Launch Yourself Into the Angular 2 and TypeScript Space

Kurt Wiersma
@kwiersma

About Me

* Software Development Leader/Manager
* From Minneapolis, MN
* Over 15 years of development
* Favs: Python, Typescript, C#

Agenda

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


USAF
IMAX
FIGHTER PILOT
NOW SHOWING
IMAX 3D
NARRATED BY AL ROKER
3D SUN
AS CLOSE AS YOU EVER WANT TO GET
EXIT
DONATIONS
Fund Our Flight Plan

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 | Run | JavaScript

```
function greeter(person) {
    return "Hello, " + person;
}

var user = "Jane User";

document.body.innerHTML = greeter(user);
```

```
function greeter(person) {
    return "Hello, " + person;
}

var user = "Jane User";

document.body.innerHTML = greeter(user);
```

JavaScript is Valid TypeScript

TypeScript Syntax

```typescript
/// <reference path='../_all.ts' />

module djleague {

    export class FantasyTeamService {

        public teams: FantasyTeam[];

        private httpService: ng.IHttpService;

        constructor ($http: ng.IHttpService) {
            this.httpService = $http;
        }

        getTeams(): ng.IPromise<FantasyTeam[]> {
            return this.httpService.get('/api/teams')
                .then(function (response) {
                    var data = response.data;
                    this.teams = new Array<FantasyTeam>();

                    for (var i = 0; i < data.length; i++) {
                        var team: FantasyTeam = new FantasyTeam();
                        team.id = data[i].id;
                        team.name = data[i].name;
                        team.draftorder = data[i].draftorder;
                        team.owner = data[i].owner;
                        this.teams.push(team);
                    }

                    return this.teams;
                });
        }

    }
}
```

TypeScript

```javascript
/// <reference path='../_all.ts' />
var djleague;

(function (djleague) {
    var FantasyTeamService = (function () {

        function FantasyTeamService($http) {
            this.httpService = $http;
        }

        FantasyTeamService.prototype.getTeams = function () {
            return this.httpService.get('/api/teams').then(function (response) {
                var data = response.data;
                this.teams = new Array();

                for (var i = 0; i < data.length; i++) {
                    var team = new djleague.FantasyTeam();
                    team.id = data[i].id;
                    team.name = data[i].name;
                    team.draftorder = data[i].draftorder;
                    team.owner = data[i].owner;
                    this.teams.push(team);
                }

                return this.teams;
            });
        };

        return FantasyTeamService;
    })();

    djleague.FantasyTeamService = FantasyTeamService;

})(djleague || (djleague = {}));
```

JavaScript

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

```
{
  "compileOnSave": false,
  "compilerOptions": {
    "declaration": false,
    "emitDecoratorMetadata": true,
    "experimentalDecorators": true,
    "mapRoot": "",
    "module": "system",
    "moduleResolution": "node",
    "noEmitOnError": true,
    "noImplicitAny": false,
    "outDir": "../dist/",
    "rootDir": ".",
    "sourceMap": true,
    "target": "es5",
    "inlineSources": true
  },
  "files": [
    "app.ts",
    "typings.d.ts"
  ],
  "exclude": [
    "node_modules"
  ]
}
```

Mapping Concepts From Angular 1 to 2

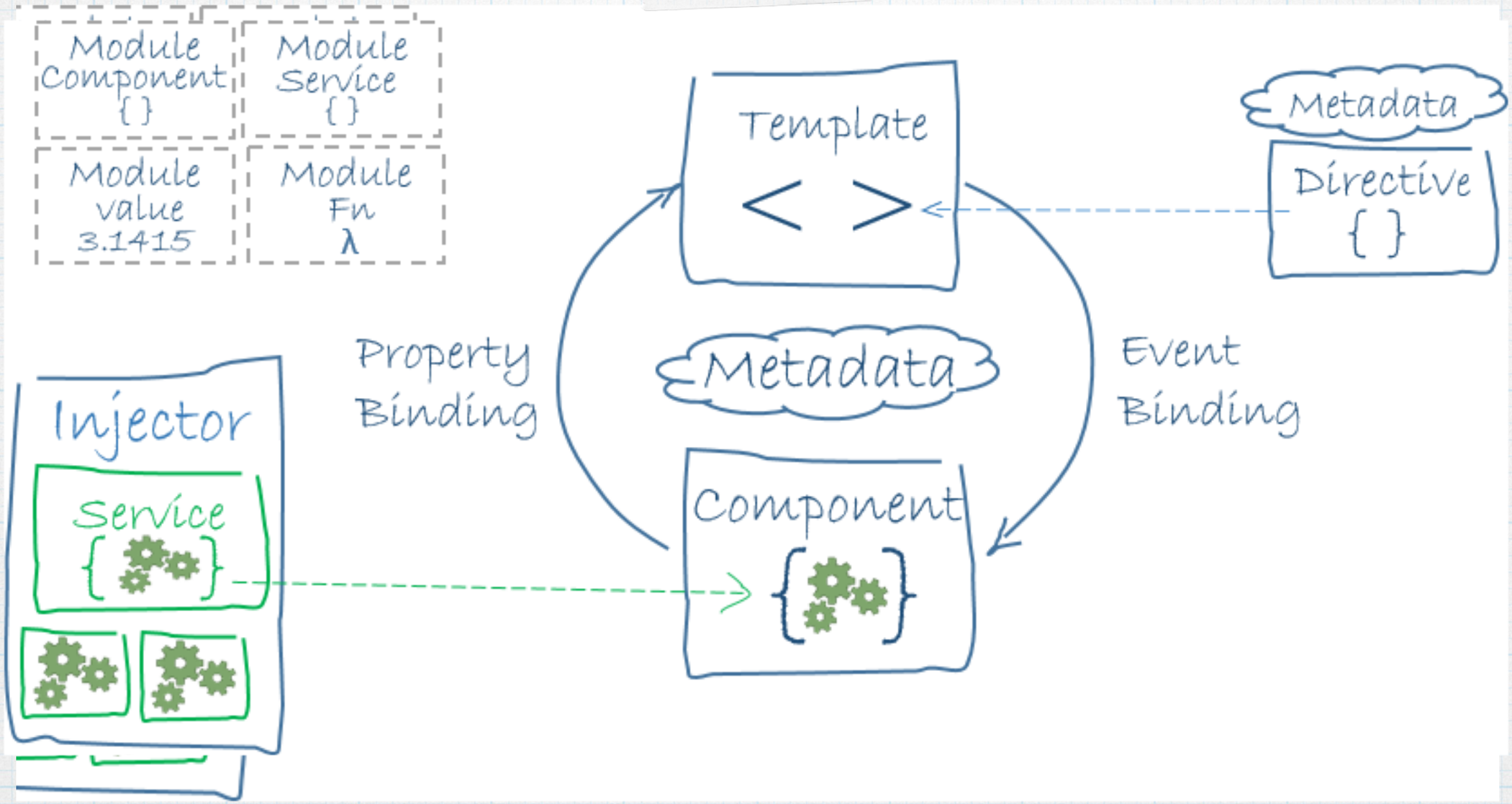
Angular 1 to 2

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/a1-a2-quick-reference.html



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

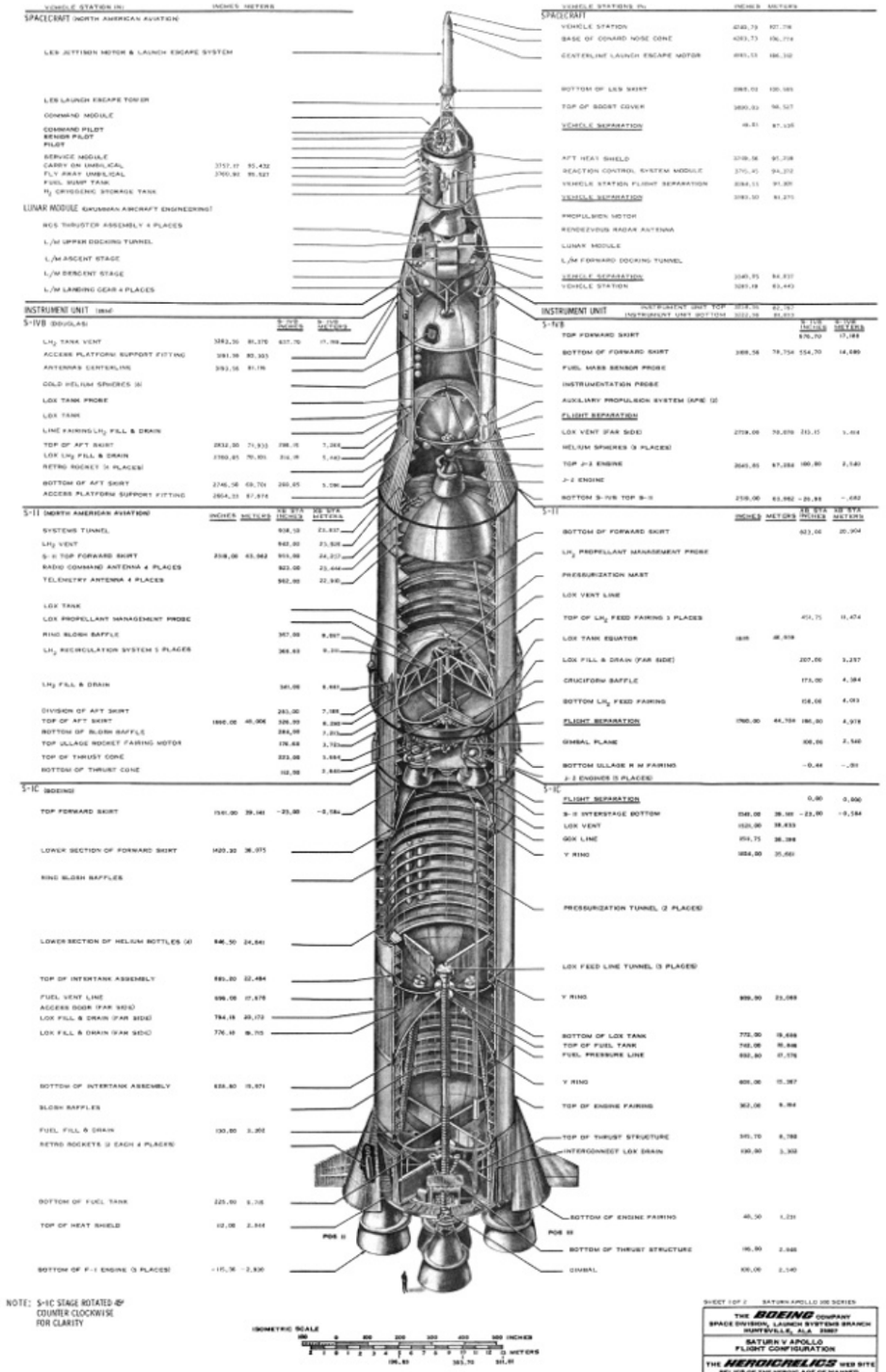
index.html

```
58    <script>
59      System.config({
60        packages: {
61          app: {
62            format: 'register',
63            defaultExtension: 'js'
64          }
65        }
66      });
67      System.import('app.js').then(null, console.error.bind(console));
68    </script>
```

app.ts

```
1   import {bootstrap} from 'angular2/platform/browser';
2   import {AppComponent} from './app/AppComponent';
3
4   bootstrap(AppComponent, []);
```

SATURN V APOLLO FLIGHT CONFIGURATION



NOTE: S-IC STAGE ROTATED 45° COUNTER CLOCKWISE FOR CLARITY

ISOMETRIC SCALE

SHEET 1 OF 2 SATURN APOLLO 300 SERIES

THE BOEING COMPANY
SPACE DIVISION, LAUNCH SYSTEMS BRANCH
HUNTSVILLE, ALA. 35807

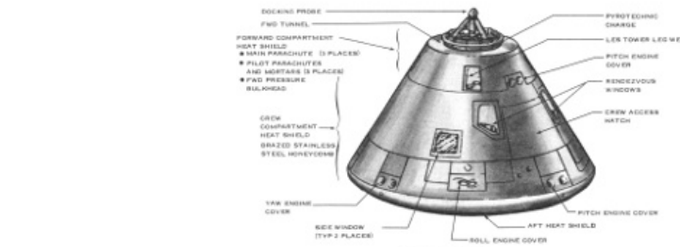
SATURN V APOLLO
FLIGHT CONFIGURATION

THE HEROICRELICS WEB SITE
RELICS OF THE HEROIC AGE OF MANNED SPACE FLIGHT
HTTP://HEROICRELICS.ORG

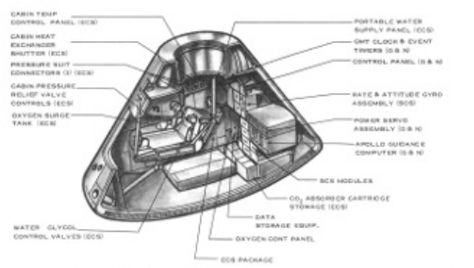
DRAWING ORIGINATED BY: HUNTSVILLE ENGINEERING
DATE: 1 JANUARY 1963
DRAWN BY: DON SPRAGUE

FOR ADDITIONAL COPIES CONTACT:
PUBLIC RELATIONS OFFICE, P. O. BOX 840
OR PHONE 205-895-6320

SATURN V APOLLO FLIGHT CONFIGURATION

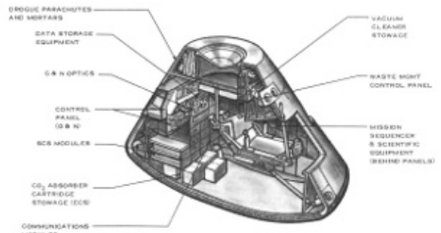


COMMAND MODULE



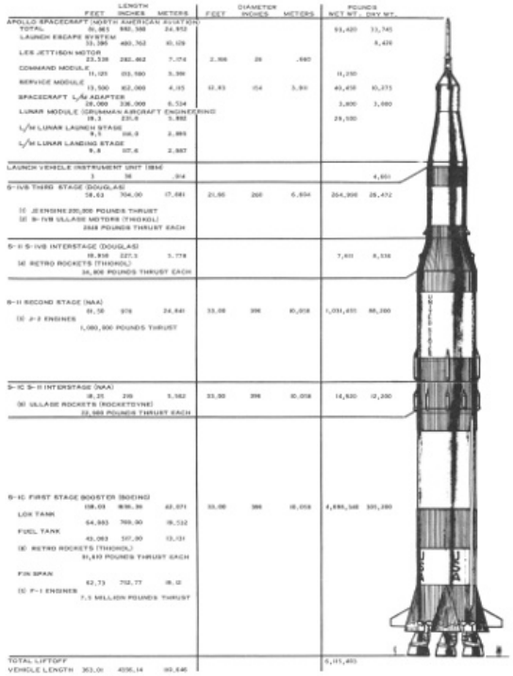
LEFT HAND VIEW OF FORWARD EQUIPMENT COMPARTMENT

COMMAND MODULE CONTROLS & DISPLAYS



RIGHT HAND VIEW OF FORWARD EQUIPMENT COMPARTMENT

A6L APOLLO BLOCK II PRESSURE SUITS & PORTABLE LIFE SUPPORT SYSTEM



SHEET 2 OF 2 SATURN APOLLO 300 SERIES

THE BOEING COMPANY
SPACE DIVISION, LAUNCH SYSTEMS BRANCH
HUNTSVILLE, ALA. 35807

SATURN V APOLLO
FLIGHT CONFIGURATION

THE HEROICRELICS WEB SITE
RELICS OF THE HEROIC AGE OF MANNED SPACE FLIGHT
HTTP://HEROICRELICS.ORG

DRAWING ORIGINATED BY: HUNTSVILLE ENGINEERING
DATE: 1 JANUARY 1963
DRAWN BY: DON SPRAGUE

FOR ADDITIONAL COPIES CONTACT:
PUBLIC RELATIONS OFFICE, P. O. BOX 840
PHONE 205-895-6320

Components

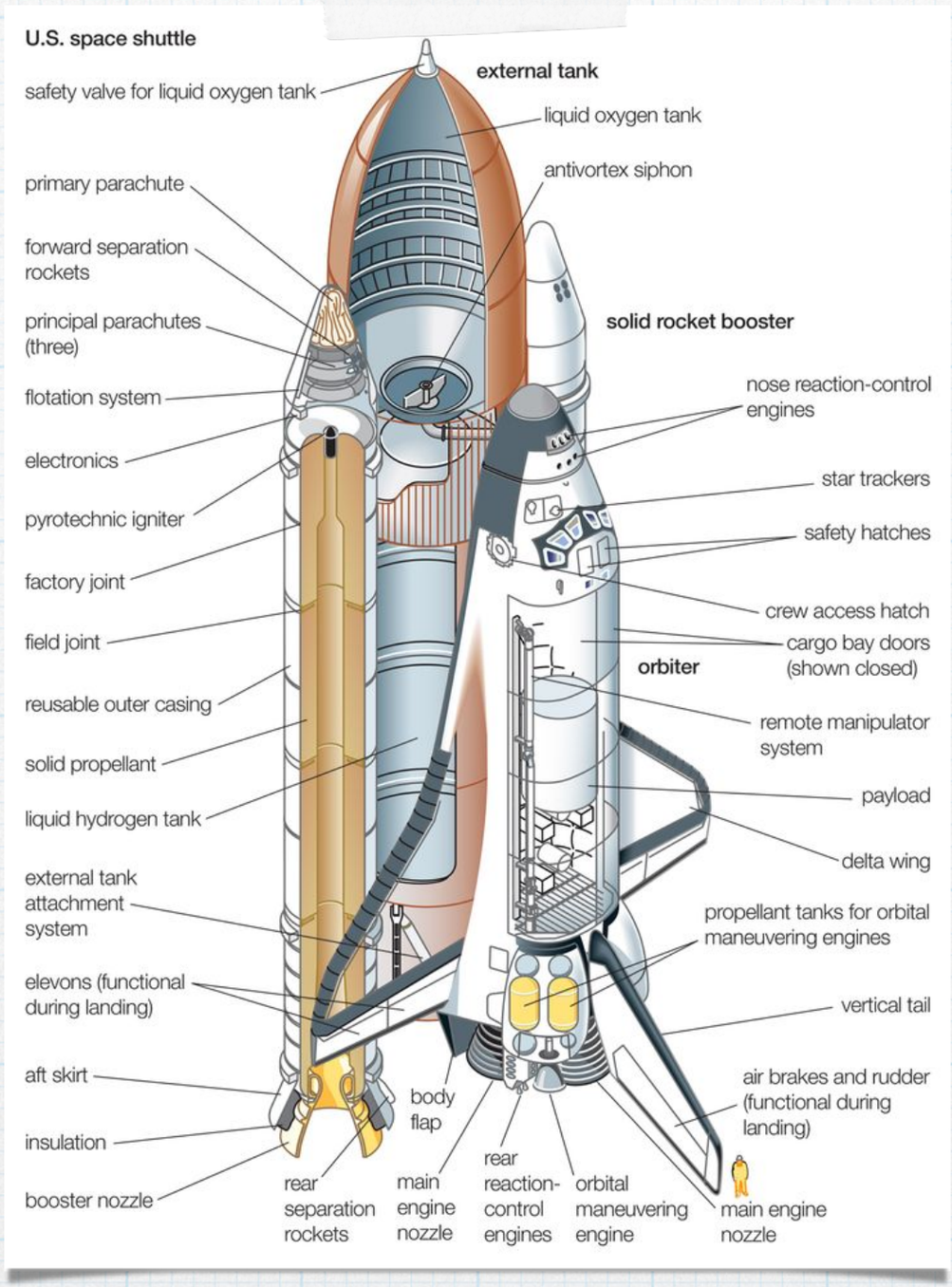
Draft System Demo

Component Layout

App Component

Navbar Component

Router Outlet
(Draft Component, Fantasy Teams Component)



Component Anatomy

```ts
import {RouterLink, Location} from 'angular2/router';
import {Component} from "angular2/core";

@Component({
    selector: 'navbar',
    templateUrl: 'app/navbar/navbar.component.html',
    directives: [RouterLink]
})
export class NavbarComponent {
    constructor(private location: Location) {}

    highlight(path: string) {
        if (path === '/') {
            path = '';
        }
        var currentPath:string = this.location.path();
        return currentPath === path;
    }
}
```

```html
<header class="navbar navbar-inverse navbar-fixed-top">
    <nav class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"...>
            <span class="navbar-brand">Angular 2 Draft</span>
        </div>
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
            <ul class="nav navbar-nav">
                <li [class.active]="highlight('/')">
                    <a [routerLink]="['Draft']">Players</a>
                </li>
                <li [class.active]="highlight('/teams')">
                    <a [routerLink]="['Teams']">Teams</a>
                </li>
            </ul>
        </div>
    </nav>
</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      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/draftorder.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

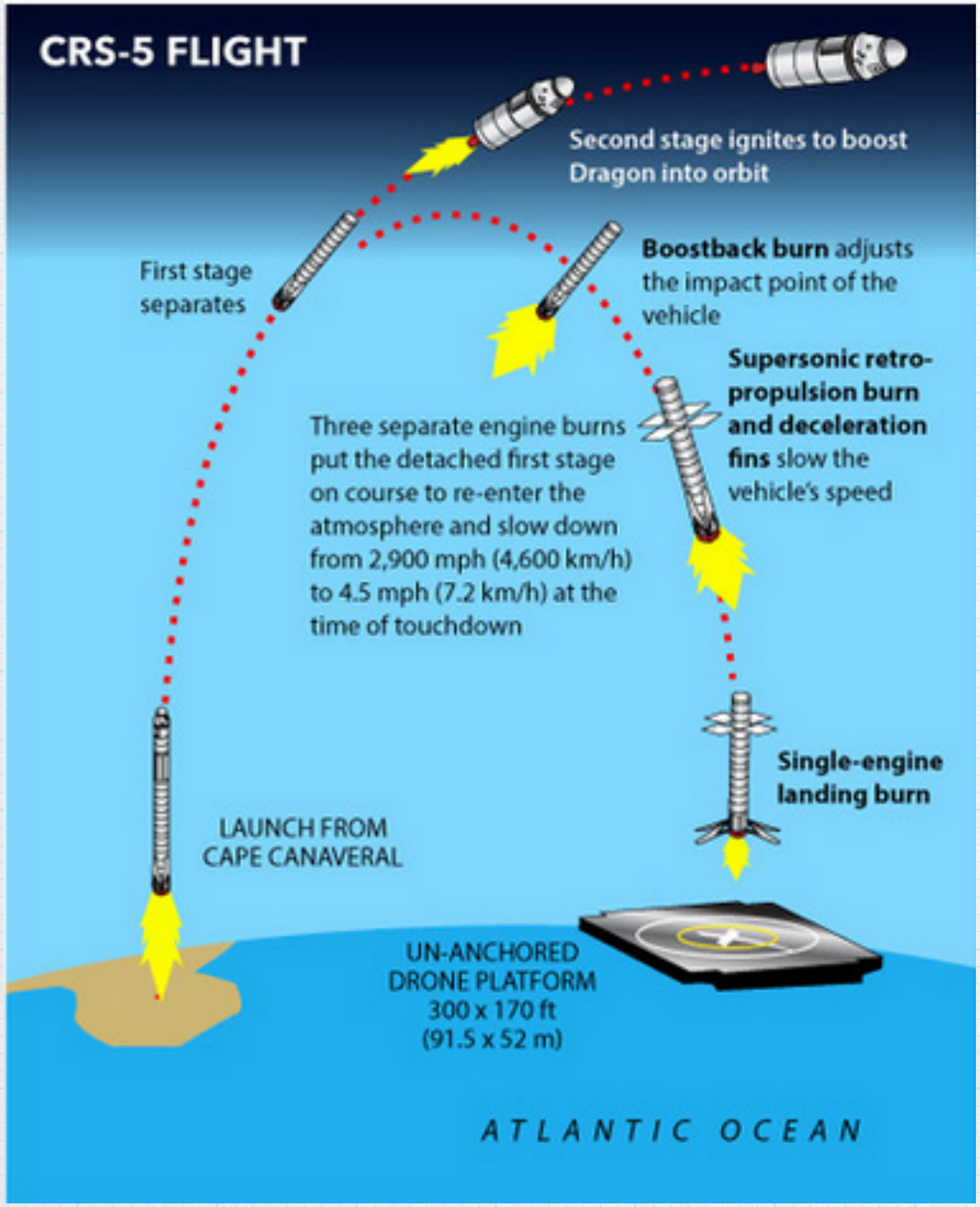
```
<player-filter (changed)="playerFilterChanged($event)"></player-filter>
```

```
 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() {...}
23
24          playerFilterChanged(lastname: string) {
25              if (lastname && this.players) {
26                  lastname = lastname.toLowerCase();
27                  var filtered = this.players.filter((player: Player) => {
28                      var match = false;
29                      var regExp = new RegExp("^" + lastname, "i");
30                      if (player.lastname.search(regExp) != -1) {
31                          match = true;
32                      }
33                      return match;
34                  });
35                  this.filteredPlayers = filtered;
36              } else {
37                  this.filteredPlayers = this.players;
38              }
39          }
40      }
```

Dependency Injection

```
12    @Component({
13      selector: 'app-component',
14      providers: [ROUTER_PROVIDERS, HTTP_PROVIDERS, FantasyTeamService, PlayerService, LogService],
15      templateUrl: 'app/app.component.html',
16      directives: [ROUTER_DIRECTIVES, NavbarComponent],
17      pipes: []
18    })
19    @RouteConfig([...].concat(CliRouteConfig))
23
24    export class AppComponent {
25
26      constructor() {
27
28      }
29
30    }
```

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


CRS-5 FLIGHT
Second stage ignites to boost Dragon into orbit
First stage separates
Boostback burn adjusts the impact point of the vehicle
Supersonic retro-propulsion burn and deceleration fins slow the vehicle's speed
Three separate engine burns put the detached first stage on course to re-enter the atmosphere and slow down from 2,900 mph (4,600 km/h) to 4.5 mph (7.2 km/h) at the time of touchdown
Single-engine landing burn
LAUNCH FROM CAPE CANAVERAL
UN-ANCHORED DRONE PLATFORM 300 x 170 ft (91.5 x 52 m)
ATLANTIC OCEAN

Routing

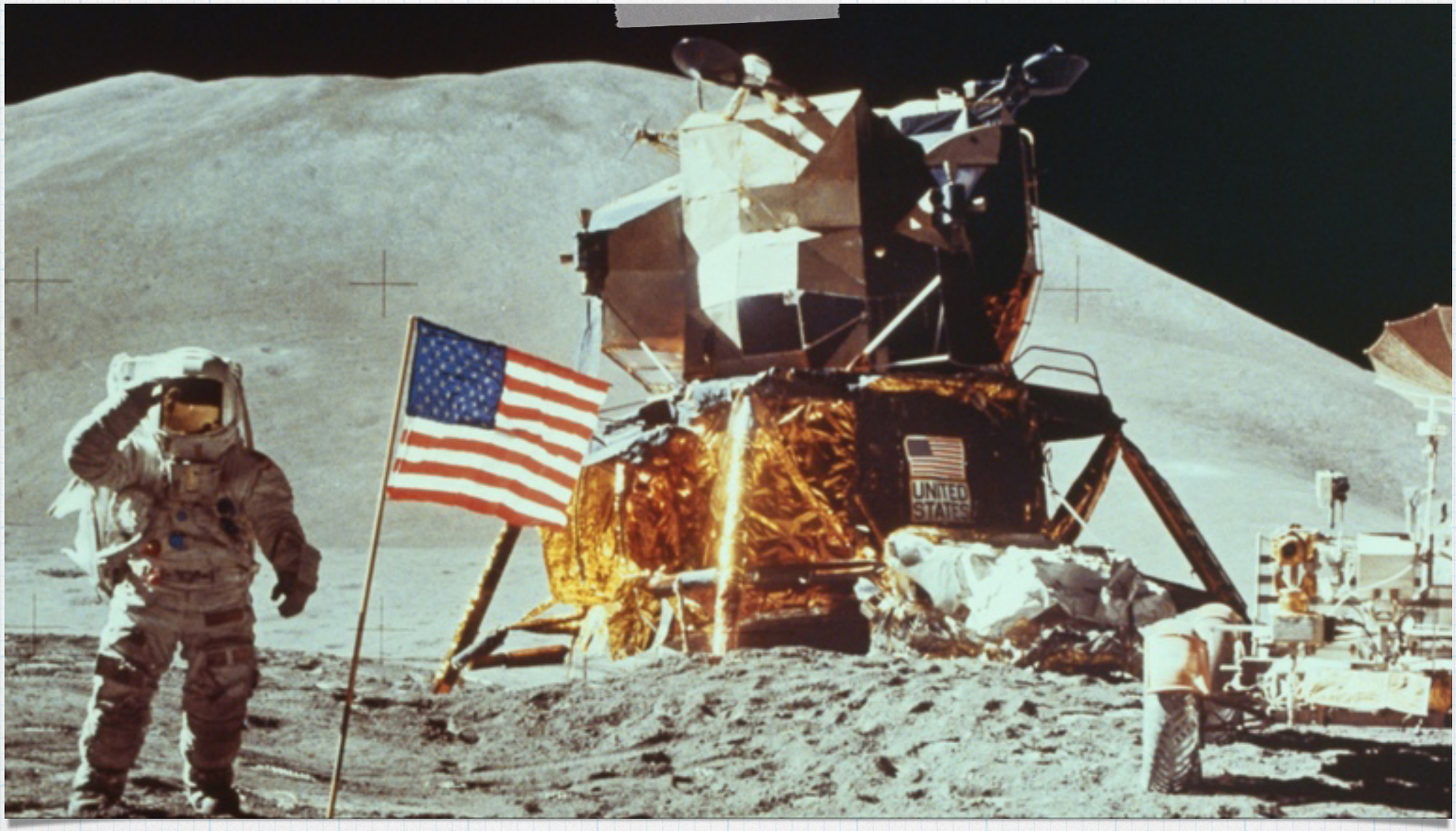
Routing Demo

Routing

```
1  import {Component} from 'angular2/core';
2  import {HTTP_PROVIDERS} from 'angular2/http';
3  import {RouteConfig, ROUTER_DIRECTIVES, ROUTER_PROVIDERS} from 'angular2/router';
4  import {CliRouteConfig} from './route-config';
5  import {DraftComponent} from "./draft/DraftComponent";
6  import {FantasyTeamService} from "./shared/services/FantasyTeamService";
7  import {FantasyTeamsComponent} from "./fantasyteams/FantasyTeamsComponent";
8  import {LogService} from "./shared/services/LogService";
9  import {NavbarComponent} from "./navbar/NavbarComponent";
10 import {PlayerService} from "./shared/services/PlayerService";
11
12 @Component({
13   selector: 'app-component',
14   providers: [ROUTER_PROVIDERS, HTTP_PROVIDERS, FantasyTeamService, PlayerService, LogService],
15   templateUrl: 'app/app.component.html',
16   directives: [ROUTER_DIRECTIVES, NavbarComponent],
17   pipes: []
18 })
19 @RouteConfig([
20   { path: '/', as: 'Draft', component: DraftComponent, useAsDefault: true },
21   { path: '/teams', as: 'Teams', component: FantasyTeamsComponent }
22 ].concat(CliRouteConfig))
23
24 export class AppComponent {
25
26   constructor() {
27
28   }
29
30 }
```

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"...>
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]="['Draft']">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

```
import {Injectable} from 'angular2/core';
import {Http, Response} from "angular2/http";
import {Observable} from "rxjs/Rx";
import {LogService} from "./LogService";
import {FantasyTeam} from "../model/model";

@Injectable()
export class FantasyTeamService {
    private teamsObservable: Observable<FantasyTeam[]>;
    private teamsData:FantasyTeam[];

    constructor(private http: Http, private logService: LogService) {}

    getFantasyTeams(): Observable<FantasyTeam[]> {
        //...
        if (this.teamsData) {
            // if `data` is available just return it as `Observable`
            this.logService.log('teams.json loaded from cache');
            return Observable.of(this.teamsData);
        } else {
            if(this.teamsObservable) {
                // if `this.observable` is set then the request is in progress
                // return the `Observable` for the ongoing request
                return this.teamsObservable;
            } else {
                // create the request, store the `Observable` for subsequent subscribers
                this.logService.log('requesting teams.json');
                this.teamsObservable = this.http.get("/teams.json")
                    .map((response:Response) => {
                        return <FantasyTeam[]> response.json();
                    })
                    .do((val) => {
                        this.teamsData = val;
                        // when the cached data is available we don't need the 'Observable' anymore
                        this.teamsObservable = null;
                        this.logService.debug('teams.json fetched', this.teamsData);
                    })
                    .catch(this.handleError)
                    // make it shared so more than one subscriber can get the result
                    .share();
                return this.teamsObservable;
            }
        }
    }

    handleError(error: any) {
        this.logService.log('Error: ' + error);
        return Observable.throw(error.json().error || 'Server error');
    }

}
```

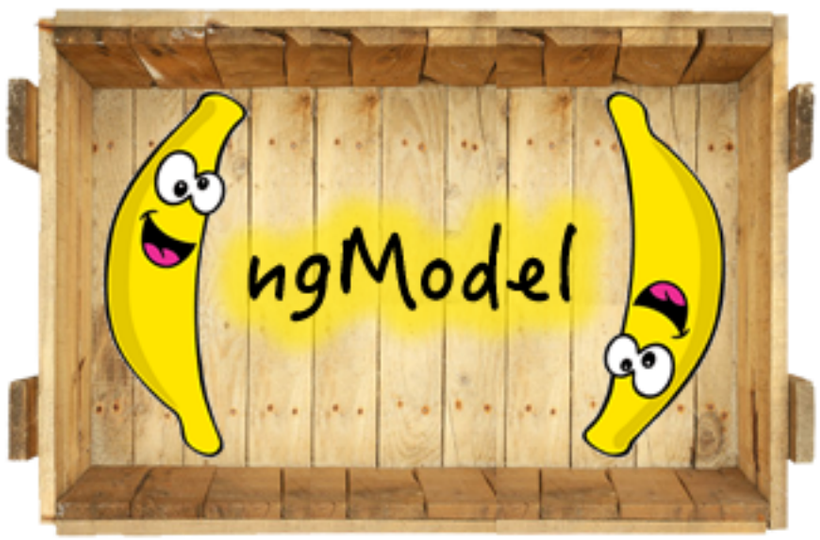
Template Syntax

Template Syntax

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

Two Data Binding

[(ngModel)] = "someValue"



= "someValue"

BenNadel.com

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 Walhin'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