScript.

TypeScript is typed superset of JavaScript and that typing helps develop larger and more maintainable apps using JavaScript.

# Hello App w/ TypeScript

Clone the hello-angular2-typescript repo:

<https://github.com/htc-ccis2591/hello-angular2-typescript>

The repo includes an index.html file and app folder. The app folder includes two TS files that make up our simple intro application.

# Sample Component in TS

```
import {Component} from 'angular2/core';
```

```
@Component({  
  selector: 'hello-app',  
  template: '<h1>Hello Angular 2 & TypeScript!</h1>'  
})  
export class AppComponent { }
```

# Bootstrapping in TS

```
import {bootstrap}    from 'angular2/platform/browser';  
import {AppComponent} from './app.component';  
  
bootstrap(AppComponent);
```

# Updated HTML file

In our index HTML file, we need to include some additional pieces.

- Polyfills - for IE and angular
- Reactive Extensions RxJS (for Ajax)
- Angular 2
- System JS with config (handles transpiling to JS & imports)

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Angular 2 QuickStart</title>
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <link rel="stylesheet" href="styles.css">
7
8      <!-- IE required polyfills, in this exact order -->
9      <script src="https://cdnjs.cloudflare.com/ajax/libs/es6-shim/0.35.0/es6-shim.min.js"></script>
10     <script src="https://cdnjs.cloudflare.com/ajax/libs/systemjs/0.19.20/system-polyfills.js"></script>
11     <script src="https://npmcdn.com/angular2@2.0.0-beta.13/es6/dev/src/testing/shims_for_IE.js"></script>
12
13     <script src="https://code.angularjs.org/2.0.0-beta.13/angular2-polyfills.js"></script>
14     <script src="https://code.angularjs.org/tools/system.js"></script>
15     <script src="https://npmcdn.com/typescript@1.8.9/lib/typescript.js"></script>
16     <script src="https://code.angularjs.org/2.0.0-beta.13/Rx.js"></script>
17     <script src="https://code.angularjs.org/2.0.0-beta.13/angular2.dev.js"></script>
18
19     <!-- Configure SystemJS -->
20     <script>
21         System.config({
22             transpiler: 'typescript',
23             typescriptOptions: { emitDecoratorMetadata: true },
24             packages: {'app': {defaultExtension: 'ts'}}
25         });
26         System.import('app/main')
27             .then(null, console.error.bind(console));
28     </script>
29 </head>
30 <body>
31     <!-- hello-app is the selector used by the Component -->
32     <hello-app>Loading...</hello-app>
33 </body>
34
35 </html>

```

# HTML file

# Learning Angular 2

Currently almost all examples (including from the creators) are done using TypeScript.

Eventually, we will see the JS examples as well, but perhaps now that you've seen how simple TypeScript can be you'll want to stick with it.