Learn Angular

BY MAKING A QUIZ APP FROM SCRATCH

HUNGRYTURTLECODE.COM

Part 5

Check Out The Whole Course Index

Angular Filters Create A Magic Search!

Ok, enough playing around, let’s really dig into the power of AngularJS. Creating search functionality from scratch can be notoriously hard, but in this tutorial we will see how easy it is to create an automatically updating search feature using Angular filters.

Using Angular filters in this way is definitely one of my favourite features of AngularJS. It makes seemingly difficult tasks so easy and straight forward to do.

If you want to take a look at this search functionality in action, it can be seen in the video below or check out the app here. As always this video is exactly the same as this article just to give you some options of how you want your information.

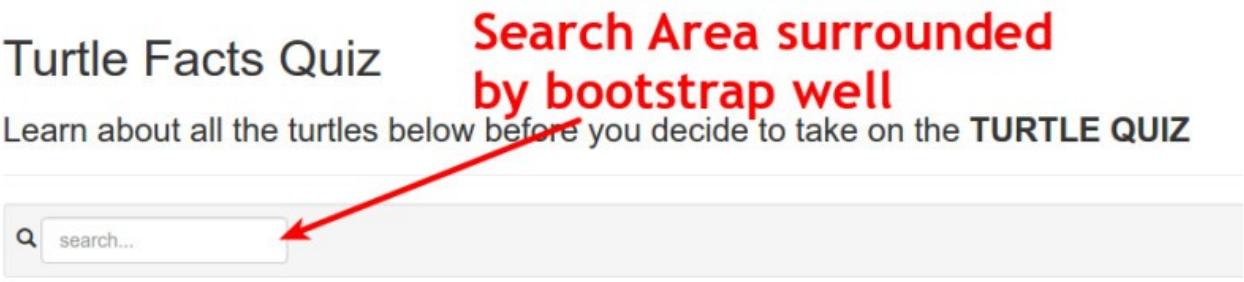
The git repo can be found here.

Video Killed The Radio Stars

Click on the image to go to the video on youtube or go here https://www.youtube.com/watch?v=zisGjJySdLA&index=5&list=PLqr0oBkln1FBmOjK24_B4y_VAA8736wPq

[The next part can be found here]

Getting straight into it, will start off by creating an HTML form which will style with some bootstrap classes. We will add this code right at the top of the HTML markup for our list controller, as we want the search form to be at the top of the page.



This area is the whole grey box at the top of the page that will contain the search box as well as the start quiz button that we

will create later. Right now we will focus on creating the search box by adding the icon and the text input area. The icon will be a glyphicon which comes bundled with Bootstrap these days.

```
<form class="form-inline well well-sm clearfix">
    <span class="glyphicon glyphicon-search"></span>
    <input
        type="text"
        placeholder="Search..."
        class="form-control">
</form>
```

The form control class we used on the input is another bootstrap class just to let bootstrap know what this input is doing.

Ng-Model Directive

Now that we have the markup done we can start creating the magic of the search functionality. To start this, we're going to introduce a new directive – ng-model.

What ng-model does it allows us to bind an input on our page directly to a property on the view model. Remember we used the alias "vm" in our controller for the properties that the view model has access to.

So now if we create a property inside the controller and attach it on to the view model we can use ng-model to bind that directly to our input. This means that if we programmatically change the property inside the controller it will automatically update the input and vice versa.

```
function ListController(){
    var vm = this;

    vm.data = turtlesData;
    vm.activeTurtle = {};

    // Adding the Search property to be used in the ng-model
    vm.search = "";



    vm.changeActiveTurtle = changeActiveTurtle;

    function changeActiveTurtle(index){
        vm.activeTurtle = index;
    }
}
```

Here we created a property called “search” which would just be a string that contains the text the user is searching for. Which of course, is set to an empty string initially. Now let’s add the ng-model directive on to our input and bind to the search property.

```
<form class="form-inline well well-sm clearfix">
    <span class="glyphicon glyphicon-search"></span>
    <input
        type="text"
        placeholder="Search..."
        class="form-control"
        ng-model="list.search">
</form>
```

Using The Search Property With Angular Filters

If you’re used to using Unix command line commands then you might be used to using the pipe symbol | to quite literally “pipe” the output of one command into the input of another. This is how Angular filters work.

We want to filter the output of the ng-repeat which is what constructs the entire list of turtles. So inside the quotes of our

ng-repeat where it says “turtle in list.data” we had the pipe symbol at the end and then add the filters we want.

There are many filters that are included with Angular and you can also make your own if you want to. But the filter that we’re concerned with for the search functionality is called “filter” – the filter filter.

We will also be taking a look at other filters like the number filter later on this course.

The way filters work in angular is you specify the name of the filter you want to use, then a colon, followed by any argument she want to give the filter. In this case the argument we want to give the filter filter is the text that the user is searching for.

```
<div class="col-sm-6" ng-repeat="turtle in list.data | filter:list.search">
```

How Is This Working?

This is a good illustration of the dynamic nature of AngularJS. I mentioned earlier the ng-repeat directive works something similar to a for loop in normal programming languages. But as you can see hear this is not quite right.

In a programming language, when a for loop is finished running that’s it. The ng-repeat and all other directives are constantly updating when you inputs are given.

This is us to use the filter in real time. When we type something into the input comma updates search property in our controller which is what is filtering the ng-repeat. Because this is now changed the ng-repeat will run again and repopulate the list

with only entries that satisfy the filter. In other words only with Turtle Data that contain the search term.

Moving On To Part 6

Our list of turtles is really taking shape now! In the next part we will add the final touches which will been allowed to start building the quiz controller.

So I'll see you over there in part 6.

Adrian