


Exercise 4 - Overview of Angular

Objective

To have a high-level understanding of how the various building blocks of Angular come together to create one cohesive framework.

Overview

You'll find a starter folder containing an 'assets' folder associated with this exercise, using command line tools we are going to initialise a new angular application which we will then tailor to our needs. The end result should look like the following image.

 transforming performance
through learning

COURSES INSTRUCTORS LOCATIONS CONTACT US

Courses available at QA

Building JavaScript Applications Using NodeJS and the MEAN Stack QANODEDEV Classroom 3 days	Developing MVC single-page web applications using AngularJS QAANGULARJS Classroom 2 days	Developing Web Applications Using HTML5 QAWEBUI Classroom 5 days
Leveraging the Power of JQuery QAJQUERY Classroom 3 days	Programming with JavaScript QAJAVSC Classroom 5 days	Next Generation JavaScript: ECMAScript2015 QAES2015 Classroom 2 days
Building Web Applications Using Angular QAANGULAR Classroom 3 days		

© 1996-2017 QA Limited

Instructions

Part 1 - Setting up the root component

1. Create a new angular application called "qa-app" within the starter folder.
2. Let's create an initial page for this application. In the 'assets' folder you'll find a template.html file - copy the **contents** of this file to the newly created app.component.html
3. Not much to look at right now. Our root module is bootstrapping our initial component to generate the view. We need some styles. Copy the contents of the /css/styles.css document into the root component's css file

4. Almost there. Except some of our styles haven't taken affect because they are intended for the HTML and Body elements which we can't style in a component's stylesheet. So, take those first two selectors out of aap.component.css which deal with body and html and put them instead into the styles.css file at the root of this app.
5. Finally copy the images folder from the starter assets folder, and place it within this applications assets folder.

Part 2 - Adding course details

Currently we have some placeholder text within our main element. Let's show some course details to our visitors.

6. Within the class definition for our `AppComponent` (app.component.ts) remove the `title` property and replace it with `courses`. Initialise this new `courses` property with the array of courses in the starter folder's /assets/courses.json
7. Replace the paragraph holding the 'lorem ipsum' text with the following:

```
<h1>Courses available at QA</h1>

<section class="qa-courses">
  <ul>
    <li>{{courses[0].title}}</li>
    <li>{{courses[0].code}}</li>
    <li>{{courses[0].delivery}}</li>
    <li>{{courses[0].days}} days</li>
  </ul>
</section>
```

The curly braces indicate that Angular should fetch this data from the component class.

8. We could repeat this block of code for each course in the array of courses. Fortunately, Angular provides a number of built-in directives, one of which can help us here: `ngFor`. Amend the `ul` element so it is as follows:

```
<ul *ngFor="let course of courses ">
  <li>{{course.title}}</li>
  <li>{{course.code}}</li>
  <li>{{course.delivery}}</li>
  <li>{{course.days}} days</li>
</ul>
```

9. As you can hopefully see in the browser, this Directive has enabled us to repeat our `ul` element for each entry in our array.