Mouse Hover Example

Overview

Following a test-first methodology, create an Angular application that shows a button and an image of a lightbulb that is off. When the user hovers the mouse pointer over the button the lightbulb changes to an image of a turned on lightbulb, and the image turns to an off lightbulb when the mouse pointer leaves the button.

Note: You may use Angular 1 or Angular 2 for this workshop.

Getting Set up

Getting Set up: You should scaffold a fresh project for this example that already sets up up for unit testing and end-to-end integration testing. We reccommend the yeoman generator gulp-angular for scaffolding and Angular 1 project and the Angular CLI for Angular 2.

You can find sample images for this project here: http://tinyurl.com/zehj8cb

Part 1. TDD / Unit Testing

Create this: an Angular controler with a boolean variable (referred to here as vm.litUp1) and an image whose src attrivute will change.

1You can really name this variable anything you like.

The controller should:

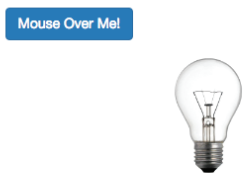
- begin with vm.litUp equal to false.
- begin with the lightbulb displaying the “off” image.
- set vm.litUp to be true when button is hovered over.
- change the lightbulb image src to “on” image when hovered over.
- change vm.litUp text to be false when button is hovered out of.
- change image src to be “off image” when button is hovered out of.

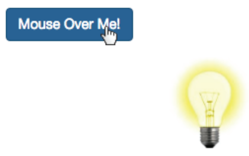
Part 2. E2e Protractor Testing

The page should:

- begin with the lightbulb showing the “off” image.
- change the lightbulb image src to “on” image when hovered over.
- change the lightbulb back to the off image when the cursor moves off of the button.

Page examples:





FizzBuzz Service

Overview

Following a test-first methodology, create an Angular service which exposes a public method named ‘fizzbuzz’ that takes a number input, processes it according to the fizzbuzz algorithm specifications, and returns a string.

Getting Set up

You should scaffold a fresh project for this example that already sets you up for unit testing and end-to-end integration testing. We reccommend the yeoman generator gulp-angular for scaffolding an Angular 1 project and the Angular CLI for Angular 2.

Part 1. TDD / Unit Testing

Create this: an Angular “service service” with a public method `fizzbuzz()`

The fizzbuzz method should:

- return something.
- return a string.
- return ‘`fizz`’ if the number passed in is divisible by 3.
- return ‘`buzz`’ if the number passed in is divisible by 5.
- return ‘`fizzbuzz`’ if the number passed in is divisible by 3 and 5.
- if the number is divisible by neither 3 nor 5, return the number as a string[1].
- return ‘`Whoops! Please pass a whole number into fizzbuzz!`’ if the number passed into fizzbuzz is not a full digit value with data type Number.
- be able to handle whole numbers in string data type[2].
- return ‘`0 the hero`’ when the number passed in is zero.
- support negative numbers according to the same rules as above.

Part 2. E2e Protractor Testing

Create this: an html file with a text input and a <p> tag bound to a controller variable `inputValue`. `inputValue` variable is found by taking the value of the text input and sending it through the `fizzbuzz` method you created.

The page should:

- change the text in the <p> tag when the text in the input is changed.
- change the text in the <p> tag to ‘`fizz`’ when the text in the input is `3`.
- change the text in the <p> tag to ‘`buzz`’ when the text in the input is `10`.
- change the text in the <p> tag to ‘`fizzbuzz`’ when the text in the input is `3000`.
- change the text in the <p> tag to ‘`Whoops! Please pass a whole number into fizzbuzz!`’ when the text in the input is ‘`bazoinga`’.
- change the text in the <p> tag to ‘`fizz`’ when the text in the input is `-6`.
- change the text in the <p> tag to ‘`0 the hero`’ when the text in the input is `0`.
- change the text in the <p> tag to ‘`fizzbuzz`’ when the text in the input is `-30`.
- change the text in the <p> tag to `11` when the text in the input is `11`.
- change the text in the <p> tag to ‘`Whoops! Please pass a whole number into fizzbuzz!`’ when the text has no input.
- when app first loads up, show nothing in the <p> tag until the text input is changed by the user.

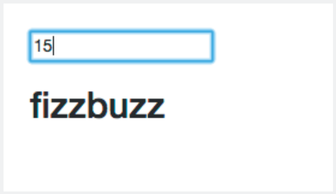
Note: You may use Angular 1 or Angular 2 for this workshop.

[1]E.g. `fizzbuzz(42)` should return the string ‘`42`’.

[2]E.g. the string ‘`45`’ passed into `fizzbuzz` should be processed as just the number 45.

Bonus Challenge UI & End-to-end Tests

Page example:



15

fizzbuzz