AngularJS – Part 3

Agenda

- Use of some built in AngularJS services
- Creating a custom service
-

Tools

- Visual Studio 2013

Pre-requisite

- AngularJS – Part 2

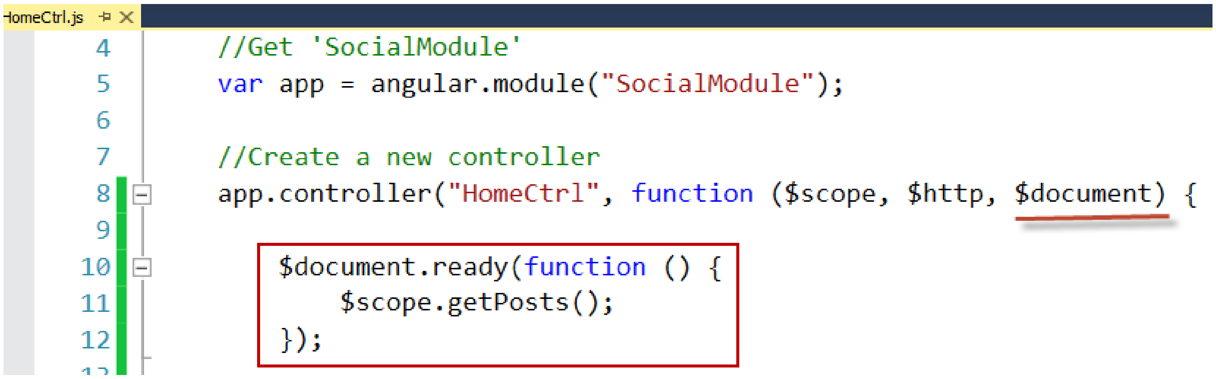
Version	**Last updated**	**Comments**	**Modified By**
V1.0	26-05-2016		Bilal Shahzad

Brief Introduction

1- Introduction to Ready function ($document service usage)
2- Introduction to $interval service.
3- How to create a service in angularJS

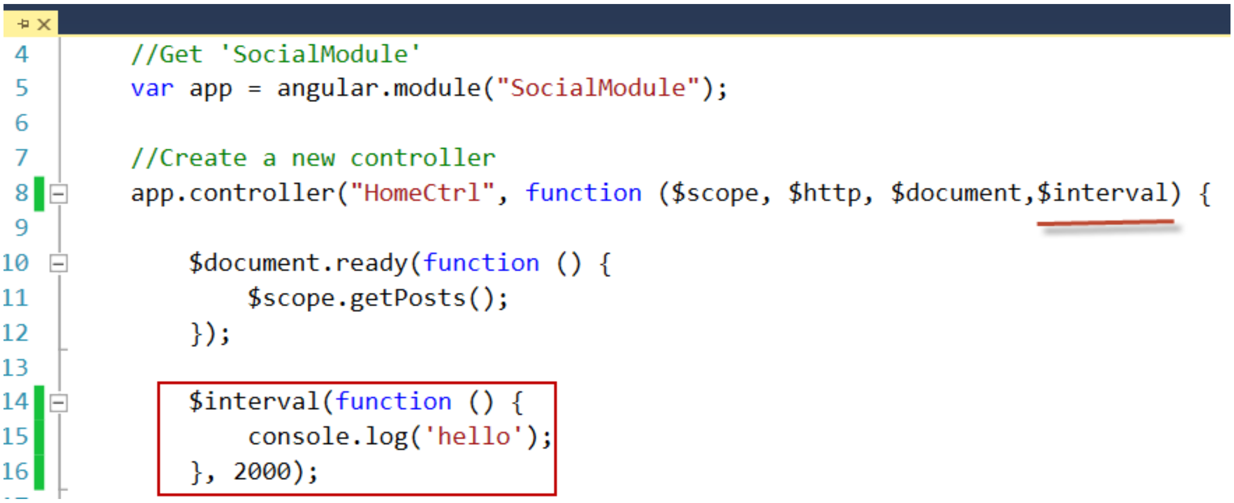
Step by Step Walkthrough

1- **Executing some code at Start:** Let’s say we want to do something (e.g. load posts) when page is loaded (or when DOM is ready) instead of clicking the button. You can inject “$document” object in your controller and then can use its ready function. Here we are calling our “getPosts()” function on ready. **Note:** here $document is a wrapper around our “document” object.



```
//Get 'SocialModule'
var app = angular.module("SocialModule");

//Create a new controller
app.controller("HomeCtrl", function ($scope, $http, $document) {

    $document.ready(function () {
        $scope.getPosts();
    });
```

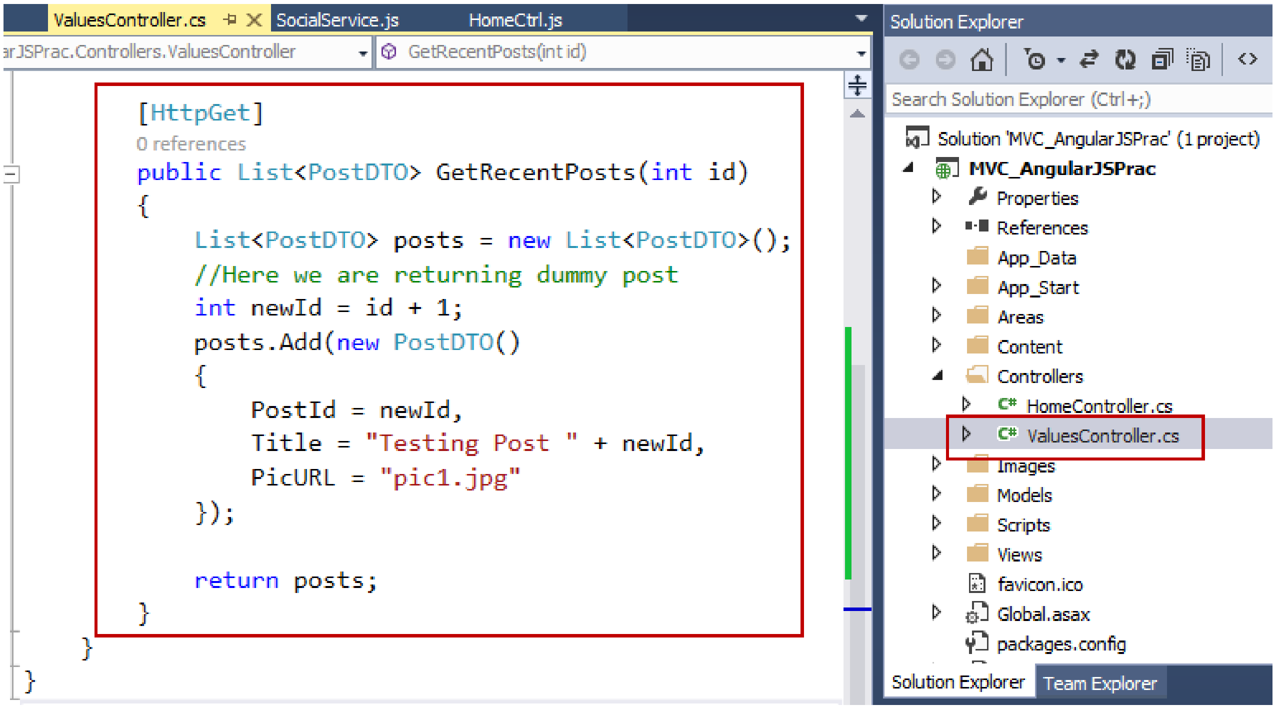
2- **Executing some code after specific interval:** We’ve two wrapper services ($timeout & $interval). And your guess is right. $timeout is wrapper around “setTimeout” function and $interval is wrapper around “setInterval” function. Here is example of how to use $interval in our controller. We’ll have to first inject $interval in controller function. Currently, $interval function is going to call a function (which prints hello) after every 2 seconds.



```
//Get 'SocialModule'
var app = angular.module("SocialModule");

//Create a new controller
app.controller("HomeCtrl", function ($scope, $http, $document,$interval) {

    $document.ready(function () {
        $scope.getPosts();
    });

    $interval(function () {
        console.log('hello');
    }, 2000);
```

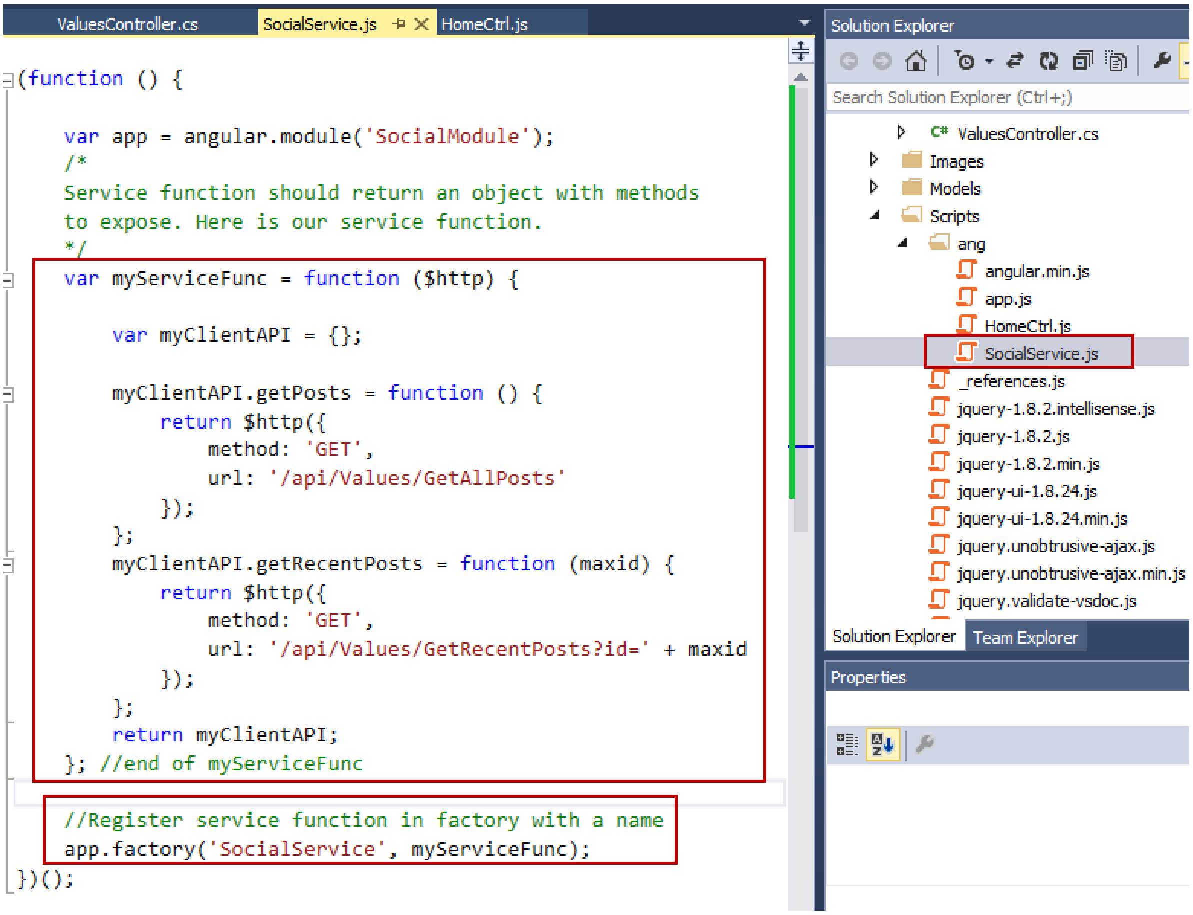
3- **Get Recent Posts**: Let’s see how to get recent posts and how to write services in AngularJS.

4- Create a function in your Web API controller (i.e. ValuesController). This function will get an ID and return the posts which are entered after this ID (i.e. where PostID > ID). So basic purpose of this function is to return recent posts.



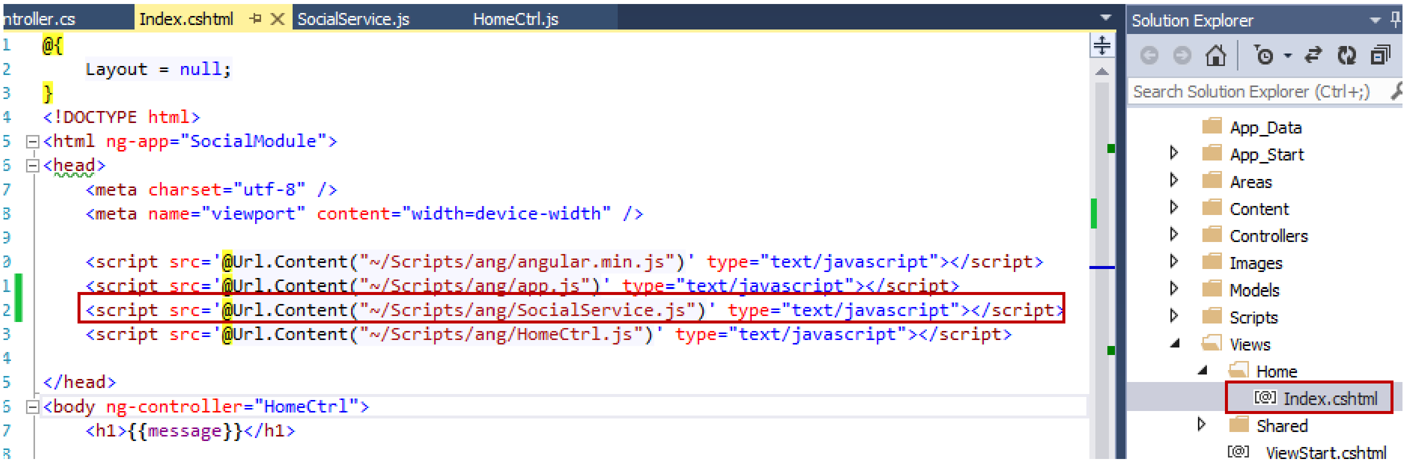
```
        [HttpGet]
        0 references
        public List<PostDTO> GetRecentPosts(int id)
        {
            List<PostDTO> posts = new List<PostDTO>();
            //Here we are returning dummy post
            int newId = id + 1;
            posts.Add(new PostDTO()
            {
                PostId = newId,
                Title = "Testing Post " + newId,
                PicURL = "pic1.jpg"
            });

            return posts;
        }
    }
}
```

1- **Creating Services**: Add a new JavaScript file (SocialService.js) in “Scripts/ang” folder and write following code as shown in the screenshot below. We want that service should be created instantly when file is loaded on client side therefore we are using same IIFE (Immediately-Invoked Function Expression) here. “(function(){})()” syntax executes the enclosed function immediately. In code, we are getting (already created) module object and storing it in “app” variable. Then we are creating a function (which depends on $http or going to use $http service). This function is our service function. So this function should return an object (which will have all its exposed functions). We can also consider “Service” as a static class (which has only one instance). In last lines, we are registering our service function with a name “SocialService” in factory. Also we’ve moved “getPosts()” AJAX logic in this function and have written a new function.



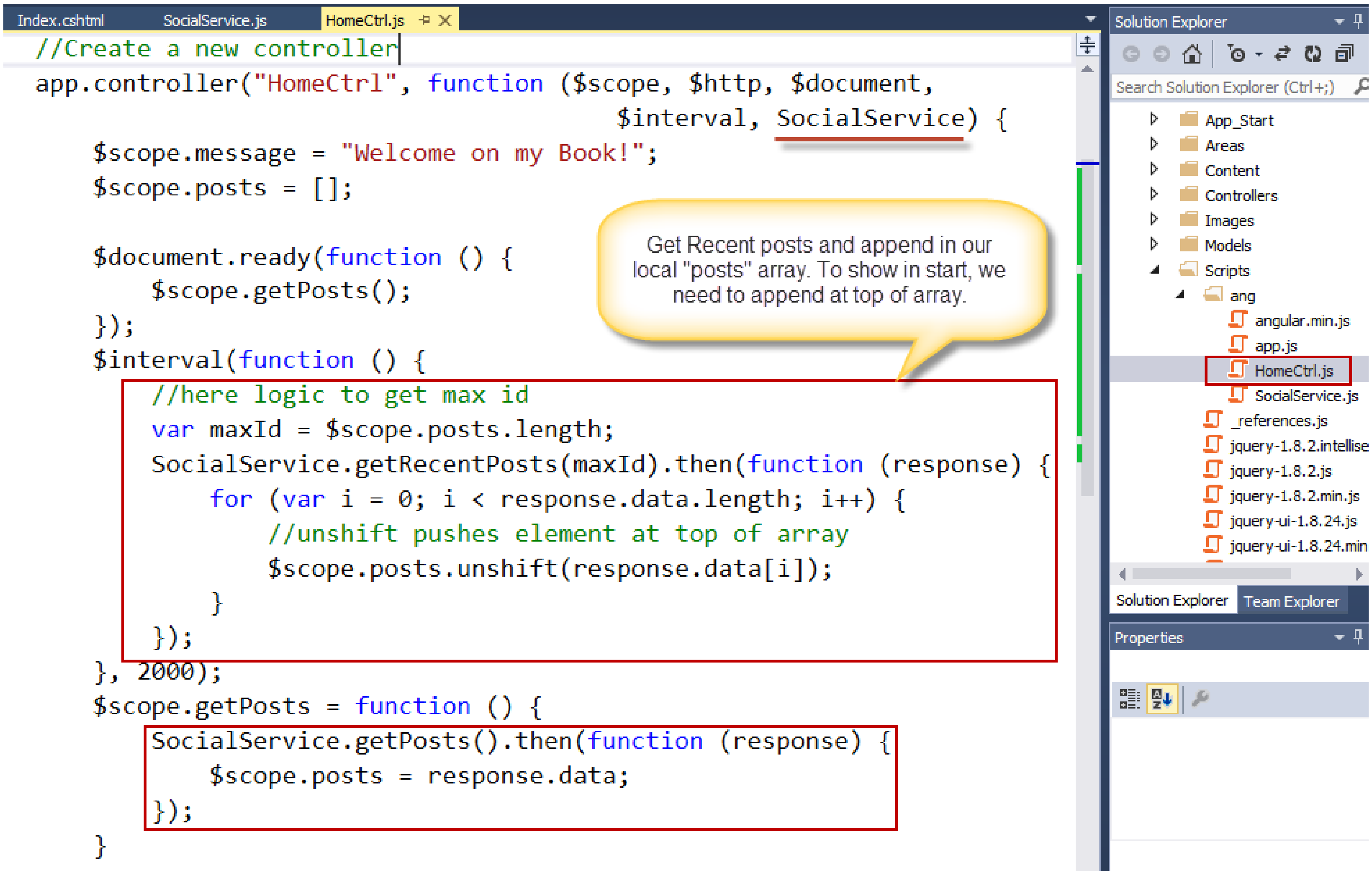
```
(function () {

    var app = angular.module('SocialModule');
    /*
    Service function should return an object with methods
    to expose. Here is our service function.
    */
    var myServiceFunc = function ($http) {

        var myClientAPI = {};

        myClientAPI.getPosts = function () {
            return $http({
                method: 'GET',
                url: '/api/Values/GetAllPosts'
            });
        };
        myClientAPI.getRecentPosts = function (maxid) {
            return $http({
                method: 'GET',
                url: '/api/Values/GetRecentPosts?id=' + maxid
            });
        };
        return myClientAPI;
    }; //end of myServiceFunc

    //Register service function in factory with a name
    app.factory('SocialService', myServiceFunc);
})();
```

2- Now add reference of this JS file in our “Views/Home/Index.cshtml” file. Note that we’ve added reference of our service file before controller as controller is going to use service.



```
@{
    Layout = null;
}
<!DOCTYPE html>
<html ng-app="SocialModule">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />

    <script src='@Url.Content("~/Scripts/ang/angular.min.js")' type="text/javascript"></script>
    <script src='@Url.Content("~/Scripts/ang/app.js")' type="text/javascript"></script>
    <script src='@Url.Content("~/Scripts/ang/SocialService.js")' type="text/javascript"></script>
    <script src='@Url.Content("~/Scripts/ang/HomeCtrl.js")' type="text/javascript"></script>

</head>
<body ng-controller="HomeCtrl">
    <h1>{{message}}</h1>
```

3- Now let’s make changes in our “HomeCtrl.js” file to use “Service” functions instead of writing AJAX logic inside controller. Note that we are injecting “SocialService” service in our controller as we want to use it.



```
//Create a new controller
app.controller("HomeCtrl", function ($scope, $http, $document,
                                     $interval, SocialService) {
    $scope.message = "Welcome on my Book!";
    $scope.posts = [];

    $document.ready(function () {
        $scope.getPosts();
    });
    $interval(function () {
        //here logic to get max id
        var maxId = $scope.posts.length;
        SocialService.getRecentPosts(maxId).then(function (response) {
            for (var i = 0; i < response.data.length; i++) {
                //unshift pushes element at top of array
                $scope.posts.unshift(response.data[i]);
            }
        });
    }, 2000);
    $scope.getPosts = function () {
        SocialService.getPosts().then(function (response) {
            $scope.posts = response.data;
        });
    }
```

Get Recent posts and append in our local "posts" array. To show in start, we need to append at top of array.

Tasks for Practice

Once you are done with above tutorial. Try to work on these tasks.

1- Explore $http
   a. https://docs.angularjs.org/api/ng/service/$http
2- Explore how to redirect in AngularJS

Useful Links

http://stackoverflow.com/questions/18646756/how-to-run-function-in-angular-controller-on-document-ready

https://docs.angularjs.org/api/ng/service/$http