Make sure everything is configured on machine:

* Dotnet --version
* Md HelloWorld
* Cd HelloWorld
* Dotnet new
* Dotnet restore
* Dotnet run

Get Sources:

* Git clone <https://github.com/JJVertical/demos.git> jjvertical

Quick tour through CustomersAPI

* Open CustomersAPI.sln
* Wait for restore to complete
* Build solution
* Bring up TestExplorer window
* Run tests
* Checkout EF
* Checkout Startup.cs where we are injecting DBContext
* Checkout Startup.cs where we are also adding Swagger support
* Checkout UnitTests which also inject DBContext
* Checkout the Controllers
* Checkout Project.cs where are setting Kestrel port
* Launch API from command line (dotnet run)
* In browser navigate to <http://localhost:6000/swagger/ui>
* Add a customer
* Get customers
* Delete a customer

Quick tour through CustomersMVC

* Leave CustomersApi running
* Open CustomersMVC solution
* Wait for restore
* Build solution
* Checkout Service creation in Startup.cs
* Checkout dependency injection in controllers
* Checkout HttpContext class
* Run the application
* [http://localhost:<port>/home/addcustomer](http://localhost:%3cport%3e/home/addcustomer)
* [http://localhost:<port>/home/customerslist](http://localhost:%3cport%3e/home/customerslist)
* [http://localhost:<port>/home/deletecustomer](http://localhost:%3cport%3e/home/deletecustomer)

Add bootstrap

* Add new bower.json file
* Add bootstrap by typing "bootstrap" : in dependencies section and choosing the 3.3.7 version
* Expend the Solution Explorer dependency node
  + Right click to see the menu options
  + Open in file explorer to see docs

Add Grunt

* Add new packages.json file with the following content

{

"version": "1.0.0",

"name": "asp.net",

"private": true,

"devDependencies": {

"grunt": "\*",

"grunt-contrib-clean": "\*",

"grunt-contrib-watch": "\*",

"grunt-contrib-copy": "\*",

"grunt-contrib-jshint": "\*",

"grunt-contrib-uglify": "\*",

"grunt-contrib-cssmin": "\*"

}

}

* Add Grunt package
* Shoe dependency node
  + Open in file explorer to see the docs
* Add new Grunt.js file with the following content

/// <binding AfterBuild='jshint, copy' Clean='clean:wwwroot' />

/\*

This file in the main entry point for defining grunt tasks and using grunt plugins.

Click here to learn more. <http://go.microsoft.com/fwlink/?LinkID=513275&clcid=0x409>

\*/

module.exports = function (grunt) {

grunt.initConfig({

clean: {

wwwroot: ["wwwroot/\*"]

},

jshint: {

files: ['scripts/\*.js']

},

copy: {

bootstrap:

{

nonull: true,

expand: true,

cwd: 'bower\_components/',

src: ['bootstrap/dist/\*\*'],

dest: 'wwwroot/lib/'

},

jquery:

{

nonull: true,

expand: true,

cwd: 'bower\_components/',

src: ['jquery/dist/\*\*'],

dest: 'wwwroot/lib/'

}

},

});

grunt.loadNpmTasks("grunt-contrib-clean");

grunt.loadNpmTasks("grunt-contrib-jshint");

grunt.loadNpmTasks("grunt-contrib-copy");

};

* Open Task Runner Explorer (View-OtherWindows-Task Runner Explorer)
  + (Note: You may have to close and re-open your project)

Update the following content into \_Layout.cshtml

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>@ViewBag.Title</title>

<link href="~/lib/bootstrap/dist/css/bootstrap.css" rel="stylesheet" />

<script src="~/lib/jquery/dist/jquery.js"></script>

<script src="~/lib/bootstrap/dist/js/bootstrap.js"></script>

</head>

<body style="padding-top:50px; padding-bottom:20px">

<div class="navbar navbar-inverse navbar-fixed-top">

<div class="container">

<div class="navbar-collapse collapse">

<ul class="nav navbar-nav">

<li>@Html.ActionLink("Home", "Index", "Home", new { }, new { @class = "navbar-brand" })</li>

<li>@Html.ActionLink("Customers", "CustomersList", "Home", new { }, new { @class = "navbar-brand" })</li>

</ul>

</div>

</div>

</div>

<div class="container body-content">

<hr />

@RenderBody()

<hr />

</div>

</body>

</html>

Update customerList.cshtml

@using System.Collections.Generic;

@model IEnumerable<CustomerEntity>

<div id="Customers">

<div class="panel panel-info">

<div class="panel-heading">

<h2 class="panel-title">@Html.ActionLink("Add Customer", "AddCustomer", "Home")</h2>

</div>

<div class="panel-body">

<table class="table table-striped" style="white-space:nowrap">

<thead>

<tr>

<th>Id</th>

<th>First Name</th>

<th>Last Name</th>

<th>Phone Number</th>

<th>Address</th>

<th>City</th>

<th>State</th>

<th>Zip</th>

</tr>

</thead>

<tbody>

@foreach (var customer in Model)

{

<tr>

<td>@customer.Id</td>

<td>@customer.FirstName</td>

<td>@customer.LastName</td>

<td>@customer.PhoneNumber</td>

<td>@customer.Address</td>

<td>@customer.City</td>

<td>@customer.State</td>

<td>@customer.ZipCode</td>

<td>

@Html.ActionLink("Remove",

"DeleteCustomer",

"Home",

new { customerId = customer.Id },

new { @class = "btn btn-link" })

</td>

</tr>

}

</tbody>

</table>

</div>

</div>

</div>

Add ability to add customers

* Add the following two methods into your HomeController

public IActionResult AddCustomer()

{

return View();

}

[HttpPost]

public async Task<IActionResult> AddCustomer(CustomerEntity customer)

{

if (ModelState.IsValid)

{

await \_customersService.AddCustomerAsync(customer);

return RedirectToAction("CustomersList");

}

return View(customer);

}

* Add a new View under the views\Home folder with the following content

@model CustomerEntity

@{

ViewBag.Title = "Add Customer";

}

<form asp-controller="Home" asp-action="AddCustomer" method="post" class="form-horizontal" role="form">

<div class="form-horizontal">

<div asp-validation-summary="All" class="text-danger"></div>

<div class="form-group">

<label asp-for="Id" class="col-md-2 control-label"></label>

<div class="col-md-10">

<input asp-for="Id" class="form-control" />

<span asp-validation-for="Id" class="text-danger"></span>

</div>

</div>

<div class="form-group">

<label asp-for="FirstName" class="col-md-2 control-label"></label>

<div class="col-md-10">

<input asp-for="FirstName" class="form-control" />

<span asp-validation-for="FirstName" class="text-danger"></span>

</div>

</div>

<div class="form-group">

<label asp-for="LastName" class="col-md-2 control-label"></label>

<div class="col-md-10">

<input asp-for="LastName" class="form-control" />

<span asp-validation-for="LastName" class="text-danger"></span>

</div>

</div>

<div class="form-group">

<label asp-for="PhoneNumber" class="col-md-2 control-label"></label>

<div class="col-md-10">

<input asp-for="PhoneNumber" class="form-control" />

<span asp-validation-for="PhoneNumber" class="text-danger"></span>

</div>

</div>

<div class="form-group">

<label asp-for="Address" class="col-md-2 control-label"></label>

<div class="col-md-10">

<input asp-for="Address" class="form-control" />

<span asp-validation-for="Address" class="text-danger"></span>

</div>

</div>

<div class="form-group">

<label asp-for="City" class="col-md-2 control-label"></label>

<div class="col-md-10">

<input asp-for="City" class="form-control" />

<span asp-validation-for="City" class="text-danger"></span>

</div>

</div>

<div class="form-group">

<label asp-for="State" class="col-md-2 control-label"></label>

<div class="col-md-10">

<input asp-for="State" class="form-control" />

<span asp-validation-for="State" class="text-danger"></span>

</div>

</div>

<div class="form-group">

<label asp-for="ZipCode" class="col-md-2 control-label"></label>

<div class="col-md-10">

<input asp-for="ZipCode" class="form-control" />

<span asp-validation-for="ZipCode" class="text-danger"></span>

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="AddCustomer" class="btn btn-default" />

</div>

</div>

</div>

</form>

* Add following import into \_ViewImports

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers