GETTING STARTED WITH
ANGULAR 2

FluentConf
March 7, 2016

Jeremy Wilken
@gnomeontherun
HI, I’M JEREMY

Experience Engineer at Teradata
Lives in Austin, TX
Author of Ionic in Action, Angular 2 in Action
Certified Beer Judge and Homebrewer
INTRODUCE YOURSELF TO AT LEAST 2 PEOPLE NEAR TO YOU

We’ve got 3 hours together, let’s make it more fun.
AGENDA

• Web applications with Angular 2
• Setup the app
• Add a service
• Create a component
• Break (30 mins)
• Add more components
• Routing
• Testing
• Review/Questions
FOLLOWING ALONG

• Code along with me as much you like
• Checkout the git tags as we go
• Use Vagrant, NVM, or Plunkr if needed
<table>
<thead>
<tr>
<th>Stock</th>
<th>Current Price</th>
<th>Change</th>
<th>Change Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAPL</td>
<td>$101.18</td>
<td>-$0.69</td>
<td>-0.68%</td>
</tr>
<tr>
<td>GOOG</td>
<td>$698.77</td>
<td>$3.61</td>
<td>0.52%</td>
</tr>
<tr>
<td>FB</td>
<td>$106.31</td>
<td>$0.58</td>
<td>0.55%</td>
</tr>
<tr>
<td>AMZN</td>
<td>$565.50</td>
<td>$2.70</td>
<td>0.48%</td>
</tr>
<tr>
<td>TWTR</td>
<td>$18.83</td>
<td>-$0.34</td>
<td>-1.77%</td>
</tr>
<tr>
<td>Stock</td>
<td>Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AAPL</td>
<td>REMOVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOOG</td>
<td>REMOVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FB</td>
<td>REMOVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMZN</td>
<td>REMOVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TWTR</td>
<td>REMOVE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
WEB APPLICATIONS WITH ANGULAR 2
ES6 IN A NUTSHELL

• New features: classes, modules, promises, arrows, generators, and more.

• Additive only, doesn’t remove existing features.

• Most features can be transpiled to current ES5.

• Will continue to add features.
TYPESCRIPT IN A NUTSHELL

• Transpiler to compile down to ES6/5/3.

• Adds strict typings to variables.

• “Typed Superset” that only extends JavaScript syntax.
var bill = 20;
var tip = document.getElementById('tip').value; // 5

var total = bill + tip; // 25? Are you sure?
JAVASCRIPT TYPE LIMITATIONS

TypeScript can enforce the right type is assigned, here tip is invalid

```javascript
var bill:number = 20;
var tip:number =
    (<HTMLInputElement>document.getElementById('tip')).value;
var total:number = bill + tip; // 25!?
```
JAVASCRIPT TYPE LIMITATIONS

TypeScript does not convert values, you must

```javascript
var bill:number = 20;
var tip:number = parseFloat((<HTMLInputElement>document.getElementById('tip')).value);

var total:number = bill + tip; // 25!
```
JAVASCRIPT IS NOW A COMPILED LANGUAGE
JAVASCRIPT IS NOW A COMPILABLE LANGUAGE
const stocks = ['AAPL', 'GOOG', 'FB', 'AMZN'];
let service = 'https://angular2-in-action-api.herokuapp.com';
let name = 'Jeremy';

let template = `'
<h1>Template by ${name}</h1>
<p>I can write markup without escaping and write on multiple lines!</p>
`;
ES6 arrow functions

```javascript
let stocks = ['aapl', 'goog', 'fb', 'amzn'];

class Dashboard {
  constructor() {
    stocks.forEach(stock => {
      this.add(stock.toUpperCase());
    });
  }
  add(stock) {
    this.stocks.push(stock);
  }
}
```
class Dashboard {
    constructor(service) {
        this.symbols = service.get();
    }

    reset() {
        this.symbols = [];
    }
}
import {service} from './service';

export class Dashboard {
    constructor() {
        this.symbols = service.get();
    }
}
ES6/TYPESCRIPT PRIMER

ES7 decorators

```javascript
@Component({
  selector: 'app'
})
export class App {
}
```
interface StockInterface {
    symbol: string,
    lastTradePriceOnly: number,
    change: number,
    changeInPercent: number
}

let stocks: Array<StockInterface> = [];

Typescript interfaces and type declarations
ANGULAR2 APP ARCHITECTURE

- Component based (tree of components)
- Modularity (ES6 modules)
- Dependency injection
- Services
- Component based routing
APP ARCHITECTURE
THE BASE

Uses the Angular CLI which has a local server. Files are transpiled with TypeScript.
Angular CLI
THE APP COMPONENT

Creating the app component, and bootstrapping it
<table>
<thead>
<tr>
<th>Stock</th>
<th>Current Price</th>
<th>Change Amount</th>
<th>Change Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAPL</td>
<td>$101.18</td>
<td>-$0.69</td>
<td>-0.68%</td>
</tr>
<tr>
<td>GOOG</td>
<td>$698.77</td>
<td>$3.61</td>
<td>0.52%</td>
</tr>
<tr>
<td>FB</td>
<td>$106.31</td>
<td>$0.58</td>
<td>0.55%</td>
</tr>
<tr>
<td>AMZN</td>
<td>$565.50</td>
<td>$2.70</td>
<td>0.48%</td>
</tr>
<tr>
<td>TWTR</td>
<td>$18.83</td>
<td>-$0.34</td>
<td>-1.77%</td>
</tr>
</tbody>
</table>
THE STOCK SERVICE
Creating a reusable service to manage stocks collection
ES6 Module

Stock Service

App Component

index.html

Angular CLI
AAPL
$101.18
-$0.69 (-0.68%)

GOOG
$698.77
$3.61 (0.52%)

FB
$106.31
$0.58 (0.55%)

AMZN
$565.50
$2.70 (0.48%)

TWTR
$18.83
-$0.34 (-1.774%)

REST API Service
THE SUMMARY COMPONENT

Using directives, types, and bindings
5 Summary Components

AAPL
$101.18
-$0.69 (-0.68%)

GOOG
$698.77
$3.61 (0.52%)

FB
$106.31
$0.58 (0.55%)

AMZN
$565.50
$2.70 (0.48%)

TWTR
$18.83
-$0.34 (-1.774%)
THE DASHBOARD COMPONENT

Loading data, using services, and importing components.
AAPL
$101.18
-$0.69 (-0.68%)

GOOG
$698.77
$3.61 (0.52%)

FB
$106.31
$0.58 (0.55%)

AMZN
$565.50
$2.70 (0.48%)

TWTR
$18.83
-$0.34 (-1.77%)
THE MANAGE COMPONENT

Use a form to manage the list of stocks.
### Manage Component

<table>
<thead>
<tr>
<th>Stock</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAPL</td>
<td>REMOVE</td>
</tr>
<tr>
<td>GOOG</td>
<td>REMOVE</td>
</tr>
<tr>
<td>FB</td>
<td>REMOVE</td>
</tr>
<tr>
<td>AMZN</td>
<td>REMOVE</td>
</tr>
<tr>
<td>TWTR</td>
<td>REMOVE</td>
</tr>
</tbody>
</table>
LET’S REVIEW

- Angular CLI generates app files and tooling
- App component is bootstraped to start rendering
- Services are used to load and manage data
- It is best to nest simple components
- Pipes and directives manipulate rendered data
- Routing determines which component to display
WHERE DO YOU LIVE, AND WHERE WOULD YOU RATHER LIVE?

Talk to Your Neighbor Short Break
LET’S BUILD AN ANGULAR 2 APP
https://github.com/gnomeontherun/angular2-fluent-conf

Follow README instructions
START THE APP

$ ng init -n tracker
CHECKOUT STEP 1

$ git checkout -f step1
• Angular CLI generates project files
• Default files need some styling and structure
• Angular CLI runs local server
• Angular CLI runs testing suite
<table>
<thead>
<tr>
<th>Stock</th>
<th>Price</th>
<th>Change</th>
<th>% Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAPL</td>
<td>$101.18</td>
<td>-$0.69</td>
<td>-0.68%</td>
</tr>
<tr>
<td>GOOG</td>
<td>$698.77</td>
<td>$3.61</td>
<td>0.52%</td>
</tr>
<tr>
<td>FB</td>
<td>$106.31</td>
<td>$0.58</td>
<td>0.55%</td>
</tr>
<tr>
<td>AMZN</td>
<td>$565.50</td>
<td>$2.70</td>
<td>0.48%</td>
</tr>
<tr>
<td>TWTR</td>
<td>$18.83</td>
<td>-$0.34</td>
<td>-1.77%</td>
</tr>
</tbody>
</table>
WHAT IS YOUR FAVORITE SONG, AND WHAT DOES IT MEAN TO YOU?

Talk to Your Neighbor Short Break
STOCKS SERVICE

$ ng generate service stocks
CHECKOUT STEP 2

$ git checkout -f step2
REVIEW STOCKS SERVICE

• Class manages list of stocks
• Http service helps makes REST calls
• Interfaces enforce typings
• Annotate class to make it injectable
REST API Service
IF YOU WROTE A BOOK FOR O’REILLY, WHAT WOULD YOUR COVER ANIMAL BE?
SUMMARY COMPONENT

$ ng generate component summary
CHECKOUT STEP 3

$ git checkout -f step3
REVIEW SUMMARY

COMPONENT

• Controller accepts an input value
• Controller has methods for template
• Pipes format data for display
• NgIf conditionally includes an element
• Use NgClass to apply colors to cards
5 Summary Components

AAPL
$101.18
-$0.69 (-0.68%)

GOOG
$698.77
$3.61 (0.52%)

FB
$106.31
$0.58 (0.55%)

AMZN
$565.50
$2.70 (0.48%)

TWTR
$18.83
-$0.34 (-1.77%)
BREAK TIME
IF YOU COULD HAVE ANY CAR, WHAT WOULD IT BE AND WHY?

Talk to Your Neighbor Short Break
DASHBOARD COMPONENT

$ ng generate component dashboard
CHECKOUT STEP 4

$ git checkout -f step4
REVIEW DASHBOARD COMPONENT

• Controller loads the stock data

• Stocks service handles the actual http requests

• Services and directives must be injected

• ngFor loops over stocks and displays a card
<table>
<thead>
<tr>
<th>Stock</th>
<th>Price</th>
<th>Change</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAPL</td>
<td>$101.18</td>
<td>-$0.69</td>
<td>-0.68%</td>
</tr>
<tr>
<td></td>
<td>$3.61</td>
<td>+0.52%</td>
<td></td>
</tr>
<tr>
<td>GOOG</td>
<td>$698.77</td>
<td>+3.61</td>
<td>+0.52%</td>
</tr>
<tr>
<td></td>
<td>$0.58</td>
<td>+0.55%</td>
<td></td>
</tr>
<tr>
<td>FB</td>
<td>$106.31</td>
<td>+0.58</td>
<td>+0.55%</td>
</tr>
<tr>
<td></td>
<td>$2.70</td>
<td>+0.48%</td>
<td></td>
</tr>
<tr>
<td>AMZN</td>
<td>$565.50</td>
<td>+2.70</td>
<td>+0.48%</td>
</tr>
<tr>
<td></td>
<td>+2.70</td>
<td>+0.48%</td>
<td></td>
</tr>
</tbody>
</table>
IF YOU COULD TRAVEL SOMEWHERE IN TIME, WHERE AND WHEN WOULD YOU GO?

Talk to Your Neighbor Short Break
MANAGE COMPONENT

$ ng generate component manage
CHECKOUT STEP 5

$ git checkout -f step5
REVIEW MANAGE COMPONENT

• Controller manages the list of stocks
• Form builder describes a form
• ngFormModel is used with form inputs
• Event bindings send data to controller
### Add Stock

<table>
<thead>
<tr>
<th>Stock</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAPL</td>
<td>REMOVE</td>
</tr>
<tr>
<td>GOOG</td>
<td>REMOVE</td>
</tr>
<tr>
<td>FB</td>
<td>REMOVE</td>
</tr>
<tr>
<td>AMZN</td>
<td>REMOVE</td>
</tr>
<tr>
<td>TWTR</td>
<td>REMOVE</td>
</tr>
</tbody>
</table>
WHAT IS YOUR BIGGEST PET PEEVE, AND WHEN WAS THE LAST TIME SOMEONE DID IT?

Talk to Your Neighbor Short Break
ROUTING
CHECKOUT STEP 6

$ git checkout -f step6
REVIEW ROUTING

• Configure routes to define links
• Components must be directly referenced in config
• Router link directive links to components
• Router outlet is where content is rendered
## Router Outlet

<table>
<thead>
<tr>
<th>Add Stock</th>
<th>REMOVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAPL</td>
<td>REMOVE</td>
</tr>
<tr>
<td>GOOG</td>
<td>REMOVE</td>
</tr>
<tr>
<td>FB</td>
<td>REMOVE</td>
</tr>
<tr>
<td>AMZN</td>
<td>REMOVE</td>
</tr>
<tr>
<td>TWTR</td>
<td>REMOVE</td>
</tr>
</tbody>
</table>
WHAT IS THE WEIRDEST HABIT YOU HAVE?
(WE ALL HAVE ONE)

Talk to Your Neighbor Short Break
UNIT TESTING
CHECKOUT STEP 7

$ git checkout -f step7
REVIEW UNIT TESTING

• Angular CLI generates test files
• Test the controller and service methods
• Use Angular’s testing services
• Testing is expected to become easier
QUESTIONS

use #topics-angular slack channel if you have other questions later
THANK YOU!

Jeremy Wilken
@gnomeontherun

http://manning.com/books/angular-2-in-action
https://speakerdeck.com/gnomeontherun