APP LIFECYCLE

Suggest improvements

Tweet

Angular apps go through a multi-stage bootstrap and lifecycle process, and we can respond to various events as our app starts, runs, and creates/destroys components.

BOOTSTRAP

Angular 2 apps (currently) need to be bootstrapped using the root component for the app.

In your main JS file for our app, we put this:

```
import { bootstrap } from '@angular/platform-browser-dynam
import { Component } from '@angular/core';

// Annotation section
@Component({
  selector: 'my-app',
```

```
  template: '<h1>Hello {{ name }}</h1>'
})
// Component controller
class MyApp {
  constructor() {
    this.name = 'Max';
  }
}

bootstrap(MyApp)
```

This component is where you can put application-level code and configuration, and its template is where the whole app component chain gets created.

COMPONENT INIT

When a component is created, its constructor is called. This is where we initialize state for our component, but if we rely on properties or data from child components, we need to wait for our child components to initialize first.

To do this, we can handle the `ngOnInit` lifecycle event. Optionally, we could call `setTimeout` in our constructor for a similar effect:

```
import {Component, bootstrap} from '@angular/core';

// Annotation section
@Component({
  selector: 'street-map',
  template: '<map-window></map-window><map-controls></map-
})
// Component controller
class StreetMap {
  constructor() {
    this.name = 'Max';
  }

  setMapWindow(mapWindow) {
    this.mapWindow = mapWindow;
  }

  setMapControls(mapControls) {
    this.mapControls = mapControls;
  }

  ngOnInit() {
```

```
    // Properties are resolved and things like
    // this.mapWindow and this.mapControls
    // had a chance to resolve from the
    // two child components <map-window> and <map-controls
  }
}
```

COMPONENT LIFECYCLE

Like `ngOnInit`, we can track several events through the lifecycle of a component. For a full list, see the official Angular 2 Lifecyle Hooks docs.

```
// Annotation section
@Component({
  selector: 'street-map',
  template: '<map-window></map-window><map-controls></map-
})
// Component controller
class StreetMap {
  ngOnInit() {
    // Properties are resolved and things like
    // this.mapWindow and this.mapControls
```

```
    // had a chance to resolve from the
    // two child components <map-window> and <map-controls
  }
  ngOnDestroy() {
    // Speak now or forever hold your peace
  }
  ngDoCheck() {
    // Custom change detection
  }
  ngOnChanges(changes) {
    // Called right after our bindings have been checked b
    // if one of our bindings has changed.
    //
    // changes is an object of the format:
    // {
    //   'prop': PropertyUpdate
    // }
  }
  ngAfterContentInit() {
    // Component content has been initialized
  }
  ngAfterContentChecked() {
    // Component content has been Checked
  }
  ngAfterViewInit() {
    // Component views are initialized
```

```
  }
  ngAfterViewChecked() {
    // Component views have been checked
  }
}
```
