Angular as a Platform and a Closer Look at the CLI

A Closer Look at "ng new"

- Creates a new **project workspace**

IDE and Project Setup

- Nothing to note

Understanding the Config Files

- **.editorconfig** → Makes it so all users have the **same code** regardless of whatever editor they use
- **.gitignore** → Lists what files/directories to **ignore**
- **angular.json** → **Manges** the entire **project** configuration
- **browserslist** → Tells the CLI what **browsers** you wish to **support**
- **karma.konf.js** → Used for **unit testing**
- **package-lock.json** → Managed by **package.json**
- **package.json** → **Manages** the **packages** used by the project
- **tsconfig.json** → Manages the **TypeScript compiler**
- **tsconfig-app.json** → Pertains to **application compilation**

Important CLI Commands

- **ng help** → **List** all **commands**
- **ng serve** → **Serves** the application
- **ng generate** → Lets up **generate things** in the application (components, services, etc)
- **ng lint** → **Alerts** you of any **linting errors**
- **ng build** → **Builds** the project

The "angular.json" File - A Closer Look

- **Automatically generated** by the CLI
- Used to **run CLI commands**

- So much **junk**

Angular Schematics - An Introduction

- **Blueprints** that Angular commands can pick up

The "ng add" Command

- **Adds** Angular-related **libraries** and **packages**
- Also capable of **updating** existing **files**

Using Custom "ng generate" Schematics

- I **don't care** anymore
- Just **end** this

Smooth Updating of Projects with "ng update"

- Used to **update** to the **current stable release** of the **core framework** and **CLI**

Simplified Deployment with "ng deploy"

- **Built-in** loaders
  - **ng build / ng test / ng lint**
  - All **execute**, **compile**, and **analyze** code
- **ng deploy**
  - Both builds and deploys the application

Understanding "Differential Loading"

- Our Angular **app** should be **hosted** off a **hosting provider**
  - **Modern browsers** need no/less polyfills, with smaller required bundles
  - **Legacy browsers** need all/more polyfills, with larger required bundles
- **Differential loading** allows for both routes, making it so that browsers get the **exact amount of code** they need to run → Fights **redundant code**

Managing Multiple Projects in One Folder

- Wasn't even paying attention

- You gotta love **deprecation**

Angular Libraries - An Introduction

- We can **generate** a **library** with **ng generate library <library-name>**
- **Libraries** are meant to hold **code shared** amongst **numerous applications**