



Launch Yourself Into the Angular 2 and TypeScript Space

Kurt Wiersma
@kwiersma

Agenda

- * TypeScript Intro
- * Mapping Concepts from Angular 1 to 2
- * Bootstrapping and Module Loading
- * Components, Services, and Routing
- * Template Syntax



TypeScript

TypeScript Intro

- * <http://typescriptlang.org>
 - * TypeScript lets you write JavaScript the way you really want to.
 - * TypeScript is a typed superset of JavaScript that compiles to plain JavaScript.
 - * Any browser. Any host. Any OS. Open Source.
- * AngularJS 2 is implemented in TypeScript

TypeScript

Select...

Share

JavaScript

Run

```
1 function greeter(person) {  
2     return "Hello, " + person;  
3 }  
4  
5 var user = "Jane User";  
6  
7 document.body.innerHTML = greeter(user);  
8  
9
```

```
1 function greeter(person) {  
2     return "Hello, " + person;  
3 }  
4  
5 var user = "Jane User";  
6  
7 document.body.innerHTML = greeter(user);  
8
```

JavaScript is Valid TypeScript

TypeScript Syntax

```
1  /// <reference path='../_all.ts' />
2
3  module djleague {
4
5    export class FantasyTeamService {
6
7      public teams: FantasyTeam[];
8
9      private httpService: ng.IHttpService;
10
11     constructor ($http: ng.IHttpService) {
12       this.httpService = $http;
13     }
14
15     getTeams(): ng.IPromise<FantasyTeam[]> {
16       return this.httpService.get('/api/teams')
17         .then(function (response) {
18           var data = response.data;
19           this.teams = new Array<FantasyTeam>();
20
21           for (var i = 0; i < data.length; i++) {
22             var team: FantasyTeam = new FantasyTeam();
23             team.id = data[i].id;
24             team.name = data[i].name;
25             team.draftorder = data[i].draftorder;
26             team.owner = data[i].owner;
27             this.teams.push(team);
28           }
29
30           return this.teams;
31         });
32     }
33   }
34 }
35 }
```

```
1  /// <reference path='../_all.ts' />
2  var djleague;
3
4  djleague = (function (djleague) {
5    var FantasyTeamService = (function () {
6
7      function FantasyTeamService($http) {
8        this.httpService = $http;
9      }
10
11      FantasyTeamService.prototype.getTeams = function () {
12        return this.httpService.get('/api/teams').then(function (response) {
13          var data = response.data;
14          this.teams = new Array();
15
16          for (var i = 0; i < data.length; i++) {
17            var team = new djleague.FantasyTeam();
18            team.id = data[i].id;
19            team.name = data[i].name;
20            team.draftorder = data[i].draftorder;
21            team.owner = data[i].owner;
22            this.teams.push(team);
23          }
24
25          return this.teams;
26        });
27      };
28
29      return FantasyTeamService;
30    })();
31
32    djleague.FantasyTeamService = FantasyTeamService;
33
34 })(djleague || (djleague = {}));
```

TypeScript



Features

- * Classes
- * Modules
- * Interfaces
- * Generics
- * Arrow Functions
 - * Better “this”
- * Imports
- * Type Definitions

```
1  /// <reference path='../_all.ts' />
2
3  module djleague {
4
5      export class FantasyTeamService {
6
7          public teams: FantasyTeam[];
8
9          private httpService: ng.IHttpService;
10
11         constructor ($http: ng.IHttpService) {
12             this.httpService = $http;
13         }
14
15         getTeams(): ng.IPromise<FantasyTeam[]> {
16             return this.httpService.get('/api/teams')
17                 .then(function (response) {
18                     var data = response.data;
19                     this.teams = new Array<FantasyTeam>();
20
21                     for (var i = 0; i < data.length; i++) {
22                         var team: FantasyTeam = new FantasyTeam();
23                         team.id = data[i].id;
24                         team.name = data[i].name;
25                         team.draftorder = data[i].draftorder;
26                         team.owner = data[i].owner;
27                         this.teams.push(team);
28                     }
29
30                     return this.teams;
31                 });
32             }
33
34         }
35     }
```

Getting Started

* Install:

- * npm install -g typescript

* Compile:

- * tsc

* tsconfig.json

* Typings manages type definitions

```
1  {
2    "compileOnSave": false,
3    "compilerOptions": {
4      "declaration": false,
5      "emitDecoratorMetadata": true,
6      "experimentalDecorators": true,
7      "mapRoot": "",
8      "module": "system",
9      "moduleResolution": "node",
10     "noEmitOnError": true,
11     "noImplicitAny": false,
12     "outDir": "../dist/",
13     "rootDir": ".",
14     "sourceMap": true,
15     "target": "es5",
16     "inlineSources": true
17   },
18   "files": [
19     "app.ts",
20     "typings.d.ts"
21   ],
22   "exclude": [
23     "node_modules"
24   ]
25 }
```

Tooling

- * Editor support
- * Refactoring
- * Source maps for debugging



Mapping Concepts From Angular 1 to 2

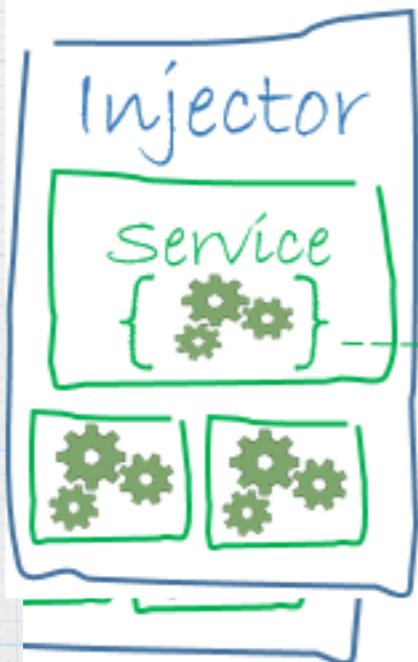
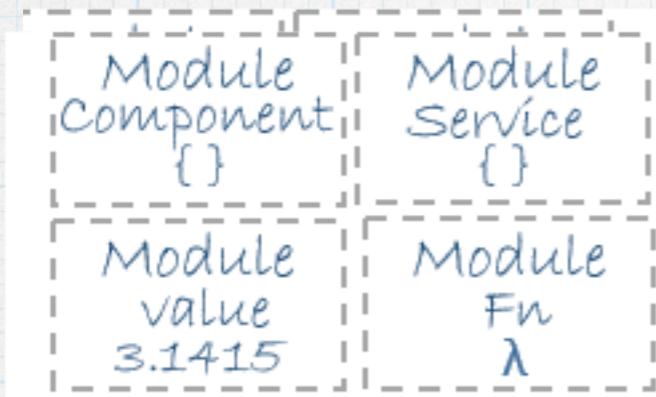
Angular 1 to 2

Angular 1.x	Angular 2.x
ES5	TypeScript
Controllers	Components
Filters	Pipes
ng-app	bootstrap
ng-class	[ngClass]
ng-click	(click)
ng-if	*ngIf

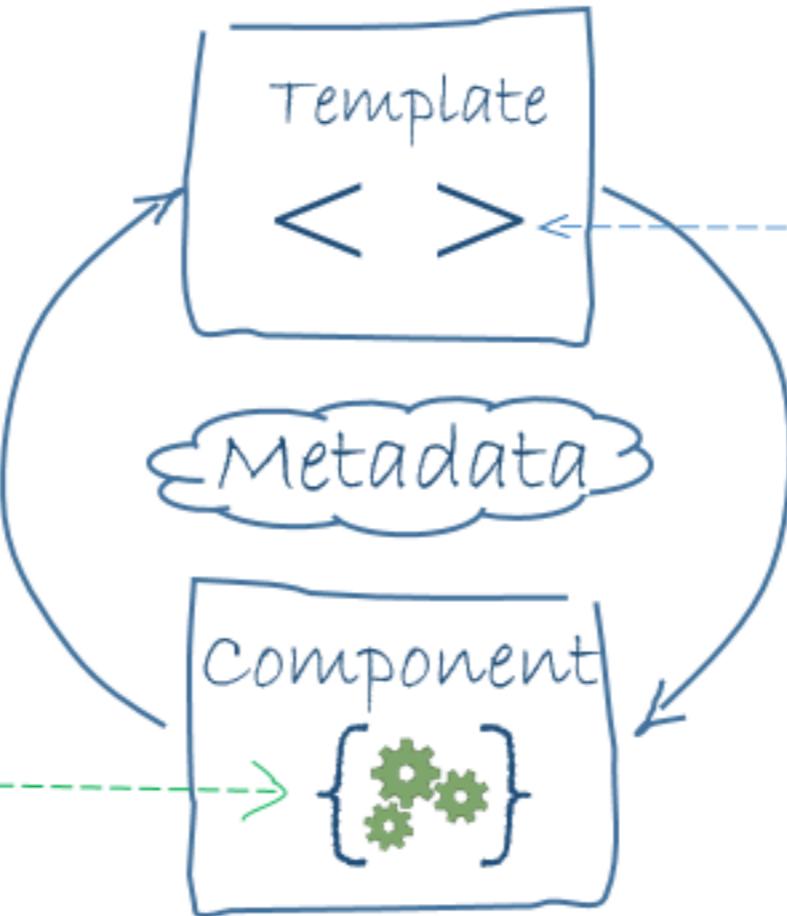
Angular 1 to 2

Angular 1.x	Angular 2.x
ng-model	[ngModel]
ng-repeat	*ngFor
ng-show	[hidden]
Promises	Observables

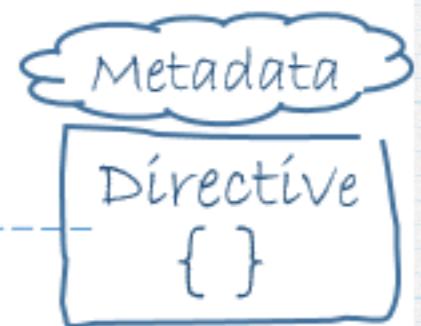
<https://angular.io/docs/ts/latest/cookbook/upgrade-al-a2-quick-reference.html>



Property Binding



Event Binding



Angular App Architecture



Getting Started

Angular CLI

- * npm install -g angular-cli
- * ng new my-ng2-project
- * ng serve
- * Watches, compiles, and serves
- * ng generate component my-component

Bootstrapping

index.html

```
22 <body style="...>
23
24     <app-component>Loading...</app-component>
25
26     <footer...>
27
28
29
30
31
32
33
34
35
36
37
38
39
40 {{#each _scripts.polyfills}}
41 <script src="{{_.}}></script>
42 {{/each}}
43
44 <script>
45     System.import('system-config.js').then(function () {
46         System.import('main');
47     }).catch(console.error.bind(console));
48 </script>
49
50 </body>
```

main.ts

```
1 import { bootstrap } from '@angular/platform-browser-dynamic';
2 import { enableProdMode } from '@angular/core';
3 import { APP_ROUTER_PROVIDERS } from './app/app.routes';
4 import { AppComponent, environment } from './app/';
5
6 if (environment.production) {
7     enableProdMode();
8 }
9
10 bootstrap(AppComponent, [APP_ROUTER_PROVIDERS])
11     .catch(err => console.error(err));
```

Breaking News!

app/app.module.ts (minimal)

```
import { NgModule }      from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import
  { AppComponent }  from './app.component';

@NgModule({
  imports: [ BrowserModule ],
  declarations: [ AppComponent ],
  bootstrap:  [ AppComponent ]
})
export class AppModule { }
```

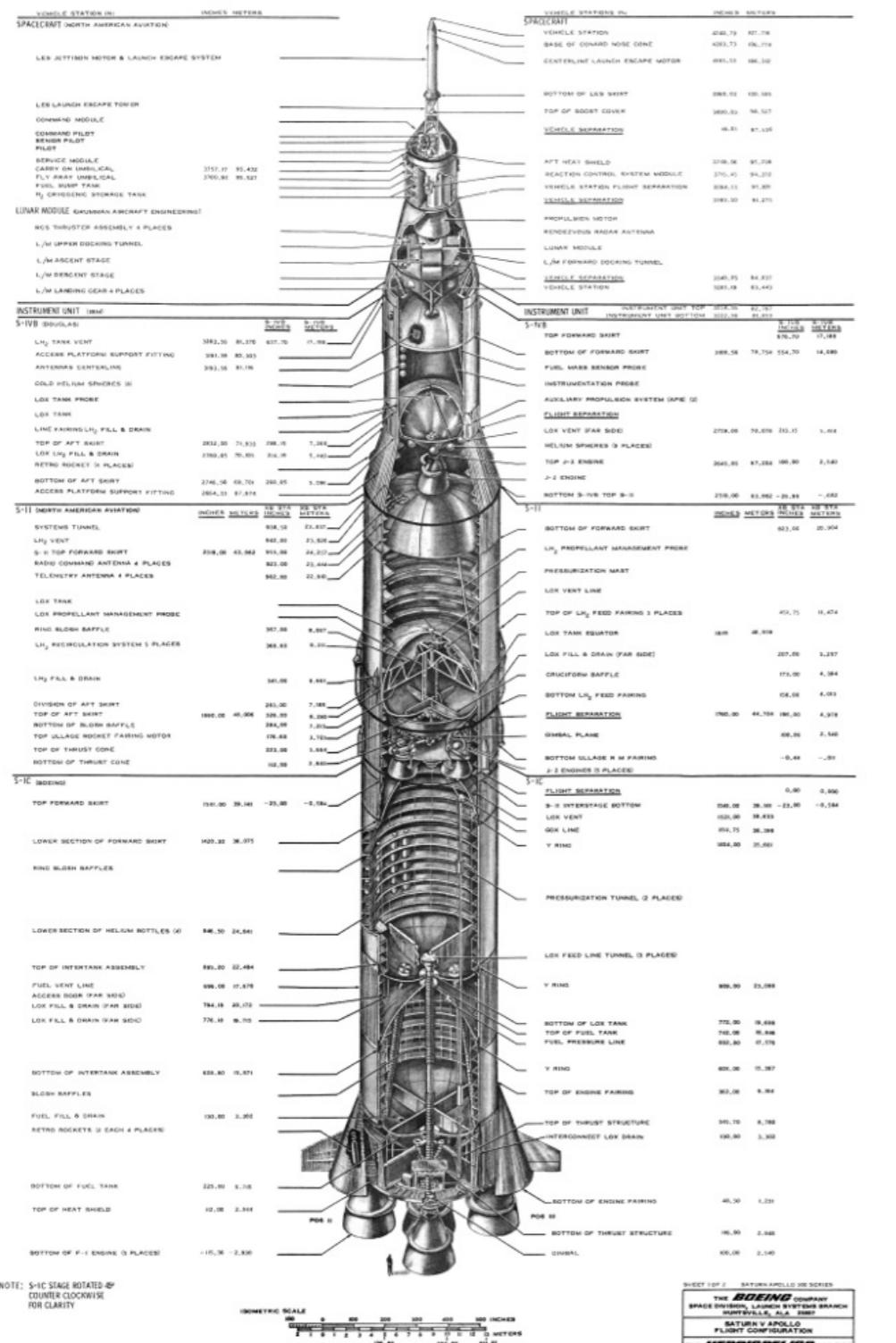
app/main.ts (dynamic)

```
// The browser platform with a compiler
import { platformBrowserDynamic } from '@angular/platform-
browser-dynamic';

// The app module
import { AppModule } from './app.module';

// Compile and launch the module
platformBrowserDynamic().bootstrapModule(AppModule);
```

SATURN V APOLLO FLIGHT CONFIGURATION





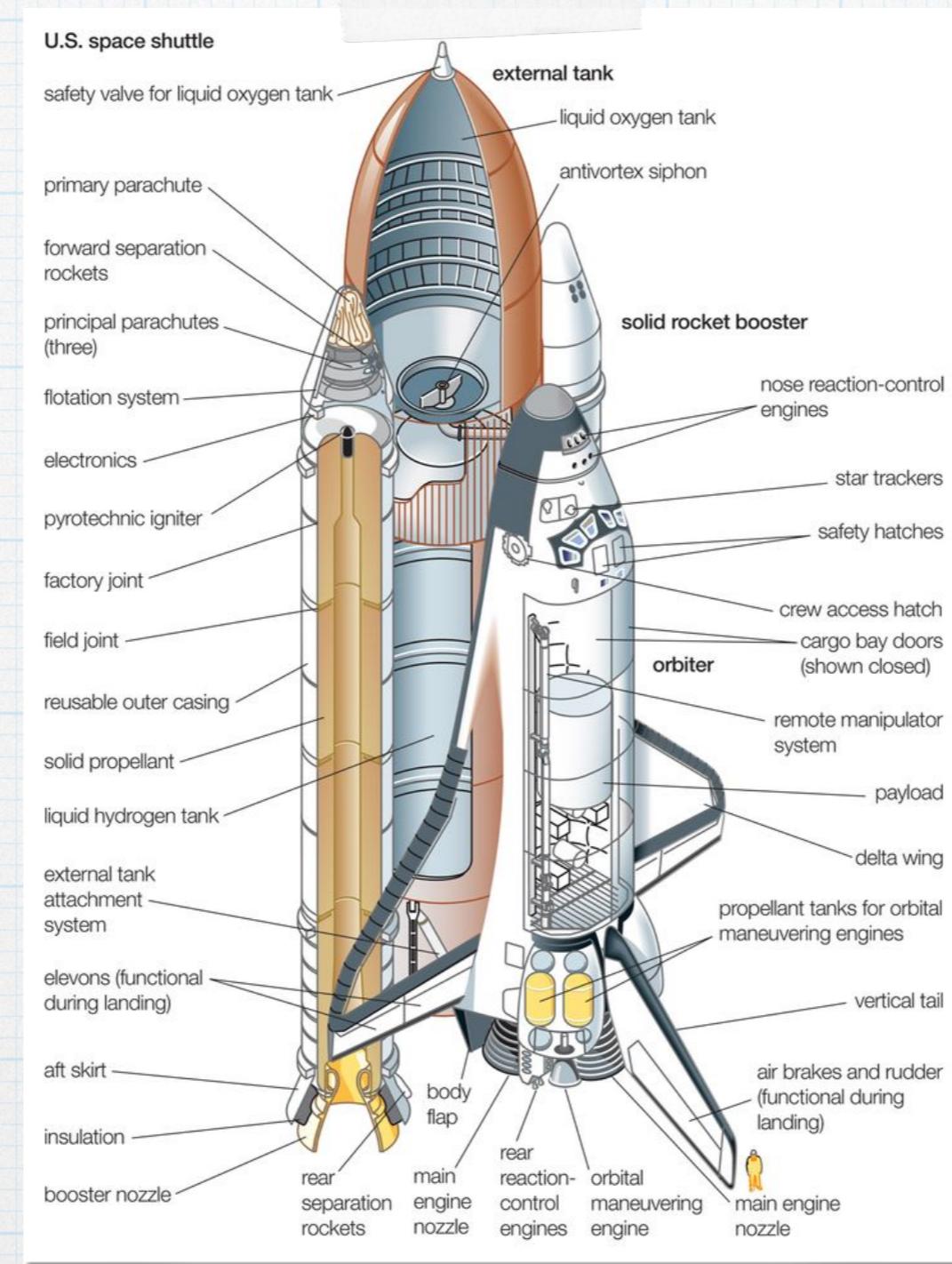
Draft System Demo

Component Layout

App Component

Navbar Component

Router Outlet
(Draft Component, Fantasy Teams
Component)



Component Anatomy

```
1 import {ROUTER_DIRECTIVES} from '@angular/router';
2 import {Component} from '@angular/core';
3 import {Location} from '@angular/common';
4
5 @Component({
6   moduleId: module.id,
7   selector: 'navbar',
8   templateUrl: 'navbar.component.html',
9   directives: [ROUTER_DIRECTIVES]
10 })
11 export class NavbarComponent {
12   constructor(private location: Location) {}
13
14   highlight(path: string) {
15     if (path === '/') {
16       path = '';
17     }
18     let currentPath: string = this.location.path();
19     return currentPath === path;
20   }
21 }
```

```
1 <header class="navbar navbar-inverse navbar-fixed-top">
2   <nav class="container">
3     <div class="navbar-header">
4       <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
5             data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
6         <span class="sr-only">Toggle navigation</span>
7         <span class="icon-bar"></span>
8       </button>
9       <span class="navbar-brand">Angular 2 Draft</span>
10    </div>
11    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
12      <ul class="nav navbar-nav">
13        <li [class.active]="highlight('/')">
14          <a [routerLink]="/">Players</a>
15        </li>
16        <li [class.active]="highlight('/teams')">
17          <a [routerLink]="/teams">Teams</a>
18        </li>
19      </ul>
20    </div>
21  </nav>
22 </header>
```

Component Lifecycle

1. **OnChanges** - runs first and when a data bound input property value changes
2. **OnInit** - after the first OnChanges
3. **DoCheck** - during every Angular change detection cycle
4. **AfterViewInit** - after init of the component's views and child views
5. **OnDestroy** - just before Angular destroys the component

Lifecycle Example

```
1 import {Component, OnInit} from "angular2/core";
2 import {PlayerService} from "../shared/services/PlayerService";
3 import {Player} from "../shared/model/model";
4 import {PlayerFilterComponent} from "./PlayerFilterComponent";
5
6 @Component({
7   selector: 'player-list',
8   templateUrl: 'app/players/playerlist.component.html',
9   directives: [PlayerFilterComponent]
10})
11 export class PlayerListComponent implements OnInit {
12   players: Player[];
13   filteredPlayers: Player[];
14
15   constructor(private playerService: PlayerService) {}
16
17 ⑪ ngOnInit() {
18    this.playerService.getPlayers()
19      .subscribe((response: Player[]) => {
20        this.players = this.filteredPlayers = response;
21      });
22  }
23
24  filterChanged(lastname: string) {...}
40}
```

Component Input

`draft.component.html`

```
1  <div class="row">
2    <div class="col-md-3">
3      <draft-order [teams]="teams"></draft-order>
4    </div>
5    <div class="col-md-9">
6      <player-list></player-list>
7    </div>
8  </div>
```

`DraftOrderComponent.ts`

```
4  @Component({
5    selector: 'draft-order',
6    templateUrl: 'app/draft/drafterorder.component.html'
7  })
8  export class DraftOrderComponent {
9    @Input()
10   teams: FantasyTeam[] = [];
11
12  constructor() {
13
14  }
15}
```

Events

PlayerFilterComponent

```
1  <form>
2    <strong>Last Name:</strong>
3    <input type="text" size="20"
4      [(ngModel)]="model.lastname"
5      (keyup)="filterChanged($event)" /> ←
6  </form>

3  @Component({
4    selector: 'player-filter',
5    templateUrl: 'app/players/playerfilter.component.html'
6  })
7  export class PlayerFilterComponent {
8
9    model: { lastname: string } = { lastname: null };
10
11   @Output()
12   changed: EventEmitter<string> = new EventEmitter(); ←
13
14   constructor() {}
15
16   filterChanged(event: any) { ←
17     event.preventDefault();
18     this.changed.emit(this.model.lastname); //Raise changed event
19   }
20
21 }
```

Events

```
<player-filter (changed)="playerFilterChanged($event)"></player-filter>
```

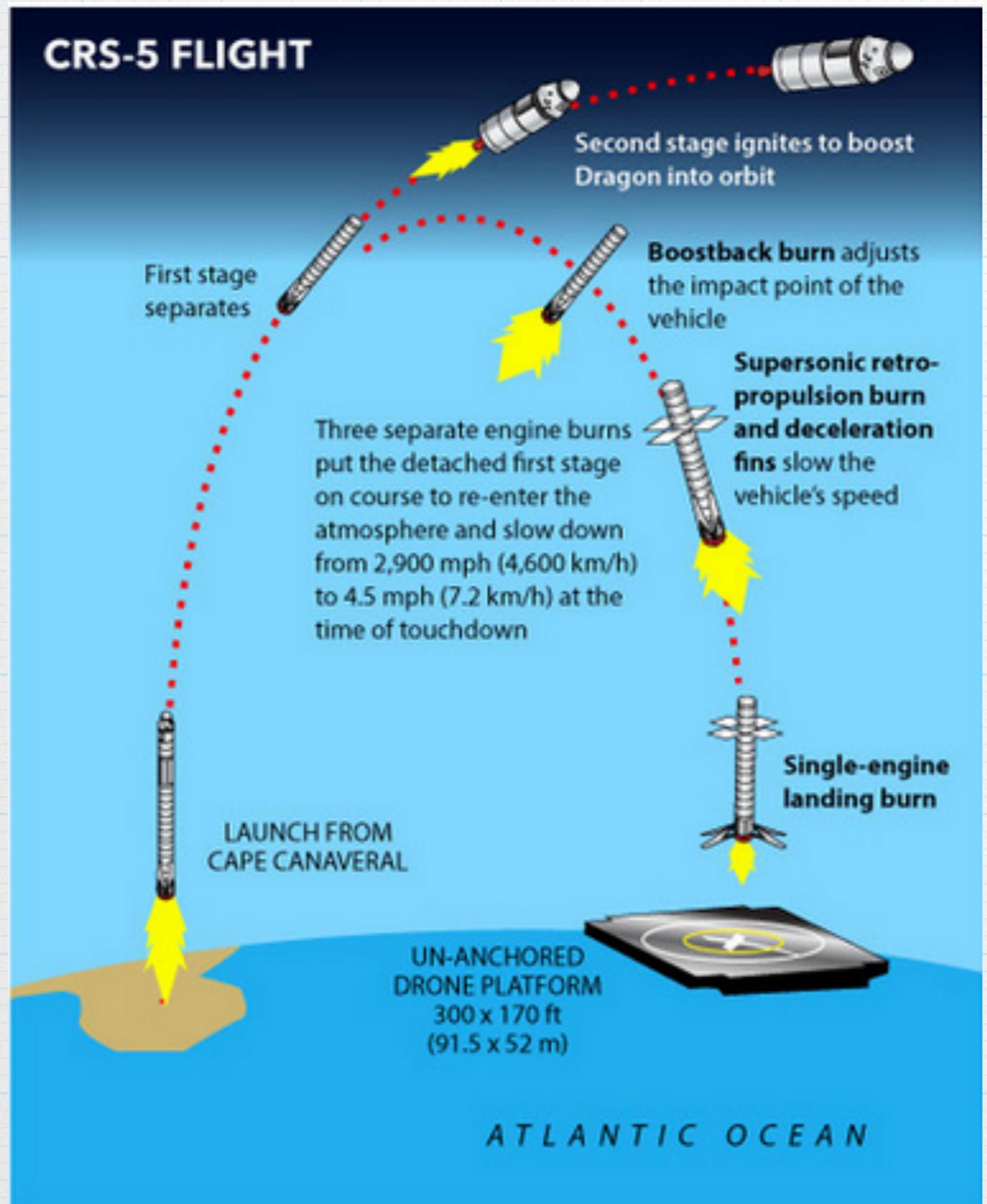
```
6   @Component({
7     selector: 'player-list',
8     templateUrl: 'app/players/playerlist.component.html',
9     directives: [PlayerFilterComponent] ←
10    })
11   export class PlayerListComponent implements OnInit {
12     players: Player[];
13     filteredPlayers: Player[];
14
15     constructor(private playerService: PlayerService) {}
16
17   ngOnInit() {...}
18
19
20   playerFilterChanged(lastname: string) { ←
21     if (lastname && this.players) {
22       lastname = lastname.toLowerCase();
23       var filtered = this.players.filter((player: Player) => {
24         var match = false;
25         var regExp = new RegExp(`^${lastname}`, "i");
26         if (player.lastname.search(regExp) != -1) {
27           match = true;
28         }
29         return match;
30       });
31       this.filteredPlayers = filtered;
32     } else {
33       this.filteredPlayers = this.players;
34     }
35   }
36
37
38
39
40 }
```

Dependency Injection

```
10  @Component({
11    moduleId: module.id,
12    selector: 'app-component',
13    providers: [HTTP_PROVIDERS, FantasyTeamService, PlayerService, LogService],
14    templateUrl: 'app.component.html',
15    directives: [ROUTER_DIRECTIVES, NavbarComponent],
16    pipes: []
17  })
18  export class AppComponent {
19
20    constructor() {
21
22    }
23
24 }
```

```
1  <navbar></navbar>
2
3  <main class="container">
4    <router-outlet></router-outlet>
5    <br /><br />
6  </main>
```

CRS-5 FLIGHT



Routing

Routing Demo

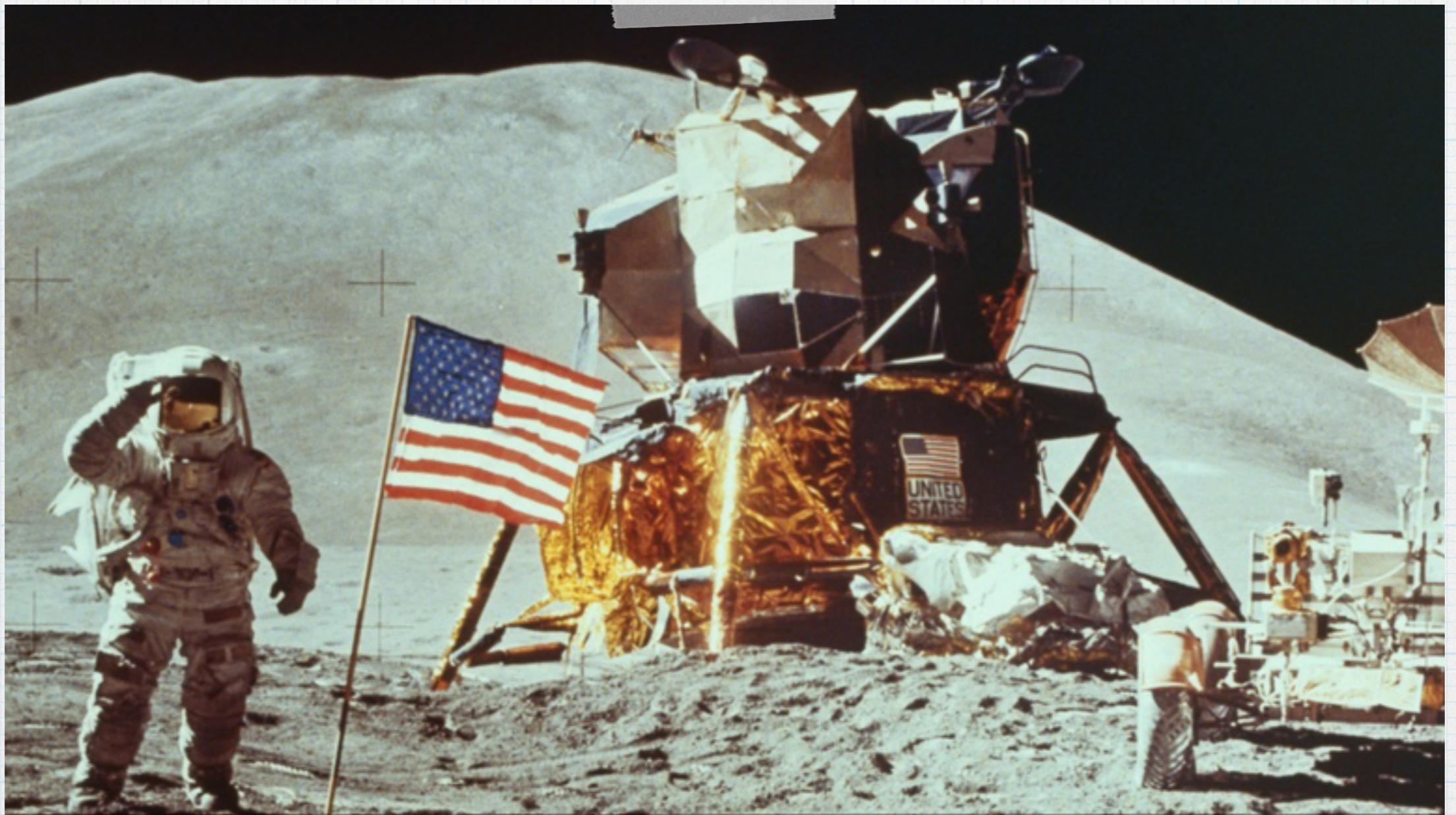
Routing

app.routes.ts

```
1 import {FantasyTeamsComponent} from './fantasyteams/FantasyTeamsComponent';
2 import {DraftComponent} from './draft/DraftComponent';
3 import {RouterConfig, provideRouter} from '@angular/router';
4
5 const appRoutes: RouterConfig = [
6   { path: '', terminal: true, component: DraftComponent },
7   { path: 'teams', component: FantasyTeamsComponent }
8 ];
9
10 export const APP_ROUTER_PROVIDERS = [
11   provideRouter(appRoutes)
12 ];
```

Linking To A Route

```
1 <header class="navbar navbar-inverse navbar-fixed-top">
2   <nav class="container">
3     <div class="navbar-header">
4       <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
5             data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
6         <span class="sr-only">Toggle navigation</span>
7         <span class="icon-bar"></span>
8       </button>
9       <span class="navbar-brand">Angular 2 Draft</span>
10      </div>
11      <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
12        <ul class="nav navbar-nav">
13          <li [class.active]="highlight('/')">
14            <a [routerLink]="/">Players</a> ←
15          </li>
16          <li [class.active]="highlight('/teams')">
17            <a [routerLink]="/teams">Teams</a> ←
18          </li>
19        </ul>
20      </div>
21    </nav>
22  </header>
```



Services

```
1 import {Injectable} from 'angular2/core';
2 import {Http, Response} from "angular2/http";
3 import {Observable} from "rxjs/Rx";
4 import {LogService} from "./LogService";
5 import {FantasyTeam} from "../model/model";
6
7 @Injectable()
8 export class FantasyTeamService {
9     private teamsObservable: Observable<FantasyTeam[]>;
10    private teamsData: FantasyTeam[];
11
12    constructor(private http: Http, private logService: LogService) {}
13
14    getFantasyTeams(): Observable<FantasyTeam[]> {
15        //...
16        if (this.teamsData) {
17            // if `data` is available just return it as `Observable`
18            this.logService.log('teams.json loaded from cache');
19            return Observable.of(this.teamsData);
20        } else {
21            if(this.teamsObservable) {
22                // if `this.observable` is set then the request is in progress
23                // return the `Observable` for the ongoing request
24                return this.teamsObservable;
25            } else {
26                // create the request, store the `Observable` for subsequent subscribers
27                this.logService.log('requesting teams.json');
28                this.teamsObservable = this.http.get("/teams.json")
29                    .map((response:Response) => {
30                        return <FantasyTeam[]> response.json();
31                    })
32                    .do((val) => {
33                        this.teamsData = val;
34                        // when the cached data is available we don't need the 'Observable' anymore
35                        this.teamsObservable = null;
36                        this.logService.debug('teams.json fetched', this.teamsData);
37                    })
38                    .catch(this.handleError)
39                    // make it shared so more than one subscriber can get the result
40                    .share();
41            }
42            return this.teamsObservable;
43        }
44    }
45
46
47    handleError(error: any) {
48        this.logService.log('Error: ' + error);
49        return Observable.throw(error.json().error || 'Server error');
50    }
51
52 }
```



Template Syntax

Template Syntax

- * One way data binding:
 - * <h1>{{ player.name }}</h1>
- * Event binding:
 - * <a (click)=handleClick()>Details

Two Way Data Binding

`[(ngModel)] = "someValue"`



`= "someValue"`

Two Way Binding

- * Two-way data binding:
 - * <input type="text" [(ngModel)]="player.name" />

DOM Binding

```
<button [disabled]="isUnchanged">Save</button>
```

Structural Directives

```
1 <table class="table table-striped">
2   <thead>
3     <tr>
4       <th>Draft Order</th>
5       <th>Team</th>
6       <th>Owner</th>
7       <th></th>
8     </tr>
9   </thead>
10  <tbody>
11    <tr *ngFor="let team of teams">
12      <td style="...">{{ team.draftorder }}</td>
13      <td>{{ team.name }}</td>
14      <td>{{ team.owner }}</td>
15      <td></td>
16    </tr>
17  </tbody>
18 </table>
```

Resources

- * angular.io
- * Angular 2 Style Guide
 - * <https://angular.io/docs/ts/latest/guide/style-guide.html>
- * Dan Wahlin's Starter App
 - * <https://github.com/DanWahlin/Angular2-JumpStart>
- * John Papa's Angular 2 Course
 - * <https://app.pluralsight.com/library/courses/angular-2-first-look>
- * ng-book 2

Questions?

- * github.com/kwiersma/djleague-ng2
- * [@kwiersma](https://twitter.com/@kwiersma)