The Future of Traffic Portal
Steve Hamrick
Introduction

Steve Hamrick
Software Engineer @ Comcast
CDN Team
shamrick@apache.org
About

→ Current state of Traffic Portal
→ High level overview
  → Whats new
  → Tech stack
→ Comparisons

Agenda

→ Traffic Portal v1
→ Angular History
→ The New Angular
  → Comparison
  → Overview
  → Out of the Box
→ Demo
→ Q&A
Traffic Portal v1

→ AngularJS
  → End of Life 12/2021
→ JQuery
→ Underscore
→ Flot
→ SCSS

→ Custom build system
  → NPM
  → Bower*
  → Grunt
    → Compass
    → Browserify
    → HTML2js
    → Cache Busting
    → Etc.
AngularJS History

→ Started at Google
→ Single Page Application
→ Open Source
→ JavaScript Framework
→ Fell behind other web frameworks
  → Performance
  → Tooling
Angular - High Level

→ No longer follows model-view-controller design
  → Uses structural or attributive directives
  → $scope and controllers are gone!
→ Now a proper framework
  → Includes CLI
  → Build system
→ Accessibility support
→ Mobile/Legacy browser support
→ Modules
  → Dynamic Loading
  → Async template compilation
  → Better callbacks
  → 6-Month release cadence
  → New binding & expression syntax
  → Typescript
Angular – Out of the Box

→ Semantic Versioning
→ Typescript
→ SCSS
→ Build System
→ Angular CLI
→ Unit/E2E Testing
→ And much more

Source: angular.io/presskit
Angular - Semantic Versioning
Typescript

→ Open source
→ Superset of JS
→ Transpiles to JS
→ Type "safety"
→ Linter

Source: wikipedia
Angular Build System

→ Introduces the Angular Compiler
  → Complements Angular Runtime
→ Internationalization
→ Polyfills
→ Bundler
→ Mobile browser support
→ Transpilation/Compilation (SCSS -> CSS, TS -> JS)
Angular CLI

→ Encompasses entire ecosystem
→ Generation
  → Modules, components, services, etc.
→ Updates
→ Schematics

Source: angular.io/presskit
Angular Tooling

- UI
  - Material
  - Layout (flex, grid)
  - Forms
  - Animations
- Component Development Kit (CDK)
- Schematics

Source: angular.io/presskit
Testing

- Every generated file includes Unit Tests
  - Karma
- Project creation also sets up E2E tests
  - Protractor
- Able to change frameworks using schematics

Source: angular.io/presskit
Angular Compiler

→ Client browser has no knowledge of this step
→ Converts templates into imperative code
  → Was declarative
→ Automatic optimizations
→ JIT/AOT compilation
  → Runtime does JIT (if necessary)
  → Compiler does AOT
Server-side Rendering

→ Improves start up performance
  → Leverages the new compiler/runtime
→ Server runs some Angular Runtime steps
  → User gets semi-processed script

Source: angular.io/presskit
Binding
Component
Theming
Modules/Lazy Loading
Routing
Guards
Interceptors
Questions?