Introduction

- Allen Conway

  - **Blog**: http://www.AllenConway.net
  - **Twitter**: @AllenConway, http://www.twitter.com/AllenConway
  - **GitHub**: https://github.com/AllenConway
  - **Email**: AllenC@Magenic.com

Agenda

- AngularJS and Angular
- TypeScript
- Decisions, decisions, decisions
- Project setup
  - project.json
  - webpack
- Hybrid bootstrap via Angular Upgrade module
- Component conversion

AngularJS and Angular

- AngularJS = Angular 1.x
  - Complete client side framework for creating JS Apps
  - Based on JavaScript
  - Dynamic content via two-way data-binding
  - Leverages $scope and controllers
- Angular = Angular 2.x +
  - Complete rewrite of AngularJS
  - Written in TypeScript
  - Improved performance
  - Leverages component based UI

TypeScript

- High level language like JavaScript yet offering familiar OO concepts and techniques
- Superset of JavaScript
- All valid JavaScript is valid TypeScript
  - Transpiled to JavaScript
  - Microsoft chose to build atop JavaScript
  - Big wins for TypeScript in Angular 2.0 adoption
- Offers compile-time checking for type safety
- Provides features ahead of ECMAScript Standards
- Pick your favorite IDE
- Intellisense!

Architectural Decisions

- Rewrite vs Hybrid
  - Size of the codebase
  - TypeScript use
  - Version of AngularJS
  - General quality of original codebase
  - Time allotment
    - Migrate over time
    - Bang Bang approach
  - Code needing refactoring
  - Longevity of the codebase
  - Front-end churn
- Hybrid Estimates

Angular Migration Assistant

- Quick check to assess ng1 application
- Install globally
  - **npm install ngma -g**
- Run against project directory
  - **ngma <directory>**
- Additional input/feedback

Migration AngularJS -> Angular

- Angular Upgrade Module
  - Allows AngularJS and Angular to coexist in a single application
- Two independent frameworks running
  - Each framework treats the other as a black box
  - Each DOM element owned by respective framework
    - AngularJS directives execute in AngularJS
    - Angular components execute in Angular
- Preparation in AngularJS
  - Component Directives
  - Migrate factories to services
  - Migrate to TypeScript

Project Environment Choices

- Visual Studio Code
  - Static Files Only
- Visual Studio
  - .NET + Angular Web
  - ASP.NET Core + Angular Web
- WebStorm
  - Static Files Only
- Sublime Text, Atom Editor…
- Windows, Mac, Linux
- Make considerations if you must work with server-side code too

Configure package.json

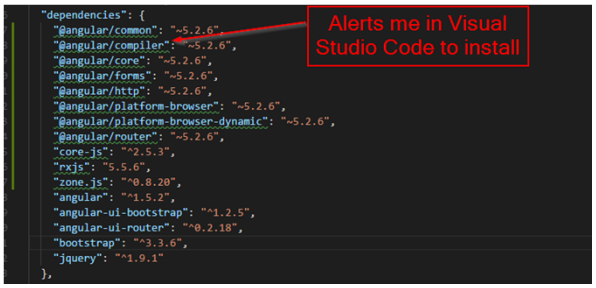
- if (package.json){ let result = “move to next step” }
- Add package.json to the application
  - Begin updating dependencies in index.html and remove from nuget
    - Replace with npm packages
- Add in node types
  - typings and tsd are deprecated
- Decide on build process and server
  - Prefer to unhook from Visual Studio .NET builds for front-end code
  - Lite-server, webpack dev server
- Configure npm scripts for running

Add a Module Loader / Bundler

- SystemJS
  - Module loader enableing dynmaic ES module workflows in browsers and NodeJS
  - `npm install systemjs`
- Webpack
  - Module bundler; emits 1…n bundles based on configuration
  - `npm install webpack`
  - Add dev dependencies for webpack loaders
  - Create configuration
    - Don't start with a blank slate
    - Start with an auto-generated configuration
    - Use a template from Angular ToH or similar

Scaffold Angular

- Add @angular and minimum required dependencies from npm
  - Be cautious with the versions of TS and Angular
- Add in the entry point of the application
  - main.ts
  - app.module.ts
- Remove <ng-app>



Hybrid Application Checkpoint 1

- STOP, GOTO line 200
  - Don't proceed with any further upgrades
  - Ensure the app runs as a hybrid Angular + AngularJS application
- Run the application in the browser
  - Ensure there aren't any console errors
  - Ensure all files are still loaded and application functions correctly
- Common issues
  - Module/loader bundler configuration
  - Name of the ng1 module not matching
  - Incorrectly configured entry point code
  - Incorrect references

Migrate to TypeScript

- TypeScript
  - Current version 2.7.2
- `npm install –g typescript`
- Add a tsconfig.json file to project
  - `tsc --init`
  - If unsure of settings, copy from Angular template as base
- Convert files to .ts
  - New files always use .ts if still in AngularJS
- Incorporate build process for TypeScript compilation

Nrwl Extensions

- Open Source extensions for ng
  - Referred to as 'Nx'
- AngularJS Upgrade Module
  - CLI commands to generate hybrid app
    - Adds UpgradeModule
    - Configure AngularJS app
    - Add needed dependencies to package.json

Create an Angular Component

- Convert a controller to an Angular component
- Be pragmatic about refactoring
  - The desire might be to overhaul bad practices
  - Time might dictate another constraint
  - Investigate beforehand and lay a trail of desired conversion techniques for the team
- Create component TypeScript file
  - Copy in contents from controller
  - Manage issues, refactor minimally at 1st
- Create view HTML file
  - Convert/modify directives as needed for Angular

Convert Services to Providers

- This is *typically* easier to do than converting components
- Dependency graph of injected ng1 services
  - Convert in small doses
  - ng1 services can still be injected into ng components, services, directives
    - Set up service as Provider to be injected in ng module
- Downgrade ng providers if needing to inject into ng1
- Update unit testing

Routing Conversion

- Add an app.component
  - Renders the content of the active route
- Add Routing Module
  - Specifies routing configuration
- Refactor direct links and router viewports

Remove AngulsrJS

- Once a fully converted app to Angular
  - Remove hybrid bootstrap
  - Remove downgraded services/components
  - Clean up unused packages
  - Remove AngularJS references from index.html
- Angular CLI considerations
- Keep current with Angular and TS!

Useful References

- GitHub Repo of this Code
  - https://github.com/AllenConway/AngularJSAngularHybrid
  - Pull the different branches for different stages!!
    - AngularJS (*before*)
    - AngularHybrid (*after*)
- Angular Upgrade Guide
  - https://angular.io/guide/upgrade
- Using Webpack with Angular
  - https://angular.io/guide/webpack
- Webpack
  - https://webpack.js.org/

- AngularJS
  - https://angularjs.org
- Angular
  - https://angular.io
- Plunker (Angular Playground)
  - http://plnkr.co
- TypeScript
  - https://www.typescriptlang.org

Thank you! Q&A

http://c1.staticflickr.com/1/28/65098350_b7bd96f38_b.jpg

Allen Conway

- **Blog**: http://www.AllenConway.net
- **Twitter**: @AllenConway
- **GitHub**: https://github.com/AllenConway
- **Email**: AllenC@Magenic.com

Thank you for attending Visual Studio LIVE!