Building Apps with Angular 11

Speaker: Sonu Sathyadas

Practice Head (Open-source and .NET)

Agenda

- History: Angular JS and Angular
- Angular framework features
- Single Page Application Vs Server-side web apps
- Project architecture and setup
- Angular Style Guide
- More than components and directives:
  - Guards, Interceptors, Resolver, i18n, Angular material
  - Asyc actions with Observables and Promises
  - Connecting to REST apis
- Migration support from AngularJS to Angular
- Angular vs other SPA frameworks/libraries

History: Angular JS and Angular

Angular JS 1.x

- Initial release on Oct 2010
- Latest release 1.8.2
- Developed by Google
- Open-source, SPA library
- Coding language: JavaScript
- Features:
  - Two-way binding and forms
  - Built-in Dependency Injection, routing
  - Lightweight
  - Large community support
- Patterns: MVC /MVVM

Angular 2+

- Initial release Sep 2016
- Versions: 2, 4, 5,6, 7,8,9, 10, 11(current)
- Developed by Google
- Open-source Web Framework for SPA
- Coding language: TS (current), DART, JS
- Easy to learn and implement
- Pattern: Component based architecture

Angular framework features

- CLI support
- Component based
- Two-way binding and form validation
- Declarative routing configurations
- Modules and lazy loading
- Built-in Dependency Injection
- Async patterns with RxJS observables and promises
- Easy to connect with backend REST services
- Internationalization support

Single Page Application Vs Server-side web apps

Single Paged Application

- SPA loads the application modules to browser on the initial request.
- Page/View changes happens with in browser, not from server.
- Views are rendered on browser
- Improved user experience (offline browsing)
- Can load additional modules on demand.
- Connect to backend APIs for dynamic data using XHR (Ajax)
- JS is used as programming language.

- Server-side web apps (Multi page applications)
  - Loads only a single page on initial request.
  - Makes round trip to server for loading views on each request.
  - Pages will be rendered on server side.
  - Can be developed using any framework like ASP.NET, Java, PHP etc

Project architecture and setup

- Install Angular CLI

  npm install -g @angular/cli
- Create project

  ng new <project-name> [options]

  eg: ng new sample-project --minimal
- Run project

  ng serve [options]

  ng serve -o --port 5600
- Build the project

  ng build [options]

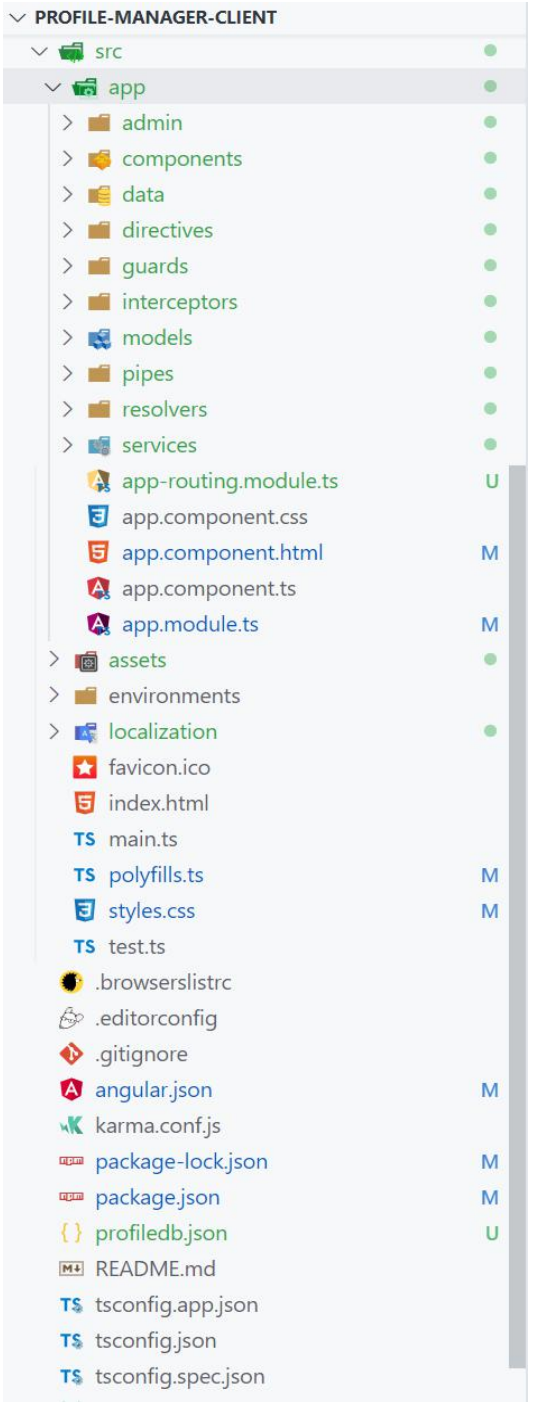
  ng build --prod
- Create components, directives, pipes, services, modules and services

  ng generate component <path/name> [options]

  ng generate component components/home --module app.module

Project architecture and setup

- package.json
- angular.json
- tsconfig.json
- main.ts
- AppModule
  - app.module.ts
- AppComponent
  - app.component.ts
  - app.component.html
  - app.component.css
- index.html



Angular Style Guide

- Guide to Angular syntax, conventions, and application structure

  https://angular.io/guide/styleguide

- Guidelines:
  - Style 01- Single responsibility
  - Style 02- Naming
  - Style 03- Application structure and NgModules
  - Style 04- Components
  - Style 05- Directives
  - Style 06- Services
  - Style 07- Data Services
  - Style 08- Life cycle hooks

More than components and directives

- Route Guards - CanActivate, CanDeactivate
- Interceptors - Http request interceptor services
- Resolver -Preload data for components
- i18n - Support multiple locale (localization)
- Angular material - A theming module for Angular
- Asyc actions with Observables and Promises
  - Handling stream of events
  - built-in promises
- Connecting to REST apis
  - HttpClientModule and HttpClient service

Migration support from AngularJS to Angular

AngularJS <=1.4	**AngularJS 1.5+**	**Angular 2+**
Controllers	Components	Components
$scope		
html templates		
Directives	Directives	
Filters	Filters	Pipes
Services	Services	Services (@Injectable)
Built-in services	Built-in services	Built-in services
$q	$q	Promise/Observables
$http,	$http,	HttpClient (HttpClientModule)
$state	$state	Router
$stateParams	$stateParams	ActivatedRoute
index.html	index.html	index.html
Module	Module	Module
Additional migration requirements		
JavaScript	JavaScript	TypeScript
No built-in bundling	No built-in bundling	Webpack
client side package manager	client side package manager	NPM package manager

Angular vs other SPA frameworks/libraries

- React:
  - A library for developing data driven views for web apps
  - Developed by facebook
  - Virtual DOM for better view rendering
  - JSX for app components
  - Supports only one-way data binding by default
  - Build native mobile apps using React Native
- Vue.JS
  - Progressive JavaScript framework
  - Uses VirtualDOM
  - small size
  - Support both JS and TS

Thank You