Angular.js Introduction Tutorial

This tutorial will walk you through creating an Angular.js application using Visual Studio. The application you will be building will be a simple commerce application. While the type of application has been re-created hundreds of times; I chose this type of application for a few reasons: (1) most people are familiar with the logic that is involved in this application and (2) we can touch many different aspects of angular so you can have a broad sense on how to use angular and how to approach different problems you may encounter.

# Note about the syntax

If you have gone through other tutorials, you will see a different syntax used than what I have used in this tutorial. The reason I chose to use this syntax is because I feel this syntax is easier to read and also it conforms to the JavaScript standards by ensuring that nothing is bleeding into the global scope. Feel free to use an alternate syntax if you feel more comfortable with the different syntax.

1. Open up index.html and add the following line to the body tag :



The ng-app attribute is used to tell Angular where the root element of the application begins. Note the name you give the application here is also the same name that you will give to the module that you will create shortly. This attribute is also where angular begins the “compile” or bootstrap process when you first run the page Angular looks for this tag and then compiles all of the other Angular tags it finds into a JavaScript Object that you will code against. This is also what will setup the $scope object for you that you will use in your controllers later.

1. Next, you will add the ng-view directive to the div with the Id of “view”.



The ng-view directive tells Angular where to load all of the templates that you will create and assign to your routes. So when we navigate to /products; the template that we will specify for this route will be loaded into this div.

1. Next, open up the app.js file found under the app folder. Once you have this file open you will need to add the module name to the following variable :



The name you specify here should be the same name that we used in step #1 where we defined the ng-app directive. This name is used to tell angular this the top-most module in our application. This module will also be used throughout the tutorial to attach different objects to. The reason why I made this variable global is so that we can ensure that it will always be available and not have to rely on typing the string every time we need to attach something to the module; as this can lead to typos and cause errors that aren’t always that clear right away what is wrong.

1. Now that we have given our module a name, we now need to tell Angular about any dependent child modules we may be using in the application. This step isn’t always needed depending on how you build your applications; however, I like to keep my applications separated by concerns so I prefer to break the pieces of the application into different modules. Also, if you ever want to use a third-party library with Angular (such as AngularUI which is the Angular-Bootstrap library that provides you with the JavaScript components of Bootstrap using Angular). You can think of this listing as your “using” statements that we use in C#.

Add the following module to the dependentModules array: (We will be continually adding new modules to this array as we progress through the tutorial.)



1. Next, we need to configure the routes that our application have. In the configuration function add the following lines:



What we are doing in this step is similar to what use in ASP.net MVC. We are telling angular that when it receives the URL “/products” that we need to load the template from the URL we provided and that use the controller we specified.

Now the last line of this file you will see that all of the objects we created above are now passed into angular to create and configure our objects with angular.



1. Next, we will create our products module. Open up the products.js file under the products folder and create a variable named “productsModule” just beneath the ‘use strict’ line of code.



1. In the dependentModules array, add the following :



1. Now add the following line to the products.js file; so that we can create our module and attach it to angular. If you remember back in step #4; we told angular that we had a dependent module called “products”; this is where we create this module.
2. Now that we have crated our child module “products”, we need to