
Angular 2: An Overview

— Ashish Narasimham —

How do you create an end-to-end test?

- Uses the Jasmine test framework
 - Has its own syntax and learning curve
- Human-readable sentence structure
- Possible to create dependency injection and mock up all the external parts of what you're testing
- Already integrated into your browser

What I got out of it

- Very easy to get an application running: clone a repository from github, install the dependencies, and start your web server
- TypeScript is pretty great for adding type checking in a dynamically typed language like JS
- There is a whole lot of stuff to learn before you can be productive
- Built-in testing framework emphasizes the important but often-ignored step of testing and more importantly, automated testing

Here's what you may run into when you start

- Large learning curve - components, modules, directives, TypeScript, Karma/Jasmine/Protractor. It has its own language.
- How do all the pieces interact with each other?
- Volume of code written is more than usual. Declarations of components in multiple places including imports, code can be split multiple ways and into multiple files
- Debugging takes some time to get used to, has some quirks
- Small nuance: lots of open and close brackets and parentheses in the code, esp. in test code. Hard to debug issues with mismatches

Bottom line: greatly improved your code structure

- TypeScript, NativeScript
- MVC paradigm vs pure JS
- Testing framework
- Universal - web, mobile, and desktop-compatible
- Can be deployed on many different platforms
- Node package manager
- Modern, uses many concepts from the recent age of software

Demos

Angular2 Basics

angular2-quickstart

What are all these files?

Adding a variable

angular2-quickstart

- Arrays
- Class variables
- HTML template syntax
- Directives
- Events

Exploring the sandbox

angular-sandbox

- Routers
- Custom Directives
- Root component vs children
- Custom objects
- Tests

Tour of Heroes

tour_of_heroes

- Concept: services
- “Real-world” application

Workshops

Feel free to reference any of the source code in the workshop repository

Let's add our own variable

angular2-quickstart

- Add a variable to the home page that prints the current date and time
- Hint: "d = new Date();" will give you the current date/time

Loops in HTML

angular2-quickstart

- Create an array of usernames in your component (source for this data can be constant)
- Display that array on the page using `*ngFor`

Creating a service

angular2-quickstart

- Create a simple service that provides a string constant to your component

Create a unit test

For the brave

Add a unit test that checks for the presence of the string you created in your service

Other building blocks

- Creating routes
- Connecting to a database
- Writing advanced tests (events, navigation, etc)
- Tour of Heroes

Where to go from here

- **Understand** the basic building blocks, a lot of which we discussed briefly
- **Learn by doing.** Use the sandbox to see how it fits
- **Pitch it** to your client. It's a hot technology, and for good reason
