Presented by: Sun Tam/Denzil Brown

League of ninja developers

New York, NY



Developer’s MVC cookbook

Various learnings

Version 1.0

December 22, 2016

# Developer’s MVC cookbook

## project setup

1. Create empty MVC project
2. Update all NuGet packages before starting any work in the solution
3. Add test project to the solution
4. Setup log4net
   1. <https://weblogs.asp.net/jhallal/configure-log4net-logging-framework-for-mvc>
   2. <http://stackoverflow.com/questions/36330299/how-to-configure-log4net-with-asp-net-mvc-c-sharp-in-visual-studio-2015>
   3. <https://code.msdn.microsoft.com/How-to-use-Apache-log4net-0d969339>
5. Separate concerns using the repository pattern or some other pattern before writing code in the controllers

## Angular Setup

1. Add NuGet Package: AngularJs.Core (1.5.9 on 12/22/2016)
2. Create an Angular bundle or similar approach to reference angular file either globally or per View using section tag.
   1. bundles.Add(new ScriptBundle("~/bundles/angular").Include(

                        "~/Scripts/angular\*"));

1. Add App folder in scripts folder: e.g: {ProjectName}\Scripts\App
   1. Create **app.module.js** in the App folder

(function () {

    'use strict';

    angular.module('app', []);

})();

1. Create a controller file eg: **flights.controller.js**: Ref file to be attached. (research naming conventions for js angular files)
2. Create a API Controller Folder under the Controllers folder then create a Web API Controller: Reference file to be attached.
   1. If there is an existing MVC controller serving a particular page modify to only return the view (absolutely no data retrieval or processing). All data request and processing will orchestrated by the angular js controller.
3. Ensure proper references to Angular bundle and js files in App folder for each View the uses Angular
   1. Example of View references

@section scripts

{

    @Scripts.Render("~/bundles/angular")

    <script src="~/Scripts/App/app.module.js"></script>

    <script src="~/Scripts/App/flightsController.js"></script>

}

1. For View pages
   1. Need an outer container (eg Div) with "ng-app" and "ng-controller" attributes to tell Angular to handle processing of data within eg
      1. <div ng-app="app" ng-controller="flightsController as vm">
   2. Script page using desired angular markup for rendering
2. ASP.NET MVC Security Considerations
   1. Note that when using API Controllers the [Authorize] attribute still works on API Actions but the redirect routing need to be specifically handled.
      1. To enable redirect with API controller comment out the following line in Startup.Auth file: {ProjectName}\App\_Start
         1. LoginPath = new PathString("/Account/Login")
      2. Add the following code in your angular controller to handle the 401 returned by Authorization failure:
         1. window.location.href = '/Account/Login?returnurl=/Flights/All';
      3. Also see reference here: <https://goo.gl/ljoiZ5>

# References

1. Mocking: <https://github.com/Moq/moq4/wiki/Quickstart>
2. Angular
   1. <https://www.asp.net/web-api/overview/getting-started-with-aspnet-web-api/build-a-single-page-application-spa-with-aspnet-web-api-and-angularjs>
   2. Miguel Castro: <https://www.youtube.com/watch?v=f67PFtrldGQ>
3. Attaching to IIS Express server
   1. <https://weblogs.asp.net/nmarun/attach-to-iisexpress-process-from-visual-studio>
4. Identity Manager
   1. <https://vimeo.com/125426951>
5. Dependency Injection
   1. For Web API 2: <https://www.asp.net/web-api/overview/advanced/dependency-injection>
   2. For Controllers: <https://app.pluralsight.com/library/courses/ioc-aspdotnet-mvc4/table-of-contents>