

2.1 Filters and a New Directive

Challenges

2.2 Using filters

350 PTS

In the previous challenge we practiced displaying the prices of gems in our `index.html`. The first gem, `Azurite`, now costs \$110.50 (due to supply shortages) but our current code doesn't display the price correctly.

Help Me

Task

Use an Angular filter to display the price as a currency.

```
<h3>
  {{product.name}}
  <em class="pull-right">{{product.price | currency }}</em>
</h3>
```

2.3 Displaying the First Image

200 PTS

We've added images to all products as well - like any self respecting store should have. A `product` has an array of `images` we can use in our HTML. Let's get the first image to show on the page!

Help Me

Task

We want to display the first full-sized image for each product. Can you figure out how?

```
</h3>
<!-- Image Gallery -->
<div class="gallery">
  
</div>
</div>
```

2.4 Display All Thumbnails

200 PTS

Our 2-dimensional clients have worked hard to produce multiple images for their `gems`. Display all the thumbnails inside our gallery without repeating yourself by using an AngularJS directive on the `li.thumbnail` element.

Help Me

Task 1/2

Make the `li`'s repeat for each `image` in the `image` array. Use the variable `image` as your identifier.

```
</div>
<ul class="img-thumbnails clear-fix">
  <li class="small-image pull-left thumbnail" ng-repeat="image in product.images">
```

Task 2/2

Now that we have `image` available from our `ng-repeat`, let's replace the `product.images[0]` with it.

```
1 <!DOCTYPE html>
2 <html ng-app="gemStore">
3   <head>
4     <link rel="stylesheet" type="text/css" href="bootstrap.min.css" />
5     <script type="text/javascript" src="angular.min.js"></script>
6     <script type="text/javascript" src="app.js"></script>
7   </head>
8   <body class="list-group" ng-controller="StoreController as store">
9     <!-- Product Container -->
10    <div class="list-group-item" ng-repeat="product in store.products">
11      <h3>{{product.name}} <em class="pull-right">{{product.price | currency}}</em></h3>
12
13      <!-- Image Gallery -->
14      <div class="gallery">
15        <div class="img-wrap">
16          
17        </div>
18        <ul class="img-thumbnails clear-fix">
19          <li class="small-image pull-left thumbnail" ng-repeat="image in product.images">
20            
21          </li>
22        </ul>
23      </div>
24    </div>
25  </body>
26 </html>
27
```

2.5 No Images, No Gallery

200 PTS

Many times a product will be ready to sell, but won't have images yet (like our first one). In this case the `images` array will just be empty. Note the change in our `gems` array.

We don't want to display the `.gallery` element if there are no images. Create an expression on our `.gallery` div to make this happen.

Help Me

Task

Use the `ng-show` directive to make sure our image gallery is only displayed if a product has images.

```
<!-- Image Gallery -->
<div class="gallery" ng-show="{{product.images > 0}}">
  <h4>{{product.description}}</h4>
  <img class="img img-circle img-thumbnail center-block
```

2.7 Tabs Inside Out

250 PTS

We just learned that cramming a bunch of logic into our HTML is kinda bad. Let's start the Right Way with a Controller which will control our tabs.

Help Me

Task 1/4

Create a controller named `TabController`.

```
app.controller("TabController", function() {
});
```

Task 2/4

An empty Controller isn't much use. Do you remember why we needed a Controller at all? That's right, we need to initialize the `tab` property. Go ahead and add that property now, setting it to `1`.

```
app.controller("TabController", function() {
  this.tab = 1;
});
```

Task 3/4

In order to set the current tab, we'll need a `setTab` method to use in our HTML. It should set the `tab` property of `TabController` to a value passed in.

```
app.controller("TabController", function() {
  this.tab = 1;
  this.setTab = function(setTab) {
    this.tab = setTab;
  };
});
```

Task 4/4

We've got a `setTab` method. Now we need an `isSet` method that accepts a value and returns whether that value matches `this.tab`.

```
1 (function() {
2   var app = angular.module('gemStore', []);
3
4   app.controller('StoreController', function() {
5     this.products = gems;
6   });
7
8   app.controller('TabController', function() {
9     this.tab = 1;
10
11     this.setTab = function(setTab) {
12       this.tab = setTab;
13     };
14     this.isSet = function(checkTab) {
15       return this.tab === checkTab;
16     };
17   });
18
19   var gems = [...];
20 })();
21
```

2.8 Using TabController

200 PTS

To connect our controller to our page, we're going to need a few directives. Try to figure out which ones to use, but if you get stuck we'll give you hints along the way. We've provided all the HTML for you. Try to accomplish the following tasks:

Help Me

Task 1/6

Attach the `TabController` to the `<section>` element with the `.tab` class. Don't forget to also add an alias.

```
<section ng-show="product.images.length"></section>

<section class="tab" ng-controller="TabController as tab">
  <ul class="nav nav-pills">
    <li>
```

Task 2/6

Trigger the `setTab` method when a link inside a `.tab` is clicked. Add this to each of the tab links. Pass in the number of the tab, in this case `1`, `2`, or `3`.

```
<section class="tab" ng-controller="TabController as tab">
  <ul class="nav nav-pills">
    <li>
      <a href ng-click="tab.setTab(1)">Description</a></li>
    <li>
      <a href ng-click="tab.setTab(2)">Specs</a></li>
    <li>
      <a href ng-click="tab.setTab(3)">Reviews</a></li>
    </ul>
    {{tab}}
  </div>
```

Task 3/6

Use the `isSet` method to show the corresponding `.tab`. You'll need to add a directive to each of the `<div>` elements in `.tab`.

```
2 <a href ng-click="tab.setTab(2)">
3 <li>
4 <a href ng-click="tab.setTab(3)">
5 </ul>
6 {{tab}}
7 <div ng-show="tab.isSet(1)">
8 <h4>Description</h4>
9 <blockquote></blockquote>
10 </div>
1 <div ng-show="tab.isSet(2)">
2 <h4>Specs</h4>
3 <blockquote>Shine: </blockquote>
4 </div>
5 <div ng-show="tab.isSet(3)">
6 <h4>Reviews</h4>
7 <blockquote></blockquote>
8 </div>
```

Task 4/6

Add the `active` class to the correct `li` when each `.tab` is clicked. You'll need to add a directive to each `` in `.nav`.

```
<section class="tab" ng-controller="TabController as tab">
  <ul class="nav nav-pills">
    <li ng-class="{'active':tab.isSet(1)}">
      <a href ng-click="tab.setTab(1)">Description</a></li>
    <li ng-class="{'active':tab.isSet(2)}">
      <a href ng-click="tab.setTab(2)">Specs</a></li>
    <li ng-class="{'active':tab.isSet(3)}">
      <a href ng-click="tab.setTab(3)">Reviews</a></li>
    </ul>
    {{tab}}
  <div ng-show="tab.isSet(1)">
    <h4>Description</h4>
```

Task 5/6

Within the Description tab, output the product's `description` within the `<blockquote>` element.

```
<div ng-show="tab.isSet(1)">
  <h4>Description</h4>
  <blockquote>{{product.description}}</blockquote>
</div>
```

Task 6/6

Within the Specs tab, output the product's `shine` value. It should look something like this within the `<blockquote>` element: "Shine: 4".

```
1 <!DOCTYPE html>
2 <html ng-app="gemStore">
3   <head>
4     <script src="angular.min.js"></script>
5     <script src="app.js"></script>
6   </head>
7   <body class="list-group" ng-controller="StoreController as store">
8     <header></header>
9
10    <div class="list-group-item" ng-repeat="product in store.products">
11      <h3>{{product.name}} <em class="pull-right">{{product.price | currency}}</em>
12
13      <div class="img-wrap">
14        
15      </div>
16
17      <div class="img-thumbnails">
18        <div class="img-wrap">
19          
20        </div>
21        <ul class="img-thumbnails">
22          <li class="small-image pull-left thumbnail" ng-repeat="image in product.images">
23            
24          </li>
25        </ul>
26      </div>
27
28      <section class="tab" ng-controller="TabController as tab">
29        <ul class="nav nav-pills">
30          <li ng-class="{'active':tab.isSet(1)}">
31            <a href ng-click="tab.setTab(1)">Description</a></li>
32          <li ng-class="{'active':tab.isSet(2)}">
33            <a href ng-click="tab.setTab(2)">Specs</a></li>
34          <li ng-class="{'active':tab.isSet(3)}">
35            <a href ng-click="tab.setTab(3)">Reviews</a></li>
36          </ul>
37          {{tab}}
38        <div ng-show="tab.isSet(1)">
39          <h4>Description</h4>
40          <blockquote>{{product.description}}</blockquote>
41        </div>
42        <div ng-show="tab.isSet(2)">
43          <h4>Specs</h4>
44          <blockquote>Shine: {{product.shine}}</blockquote>
45        </div>
46        <div ng-show="tab.isSet(3)">
47          <h4>Reviews</h4>
48          <blockquote></blockquote>
49        </div>
50      </section>
51    </div>
52  </body>
53 </html>
54
```

2.9 Creating Gallery Controller

250 PTS

Now that you're such a whiz with tabbed tab, can you apply what you've learned to our image gallery, too? Don't worry, we'll start slow... And there's a lot more in common than you might think!

Help Me

Task 1/3

Let's start with a Controller, just like last time. Call it `GalleryController`.

```
app.controller("GalleryController", function() {
});
```

Task 2/3

Initialize a `current` property of `GalleryController` with a value of `0`.

```
app.controller("GalleryController", function() {
  this.current = 0;
});
```

```
1 (function() {
2   var app = angular.module('gemStore', []);
3
4   app.controller('GalleryController', function() {
5     this.current = 0;
6
7     this.setCurrent = function(setCurrent) {
8       this.current = setCurrent || 0;
9     };
10  });
11
12   app.controller('StoreController', function() {
13     this.products = gems;
14   });
15
16   app.controller('TabController', function() {});
17
18   var gems = [...];
19 })();
```

2.10 Using Gallery Controller

200 PTS

Now that we've got an awesome `GalleryController`, let's wire it up to the page:

Help Me

Task 1/2

Attach `GalleryController` to the `.gallery` element that wraps our gallery; use the alias `gallery`.

```
<!-- Image Gallery -->
<div class="gallery" ng-show="product.images.length"
  ng-controller="GalleryController as gallery">
```

Task 2/2

Change the first `ng-src` directive to use the `current` property. Properly use `current` as the index instead of `0`.

```
1 <div class="gallery" ng-show="product.images.length"
2   ng-controller="GalleryController as gallery">
3   
4   <ul class="list-inline thumbs">
```