

An Opinionated Approach to ASP.NET Core

(50 opinions, to be exact)

K. Scott Allen
@OdeToCode

Ode to Code

Solutions
and Projects

Startup

Controllers

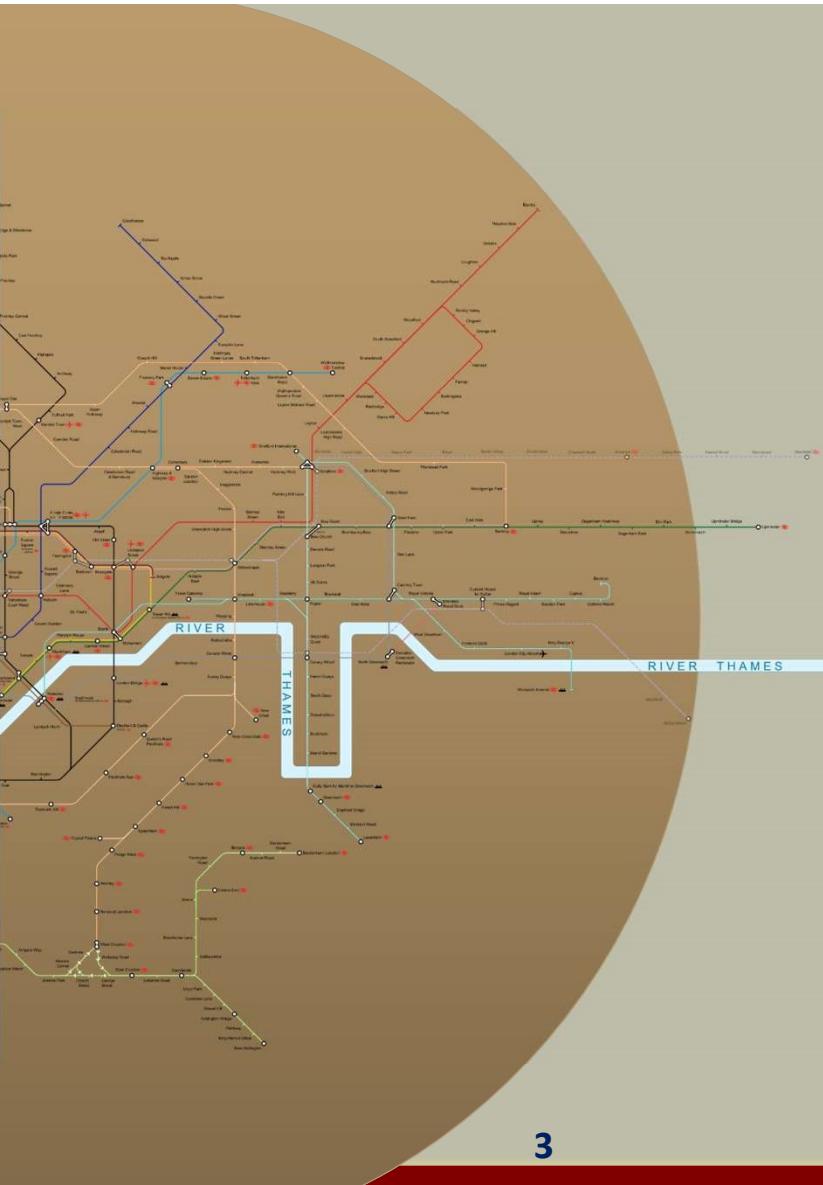
Client Code

C# Code

Views

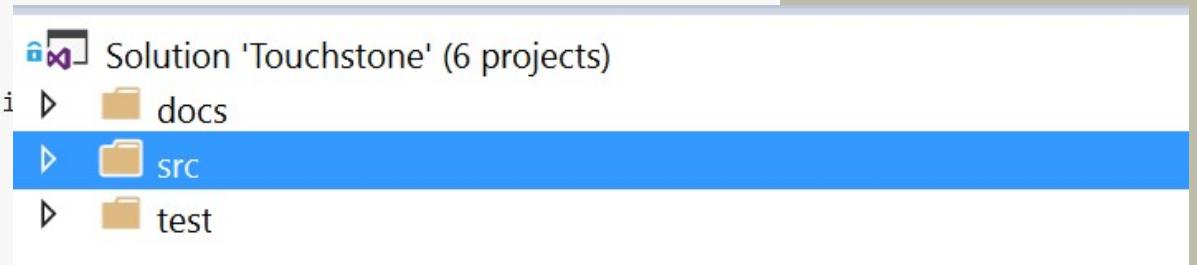
Tests

Solutions & Projects



Repository Structure

```
/Fruit.sln  
/src  
/src/Microsoft.AspNet.Banana  
/src/Microsoft.AspNet.Banana/project.json  
/src/Microsoft.AspNet.Banana/Banana.kproj  
/src/Microsoft.AspNet.Banana/Banana.cs  
/src/Microsoft.AspNet.Banana/Util/BananaUtil.cs  
/src/Microsoft.AspNet.Lychee  
/src/Microsoft.AspNet.Lychee/project.json  
/src/Microsoft.AspNet.Lychee/Lychee.kproj  
/src/Microsoft.AspNet.Lychee/Lychee.cs  
/src/Microsoft.AspNet.Lychee/Util/LycheeUtil.cs  
/test  
/test/Microsoft.AspNet.Banana.Tests  
/test/Microsoft.AspNet.Banana.Tests/project.json  
/test/Microsoft.AspNet.Banana.Tests/BananaTest.kproj  
/test/Microsoft.AspNet.Banana.Tests/BananaTest.cs  
/test/Microsoft.AspNet.Banana.Tests/Util/BananaUtilTest.cs
```



Samples & Tools

samples

src

test

tools

Branch: dev ▾ Security / samples / Create new file Upload files Find file History

| | | Latest commit 7ab28ec 27 days ago |
|---|---------------------------------|-----------------------------------|
|  pranavkm | Updating to 4.4 CoreFx packages | |
| .. | | |
|  CookieSample | Updating to 4.4 CoreFx packages | 18 days ago |
|  CookieSessionSample | Updating to 4.4 CoreFx packages | 18 days ago |
|  JwtBearerSample | Updating to 4.4 CoreFx packages | 18 days ago |
|  OpenIdConnect.AzureAdSample | Updating to 4.4 CoreFx packages | 18 days ago |
|  OpenIdConnectSample | Updating to 4.4 CoreFx packages | 18 days ago |
|  SocialSample | Updating to 4.4 CoreFx packages | 18 days ago |

Development Scripts

Solution 'BeaverLeague' (4 projects)

- scripts
 - addmigration.cmd
 - recreatedb.cmd
- Solution Items
 - .gitignore
 - global.json
- src
- test

```
addmigration.cmd  ✖  ×  recreatedb.cmd      Startup.cs
1 |pushd src\beaverleague.data
2 dotnet ef migrations add %1
3 dotnet ef database update
4 popd
```

Deployment Scripts

ps-aspnetmvc - Automation script
Resource group - PREVIEW

Download Add to library Deploy

Microsoft.Web/sites/config cannot be exported yet and is not included in the template. See error details.

Automate deploying resources with Azure Resource Manager templates in a single, coordinated operation. Define resources and config script or code. Learn more about template deployment.

Template Parameters CLI PowerShell .NET Ruby

Parameters (7) Resources (7)

```
2   "$schema":  
3     "https://schema.management.azure.com/schemas/2015-01-01/deployments.json#"  
4     "contentVersion": "1.0.0.0",  
5     "parameters": {  
6       "alertrules_Failure_Anomalies_ps_aspnetmvc_ai_name": {  
7         "defaultValue": "Failure Anomalies - ps-aspnetmvc",  
8         "type": "String"  
9       },  
10      "alertrules_High_CPU_name": {  
11        "defaultValue": "High CPU",  
12        "type": "String"  
13      },  
14      "alertrules_Server_Errors_name": {  
15        "defaultValue": "Server Errors",  
16        "type": "String"  
17      },  
18      "autoscalesettings_ps_aspnetmvc_ps_aspnetmvc_name": {  
19        "defaultValue": "ps-aspnetmvc-ps-aspnetmvc",  
20        "type": "String"  
21      },  
22      "components_ps_aspnetmvc_ai_name": {  
23        "defaultValue": "ps-aspnetmvc-ai",  
24        "type": "String"  
25      },  
26      "serverfarms_ps_aspnetmvc_name": {  
27        "defaultValue": "ps-aspnetmvc",  
28        "type": "String"  
29      },  
30      "sites_ps_aspnetmvc_name": {  
31        "defaultValue": "ps-aspnetmvc",  
32        "type": "String"  
33      },  
34      "variables": {},  
35      "resources": [
```

Announcing [Terraform Enterprise](#). Find out more >

TERRAFORM by HashiCorp



AppVeyor Builds

appveyor.yml build.ps1 README.md

```
1 version: 1.0.{build}
2 skip_non_tags: false
3 build_script:
4 - ps: ./build.ps1
5 test: off
6 branches:
7 - only:
8 - master
9 artifacts:
10 - path: .\artifacts\**\*.nupkg
11 name: NuGet
12 deploy:
13 - provider: NuGet
14 api_key:
15   secure: tg20msJwxYqlk0EwjZ1MuPXsX+ORK/7fxdeopd1
16 skip_symbols: true
17 on:
18 branch: master
19 appveyor_repo_tag: true
20 notifications:
21 - provider: Email
22 to:
23 - scott@odetocode.com
24 subject: UseNodeModules Build
25 on_build_success: true
26 on_build_failure: true
27 on_build_status_changed: false
```

appveyor.yml build.ps1 README.md

(script)

```
function Exec
{
    [CmdletBinding()] param(
        [Parameter(Position=0,Mandatory=1)][scriptblock]$cmd,
        [Parameter(Position=1,Mandatory=0)][string]$errorMessage = ($msgs.error_bad_command -f $cmd)
    )
    & $cmd
    if ($lastexitcode -ne 0) {
        throw ("Exec: " + $errorMessage)
    }
}

if(Test-Path .\artifacts) { Remove-Item .\artifacts -Force -Recurse }

exec { & dotnet restore }

exec { & dotnet build .\src\OdeToCode.AddFeatureFolders -c Release }

pushd .\test\OdeToCode.AddFeatureFolders.Tests
exec { & dotnet test -c Release }

exec { & dotnet pack .\src\OdeToCode.AddFeatureFolders -c Release -o .\artifacts }
```

VS Online Builds

Build Definitions / Build for SprintQA

Build Options Repository Variables Triggers General Retention History

Save Undo

+ Add build step...

- NuGet restore ***.sln
NuGet Installer
- .NET Core dotnet restore
.NET Core (PREVIEW)
- .NET Core dotnet build
.NET Core (PREVIEW)
- Build solution ***.sln
Visual Studio Build
- .NET Core dotnet test
.NET Core (PREVIEW)
- .NET Core dotnet publish
.NET Core (PREVIEW)
- Publish Web Artifacts
Publish Build Artifacts
- Copy and Publish DB
Copy and Publish Build Artifacts

Task catalog

All Build Utility Test Package Deploy

.NET Core (.NET Core (PREVIEW)) .NET Core (PREVIEW)
Build, test and publish using dotnet core command-line.

Android Build (deprecated; use Gradle) .NET Core (PREVIEW)
Build an Android app using Gradle and optionally start the emulator for unit tests

Android Signing .NET Core (PREVIEW)
Sign and align Android APK files

Ant .NET Core (PREVIEW)
Build with Apache Ant

CMake .NET Core (PREVIEW)
Build CMake projects

Cake Guillaume Rouchon .NET Core (PREVIEW)
Tools for building with Cake.
Download 296

Psake Guillaume Rouchon .NET Core (PREVIEW)
Tools for building with psake.
Download 57

Close

Ode to Code

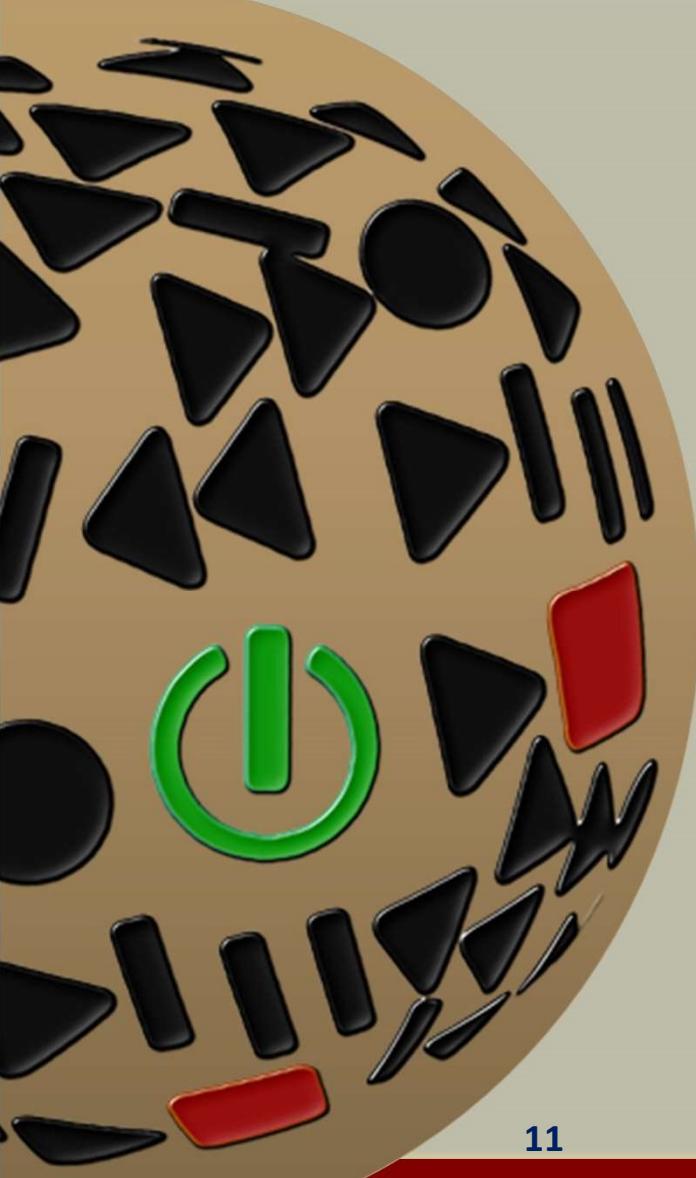
Every Class Is a Project Anti Pattern

The screenshot shows two windows side-by-side. On the left is the Microsoft Visual Studio 2015 Solution Explorer, displaying a large number of project items under the 'VisualStudio' category. On the right is a Windows Task Manager window showing the resource usage of the 'Microsoft Visual Studio 2015 (32 bit)' process.

| Name | CPU | Memory |
|---------------------------------------|-------|------------|
| Microsoft Visual Studio 2015 (32 bit) | 64.0% | 1,265.4 MB |

Task Manager details:

- Name: Microsoft Visual Studio 2015 (32 bit)
- CPU: 64.0%
- Memory: 1,265.4 MB
- Percentage: 77% CPU, 85% Memory



Startup

Startup.cs

- Three responsibilities

```
public Startup(IHostingEnvironment environment, ILoggerFactory loggerFactory){...}  
public void ConfigureServices(IServiceCollection services){...}  
public void Configure(IApplicationBuilder app){...}
```

Logger Is Configured In Constructor

- Leave Configure for middleware *only*

```
public Startup(IHostingEnvironment environment, ILoggerFactory loggerFactory)
{
    Configuration =
        new ConfigurationBuilder()
            .SetBasePath(environment.ContentRootPath)
            .AddJsonFile("appsettings.json")
            .AddJsonFile($"appsettings.{environment.EnvironmentName}.json", optional: true)
            .AddEnvironmentVariables()
            .Build();

    Log.Logger = new LoggerConfiguration()
        .MinimumLevel.Verbose()
        .WriteTo.RollingFile(new JsonFormatter(), "log-{Date}.json")
        .CreateLogger();
    loggerFactory.AddSerilog();
}
```

Keep It Clean

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddCustomizedMvc();
    services.AddSec    public static IServiceCollection AddCustomizedMvc(this IServiceCollection services)
    services.AddMec    {
        services.AddDat    var expander = new FeatureViewLocationExpander();

        services.AddMvc(options =>
    {
        options.Conventions.Add(new FeatureControllerModelConvention());
    })
    .AddRazorOptions(options =>
    {
        options.ViewLocationFormats.Clear();
        options.ViewLocationFormats.Add(@"{3}\{0}.cshtml");
        options.ViewLocationFormats.Add(@"Features\Shared\{0}.cshtml");
        options.ViewLocationExpanders.Add(expander);
    });
}

    return services;
}
```

Configure Is Allowed To Branch

```
if (env.IsDevelopment())
{
    app.UseDeveloperExceptionPage();
    app.UseDatabaseErrorHandler();
    app.UseBrowserLink();
}
else
{
    app.UseExceptionHandler("/Home/Error");
}
```

Make Use Of Your App As Executable

```
public static void Main(string[] args)
{
    var host = new WebHostBuilder()
        .UseKestrel()
        .UseContentRoot(Directory.GetCurrentDirectory())
        .UseIISIntegration()
        .UseStartup<Startup>()
        .Build();

    ProcessDbCommands(args, host);

    host.Run();
}
```

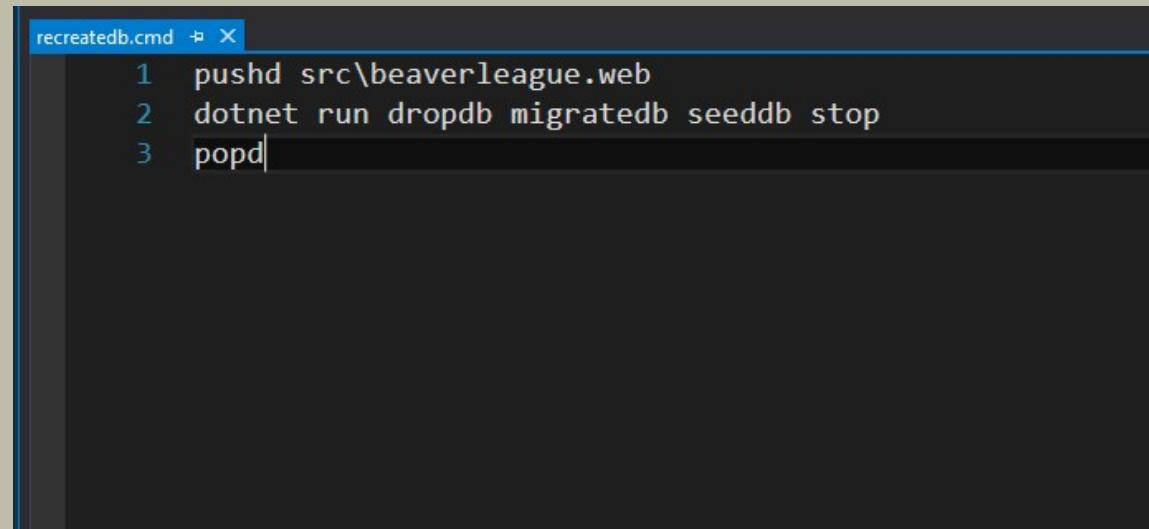
Always Use `IServiceScopeFactory` During Startup

```
private static void ProcessDbCommands(string[] args, IWebHost host)
{
    var services = (IServiceScopeFactory)host.Services.GetService(typeof(IServiceScopeFactory));

    using (var scope = services.CreateScope())
    {
        if (args.Contains("dropdb"))
        {
            Console.WriteLine("Dropping database");
            var db = GetLeagueDb(scope);
            db.Database.EnsureDeleted();
        }
        if (args.Contains("migratedb"))
        {
            Console.WriteLine("Migrating database");
            var db = GetLeagueDb(scope);
            db.Database.Migrate();
        }
        if (args.Contains("seeddb"))
        {
            Console.WriteLine("Seeding database");
            var db = GetLeagueDb(scope);
            db.Seed();
        }
    }

    if (args.Contains("stop"))
    {
        Console.WriteLine("Exiting on stop command");
        Environment.Exit(0);
    }
}
```

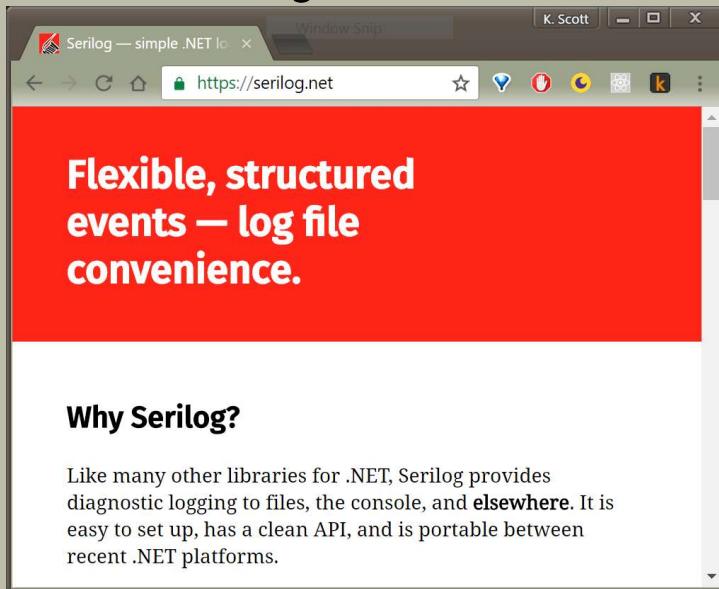
Add Scripts To Make Startup Parameters Easy



```
recreatedb.cmd ✎ ×
1 pushd src\beaverleague.web
2 dotnet run dropdb migratedb seeddb stop
3 popd
```

Logger Recommendation

- Use Serilog



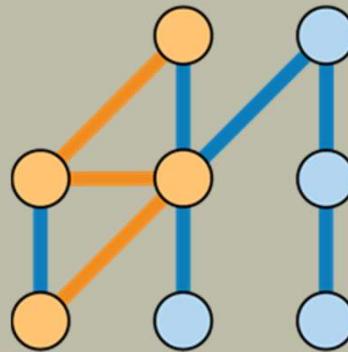
```
Log.Logger = new LoggerConfiguration()
    .MinimumLevel.Verbose()
    .WriteTo.RollingFile(new JsonFormatter(), "log-{Date}.json")
    .CreateLogger();
loggerFactory.AddSerilog();
```

```
var position = new { Latitude = 25, Longitude = 134 };
var elapsedMs = 34;

log.Information("Processed {@Position} in {Elapsed:000} ms.", position, elapsedMs);
```

Use a Full Featured Container

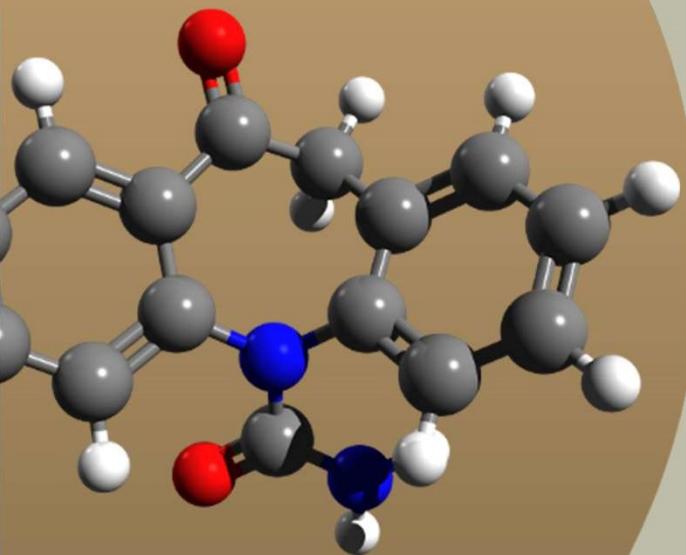
- ASP.NET Core already has a container
- Container doesn't support:
 - Auto-discovery
 - Decorators
 - Interception
 - Diagnostics



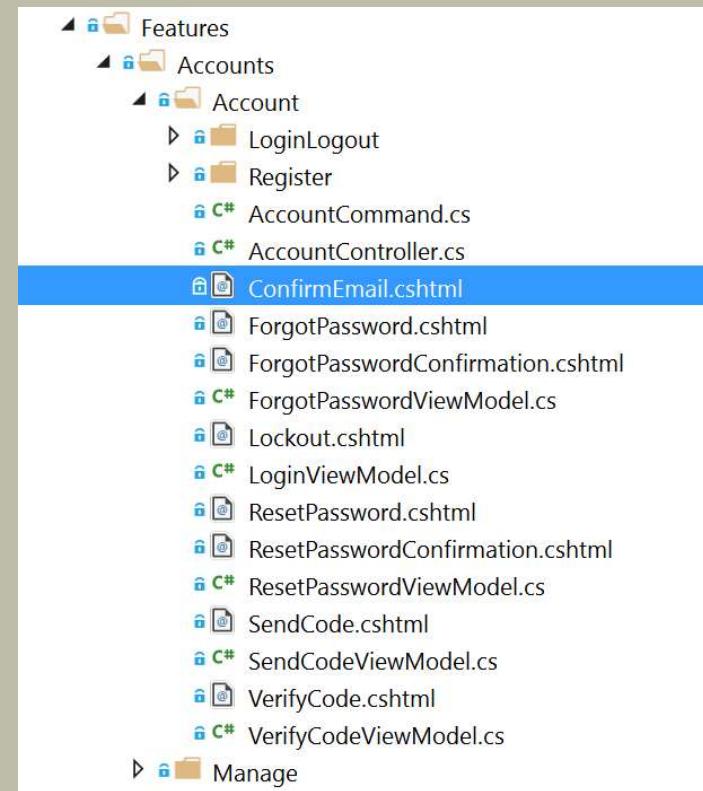
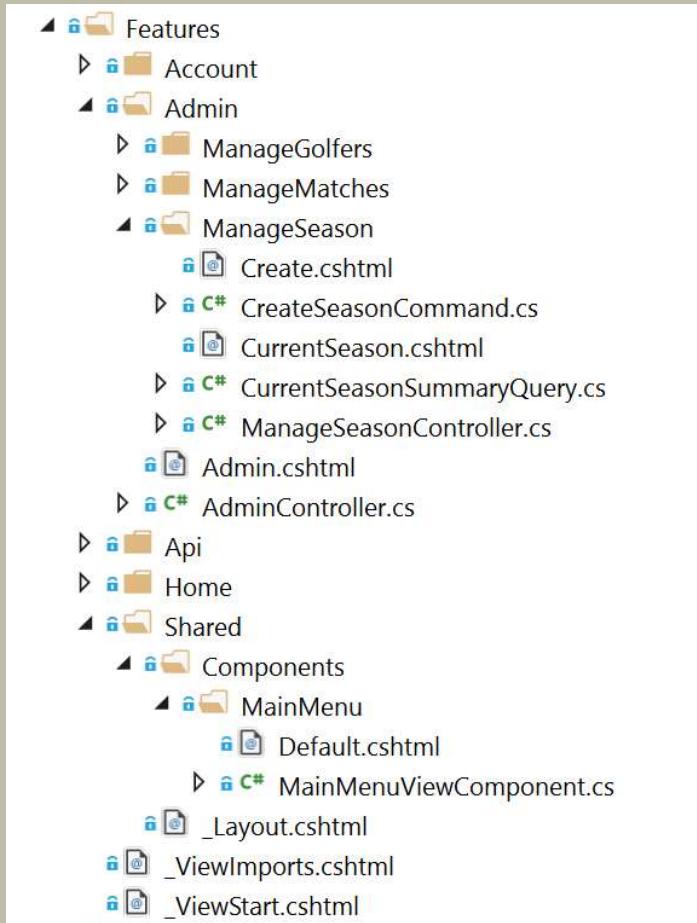
```
var container = new Container(_ =>
{
    // This usage adds the WidgetHolder as a decorator
    // on all IWidget registrations and makes AWidget
    // the default
    _ .For<IWidget>().DecorateAllWith<WidgetHolder>();
    _ .For<IWidget>().Use<AWidget>();
});

container.GetInstance<IWidget>()
    .ShouldBeOfType<WidgetHolder>()
    .Inner.ShouldBeOfType<AWidget>();
```

Project Structure



Feature Folders



Feature Folders Are Easy

Installation

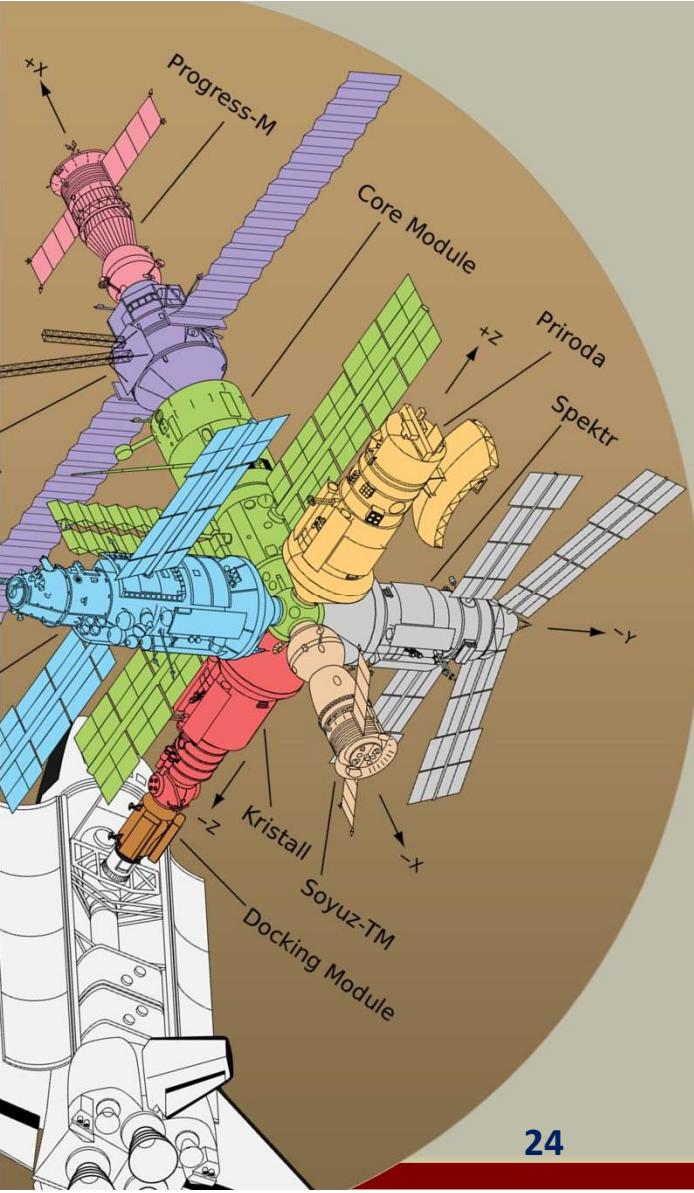
```
Install-Package OdeToCode.AddFeatureFolders
```

Usage

```
public class Startup
{
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddMvc()
            .AddFeatureFolders();

        // "Features" is the default feature folder root. To override, pass along
        // a new FeatureFolderOptions object with a different FeatureFolderName
    }

    public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory loggerFactory)
    {
        app.UseDeveloperExceptionPage();
        app.UseMvcWithDefaultRoute();
    }
}
```



Controllers

Route using [Route]

```
[Route("admin/[controller]/[action]")]
public class ManageMatchesController : Controller
{
    private readonly IMediator _mediator;

    public ManageMatchesController(IMediator mediator)
    {
        _mediator = mediator;
    }

    [Route("{seasonId}")]
    [HttpPost, ValidateAntiForgeryToken]
    public async Task<IActionResult> Create(int seasonId)
    {
        var command = new CreateMatchSetCommand {SeasonId = seasonId};
        var result = await _mediator.SendAsync(command);
        return RedirectToAction("Edit", new {matchSetId = result.MatchSet.Id });
    }

    [Route("{matchSetId}")]
    public IActionResult Edit(int matchSetId)
    {
        return View("Edit", new { matchSetId });
    }
}
```

Use Mediatr

```
[HttpPost, ValidateAntiForgeryToken]
public async Task<IActionResult> Login(LoginUserCommand command)
{
    if (ModelState.IsValid)
    {
        var result = await _mediatr.SendAsync(command);
        if (result.Success)
        {
            return Redirect("/");
        }
        ModelState.AddModelError(result.Errors);
    }
    return View(command);
}
```

Throw Away the Account Controller

```
UserController.java
package com.example.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;
import org.springframework.web.util.UriComponentsBuilder;
import com.example.model.User;
import com.example.repository.UserRepository;
import com.example.service.UserService;
import java.util.List;
import java.util.Optional;

@Controller
public class UserController {
    @Autowired
    private UserRepository userRepository;
    @Autowired
    private UserService userService;

    @GetMapping("/users")
    public ModelAndView index() {
        List<User> users = userRepository.findAll();
        return new ModelAndView("users/index", "users", users);
    }

    @GetMapping("/users/{id}")
    public ModelAndView show(@PathVariable Long id) {
        Optional<User> user = userRepository.findById(id);
        if (user.isPresent()) {
            return new ModelAndView("users/show", "user", user.get());
        } else {
            return new ModelAndView("error", "message", "User not found");
        }
    }

    @PostMapping("/users")
    public RedirectAttributes save(@ModelAttribute User user, RedirectAttributes redirectAttributes) {
        userService.create(user);
        redirectAttributes.addFlashAttribute("success", "User saved successfully");
        return redirectAttributes;
    }

    @PutMapping("/users/{id}")
    public RedirectAttributes update(@PathVariable Long id, @ModelAttribute User user, RedirectAttributes redirectAttributes) {
        userService.update(id, user);
        redirectAttributes.addFlashAttribute("success", "User updated successfully");
        return redirectAttributes;
    }

    @DeleteMapping("/users/{id}")
    public RedirectAttributes delete(@PathVariable Long id, RedirectAttributes redirectAttributes) {
        userService.delete(id);
        redirectAttributes.addFlashAttribute("success", "User deleted successfully");
        return redirectAttributes;
    }

    @GetMapping("/users/login")
    public ModelAndView login() {
        return new ModelAndView("users/login");
    }

    @PostMapping("/users/login")
    public RedirectAttributes login(@ModelAttribute User user, RedirectAttributes redirectAttributes) {
        userService.login(user);
        redirectAttributes.addFlashAttribute("success", "Login successful");
        return redirectAttributes;
    }

    @GetMapping("/users/logout")
    public RedirectAttributes logout(RedirectAttributes redirectAttributes) {
        userService.logout();
        redirectAttributes.addFlashAttribute("success", "Logout successful");
        return redirectAttributes;
    }

    @GetMapping("/users/register")
    public ModelAndView register() {
        return new ModelAndView("users/register");
    }

    @PostMapping("/users/register")
    public RedirectAttributes register(@ModelAttribute User user, RedirectAttributes redirectAttributes) {
        userService.register(user);
        redirectAttributes.addFlashAttribute("success", "Registration successful");
        return redirectAttributes;
    }

    @GetMapping("/users/confirm/{token}")
    public ModelAndView confirm(@PathVariable String token) {
        userService.confirm(token);
        return new ModelAndView("users/confirm");
    }

    @GetMapping("/users/forgotpassword")
    public ModelAndView forgotpassword() {
        return new ModelAndView("users/forgotpassword");
    }

    @PostMapping("/users/forgotpassword")
    public RedirectAttributes forgotpassword(@ModelAttribute User user, RedirectAttributes redirectAttributes) {
        userService.forgotpassword(user);
        redirectAttributes.addFlashAttribute("success", "Forgot password email sent");
        return redirectAttributes;
    }

    @GetMapping("/users/resetpassword/{token}")
    public ModelAndView resetpassword(@PathVariable String token) {
        return new ModelAndView("users/resetpassword");
    }

    @PostMapping("/users/resetpassword")
    public RedirectAttributes resetpassword(@ModelAttribute User user, RedirectAttributes redirectAttributes) {
        userService.resetpassword(user);
        redirectAttributes.addFlashAttribute("success", "Reset password successful");
        return redirectAttributes;
    }
}

UserManager.java
package com.example.service;

import com.example.model.User;
import com.example.repository.UserRepository;
import com.example.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;
import java.util.Optional;

@Service
public class UserManager {
    @Autowired
    private UserRepository userRepository;
    @Autowired
    private UserService userService;

    public List<User> getAllUsers() {
        return userRepository.findAll();
    }

    public User getUserById(Long id) {
        Optional<User> user = userRepository.findById(id);
        if (user.isPresent()) {
            return user.get();
        } else {
            throw new RuntimeException("User not found");
        }
    }

    public User createUser(User user) {
        return userService.create(user);
    }

    public User updateUser(Long id, User user) {
        return userService.update(id, user);
    }

    public void deleteUser(Long id) {
        userService.delete(id);
    }

    public void login(User user) {
        userService.login(user);
    }

    public void logout() {
        userService.logout();
    }

    public void register(User user) {
        userService.register(user);
    }

    public void confirm(String token) {
        userService.confirm(token);
    }

    public void forgotpassword(User user) {
        userService.forgotpassword(user);
    }

    public void resetpassword(User user) {
        userService.resetpassword(user);
    }
}
```

Don't Forget XSRF in APIs

```
[Route("api/[controller]")]
public class XsrfTokenController : Controller
{
    private readonly IAntiforgery _antiforgery;

    public XsrfTokenController(IAntiforgery antiforgery)
    {
        _antiforgery = antiforgery;
    }

    [HttpGet]
    public IActionResult Get()
    {
        var tokens = _antiforgery.GetAndStoreTokens(HttpContext);

        return new ObjectResult(new {
            token = tokens.RequestToken,
            tokenName = tokens.HeaderName
        });
    }
}
```

Embrace Security Policies

```
public class AuthorizationPolicies
{
    private AuthorizationOptions _options;

    public AuthorizationPolicies(AuthorizationOptions options)
    {
        _options = options;
    }

    public void Apply()
    {
        _options.AddPolicy(nameof(IsAdmin), IsAdmin());
    }

    AuthorizationPolicy IsAdmin()
    {
        return NewPolicyBuilder()
            .RequireClaim("isAdmin", "true")
            .Build();
    }

    AuthorizationPolicyBuilder NewPolicyBuilder()
    {
        var scheme = SignInManager.SchemeName;
        return new AuthorizationPolicyBuilder(scheme);
    }
}
```

Embrace Claims

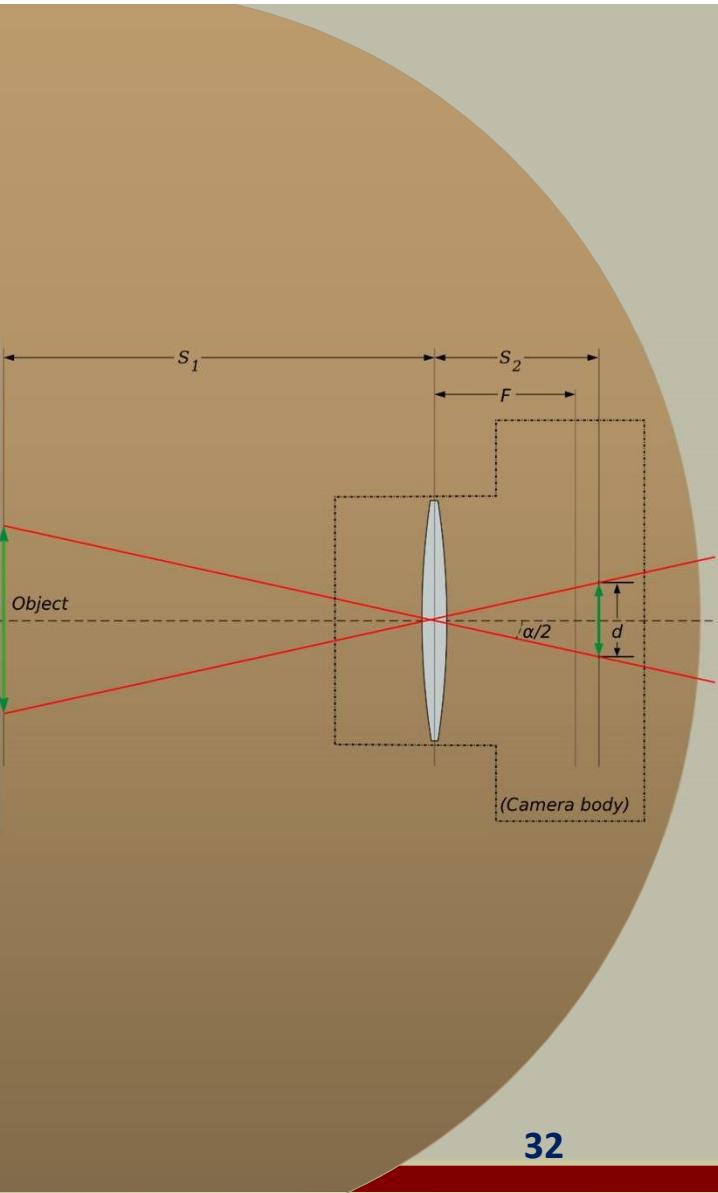
```
private ClaimsPrincipal CreateIdentity(Golfer golfer)
{
    var principal = new ClaimsPrincipal();
    var identity = new ClaimsIdentity(SchemeName);
    identity.AddClaim(new Claim(ClaimTypes.Email, golfer.EmailAddress));
    identity.AddClaim(new Claim(identity.NameClaimType, golfer.FirstName));
    if (golfer.IsAdmin)
    {
        identity.AddClaim(new Claim("isAdmin", "true"));
    }
    principal.AddIdentity(identity);
    return principal;
}
```

Create Custom Authorize Attributes

```
public class UserIsAdminAttribute : AuthorizeAttribute
{
    public UserIsAdminAttribute() : base("isAdmin")
    {

    }
}
```

Views



Avoid HTML Helpers

```
<div id="contact_form">
    @using (Html.BeginForm("SaveProfile", "Account"))
    {
        <fieldset>
            @Html.TextBoxFor(model => model.FullName, new { @class = "form_input_field" })<br />
            @Html.TextBoxFor(model => model.EMail, new { @class = "form_input_field" })<br />
            <br />
            <input type="submit" id="contact-submit" value="@Resources.GlobalResources.Save" />
        </fieldset>
    }
</div>
```

Embrace & Extend Tag Helpers

```
form method="post" asp-antiforgery="true">
    <div asp-validation-summary="ModelOnly"></div>
    <div class="form-group">
        <label asp-for="Name"></label>
        <input asp-for="Name" />
        <span asp-validation-for="Name" class="text-dan...
    </div>
    <div class="form-group">
        <input type="submit" value="Create" class="btn...
    </div>
</form>
```

```
[HtmlTargetElement("my-environment")]
public class MyEnvironment : TagHelper
{
    private readonly IHostingEnvironment _env;

    public MyEnvironment(IHostingEnvironment env)
    {
        _env = env;
    }

    public string Names { get; set; }

    public override void Process(TagHelperContext context, TagHelperOutput output)
    {
        if (Names.Split().Any(n => _env.EnvironmentName == n))
        {
            return;
        }

        output.SuppressOutput();
    }
}
```

Embrace View Components

```
public class MainMenuViewComponent : ViewComponent
{
    public IViewComponentResult Invoke()
    {
        return View();
    }
    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
        <ul class="nav navbar-nav">
            @if (User.IsAdmin())
            {
                <li><a asp-controller="Admin" asp-action="Admin">Admin</a>
            }
        </ul>
        <ul class="nav navbar-nav navbar-right">
            @if (User.Identity.IsAuthenticated)
            {
                <li class="navbar-text">Hi, @User.Identity.Name!</li>
                <li class="navbar-text">
                    <form asp-controller="Account" asp-action="Logout" asp-antiforgery="true">
                        <input class="btn btn-default btn-xs" type="submit" value="Logout"/>
                    </form>
                </li>
            }
            else
            {
                <li><a asp-controller="Account" asp-action="Login">Login</a></li>
            }
        </ul>
    </div>
```

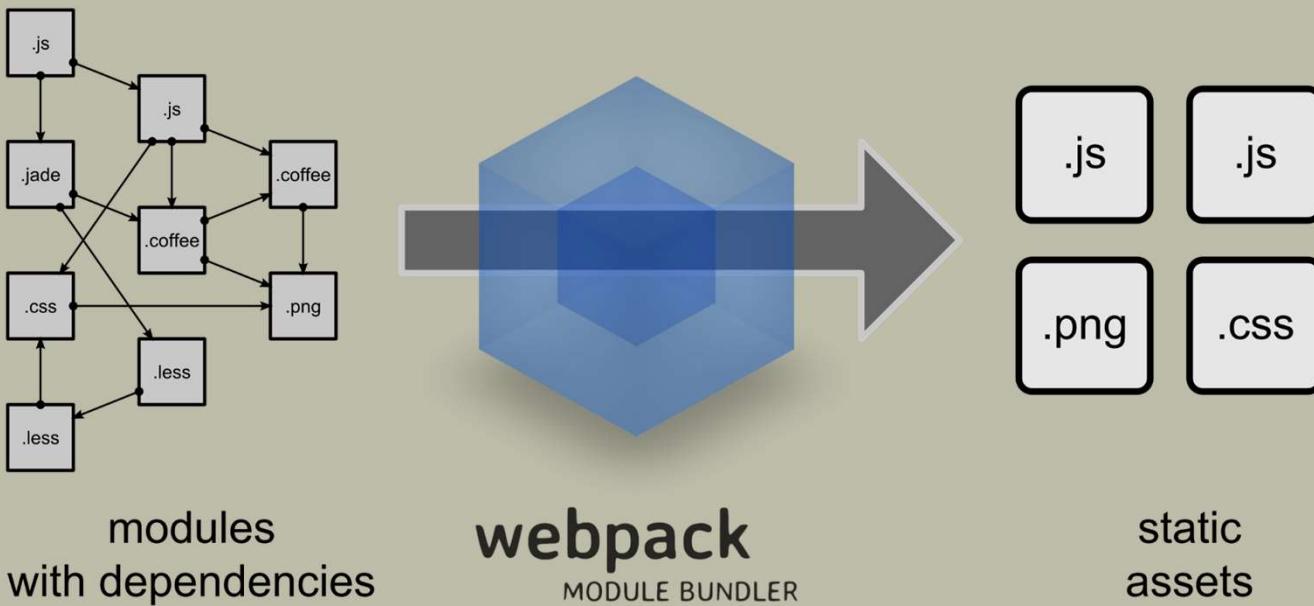
```
@await Component.InvokeAsync("MainMenu")
```

```
<vc:main-menu></vc:main-menu>
```



Client Code

Use WebPack



Use WebPack

```
const webpack = require("webpack");
const path = require("path");
const assets = path.join(__dirname, "wwwroot", "assets");

module.exports = {
    resolve: {
        extensions: ['', '.js']
    },
    entry: {
        vendor: [
            "react",
            "react-dom",
            "react-bootstrap",
            "axios",
            "es6-promise"
        ]
    },
    output: {
        path: assets,
        filename: "[name].js",
        library: "[name]_dll"
    },
    plugins: [
        new webpack.DllPlugin({
            path: path.join(assets, "[name]-manifest.json"),
            name: '[name]_dll'
        }),
        new webpack.optimize.UglifyJsPlugin({ compress: { warnings: false } })
    ]
};
```

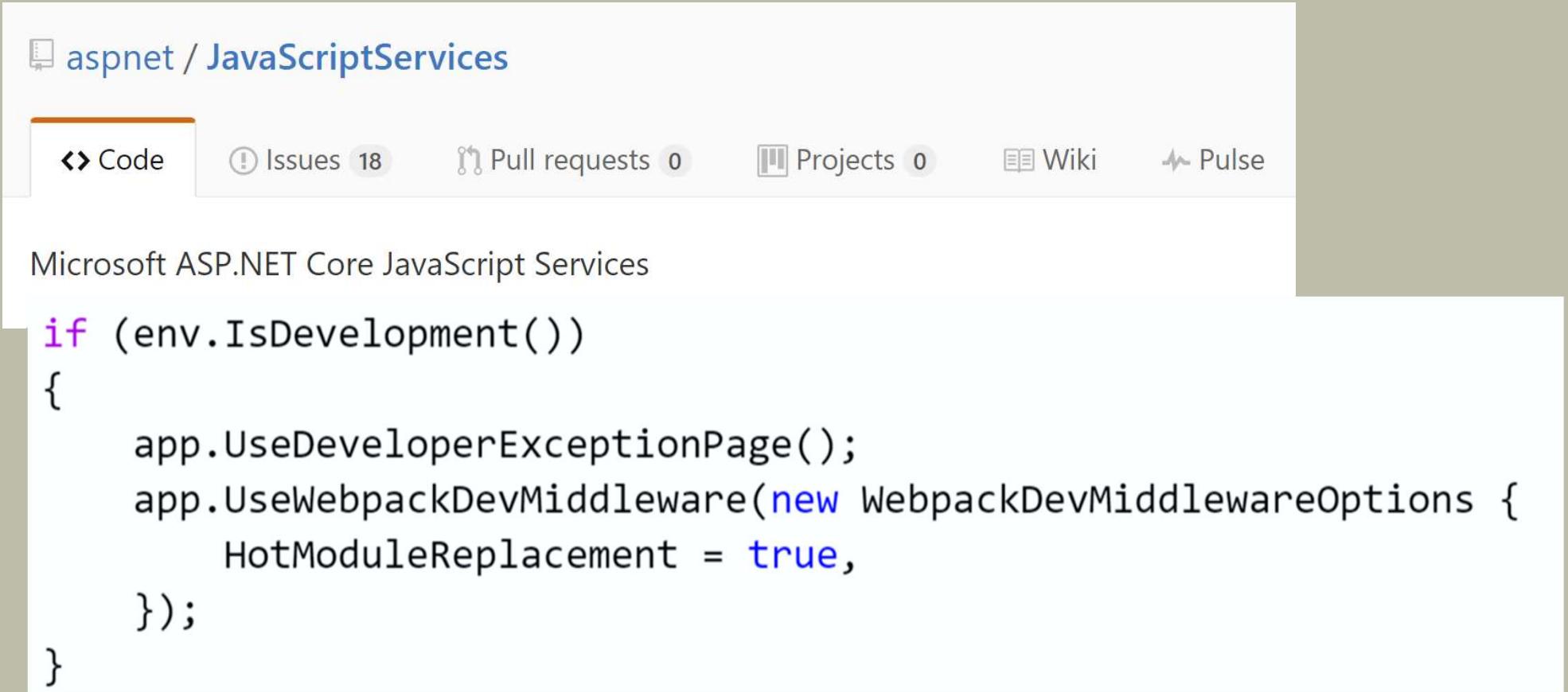
Use Multiple Configuration Files

```
const entries = {};
let files = [];
files = files.concat(glob.sync("./Features/**/*.specs.ts"));
files = files.concat(glob.sync("./Client/script/**/*.specs.ts"));

files.forEach(file => {
  var name = file.match("./(?:Features|Client\\script)(.+\\/[\\^\\/]+)\\.tsx?")[1];
  entries[name] = file;
});

module.exports = {
  resolve: [...],
  entry: entries,
  output: [...],
  devtool: "source-map",
  module: {
    loaders: [
      { test: /\.tsx?$/, loader: '_ts-loader' }
    ]
  },
  plugins: [...]
};
```

Use JavaScriptServices



The screenshot shows the GitHub repository page for "aspnet / JavaScriptServices". The repository name is displayed at the top left. Below it is a navigation bar with links: "Code" (highlighted in orange), "Issues 18", "Pull requests 0", "Projects 0", "Wiki", and "Pulse". The main content area displays the repository's description: "Microsoft ASP.NET Core JavaScript Services". Below the description is a code snippet in C#:

```
if (env.IsDevelopment())
{
    app.UseDeveloperExceptionPage();
    app.UseWebpackDevMiddleware(new WebpackDevMiddlewareOptions {
        HotModuleReplacement = true,
    });
}
```

Use Karma as a Test Runner



```
module.exports = function(config) {
  config.set({
    basePath: "",
    frameworks: ["jasmine"],
    files: [
      "./wwwroot/assets/vendor.js",
      "./wwwroot/assets/tests/**/*.*.specs.js"
    ],
    reporters: ["progress"],
    browsers: ["Chrome"]
  });
}
```

Use ConEmu or cmder

The screenshot shows three windows in the ConEmu interface:

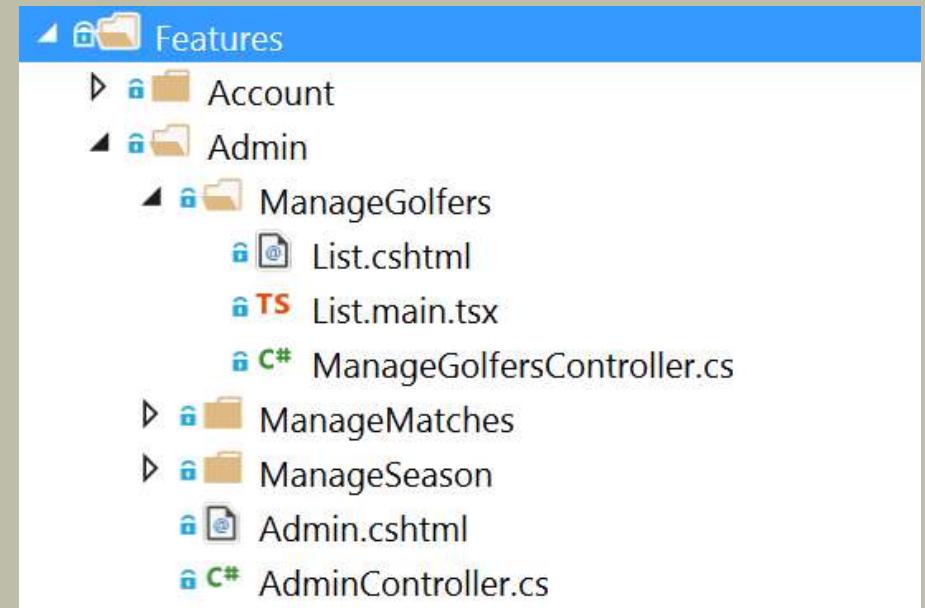
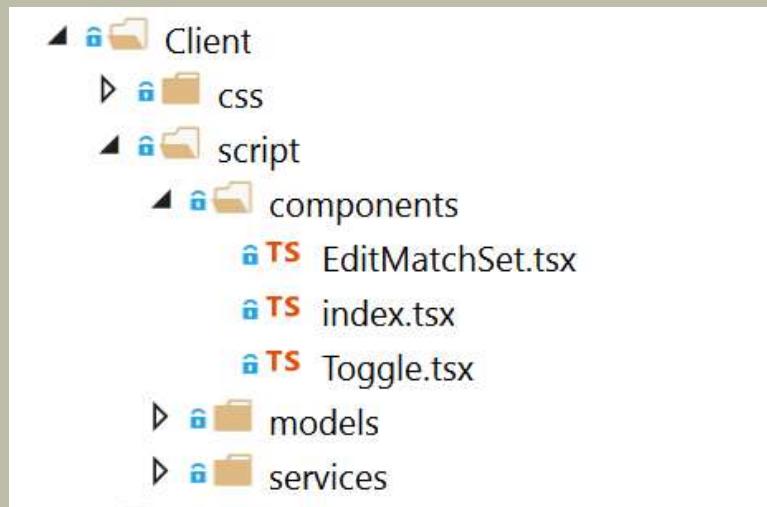
- cmd**: A terminal window showing the command `start "js" ConEmu.exe -runlist` followed by three entries: `npm run watch`, `npm run testwatch -new_console:sh`, and `npm run karma -new_console:sV`.
- npm**: A terminal window showing the output of an npm run command. It includes logs for ts-loader, webpack, and karma, along with asset compilation details:

| Time | Asset | Size | Chunks |
|---------------------|--|---------|--------|
| 5681ms | | | |
| | /models/EditMatchSetStore.specs.js | 148 kB | |
| 0 [emitted] | /models/EditMatchSetStore.specs | | |
| | /services/parameters.specs.js | 2.24 kB | |
| 1 [emitted] | /services/parameters.specs | | |
| | /models/EditMatchSetStore.specs.js.map | 172 kB | |
| 0 [emitted] | /models/EditMatchSetStore.specs | | |
| | /services/parameters.specs.map | 2.94 kB | |
| 1 [emitted] | /services/parameters.specs | | |
| + 42 hidden modules | | | |

- node.exe**: A terminal window showing karma test results. It lists failed tests and browser execution details:

| Test | Browser | Status |
|---|-------------------------------------|--------|
| iterator can delete a match | Chrome 55.0.2883 (Windows 10 0.0.0) | FAILED |
| Expected 1 to be 0. | | |
| at Object.<anonymous> (wwwroot/assets/tests/models/EditMatchSetStore.specs.js:129:46) | | |
| Expected 3 to be 5. | | |
| at Object.<anonymous> (wwwroot/assets/tests/models/EditMatchSetStore.specs.js:130:37) | | |
| Chrome 55.0.2883 (Windows 10 0.0.0): Executed 5 of | | |
| Chrome 55.0.2883 (Windows 10 0.0.0): Executed 6 of | | |
| Chrome 55.0.2883 (Windows 10 0.0.0): Executed 6 of | | |
| 6 (1 FAILED) (0.039 secs / 0.007 secs) | | |

Client Code Layout



NuGet and npm Only

```
"scripts": {
  "watch": "webpack -w",
  "testwatch": "webpack --config webpack.tests.config.js -w",
  "karma": "karma start"
},
"dependencies": {},
"devDependencies": {
  "@types/jasmine": "^2.5.38",
  "@types/react": "^0.14.50",
  "@types/react-bootstrap": "0.0.37",
  "@types/react-dom": "^0.14.19",
  "axios": "^0.15.3",
  "bootstrap": "^3.3.7",
  "del": "^2.2.2",
  "es6-promise": "^4.0.5",
  "font-awesome": "^4.6.3",
  "gulp": "^3.9.1",
  "gulp-clean-css": "^2.0.13",
  "gulp-exec": "^2.1.3",
  "jasmine-core": "^2.5.2",
  "jquery": "^3.1.1",
  "karma": "^1.3.0",
  "karma-chrome-launcher": "^2.0.0",
  "karma-jasmine": "^1.1.0",
  "karma-phantomjs-launcher": "^1.0.2",
  "react": "^15.4.1",
  "react-bootstrap": "^0.30.7",
  "react-dom": "15.4.0",
  "run-sequence": "^1.2.2",
  "source-map-loader": "^0.1.5",
  "ts-loader": "1.2.2",
  "typescript": "^2.1.4",
  "webpack": "^1.14.0"
}
```

TypeScript

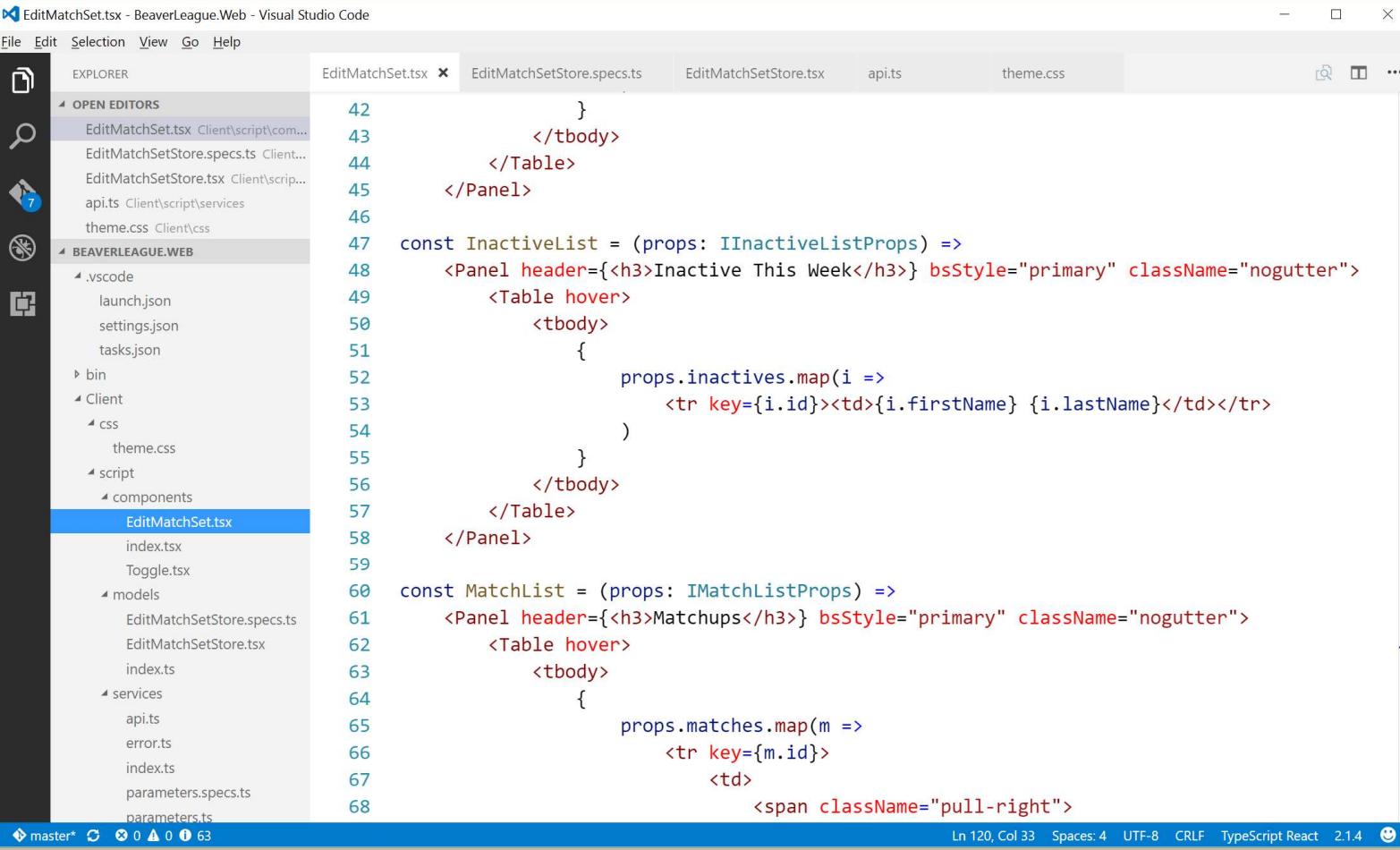
Use TypeScript

```
"compilerOptions": {  
    "noImplicitAny": true,  
    "noEmitOnError": true,  
    "removeComments": false,  
    "sourceMap": true,  
    "module": "commonjs",  
    "target": "es5",  
    "jsx": "react",  
    "baseUrl": ".",  
    "moduleResolution": "node",  
    "traceResolution": true,  
    "lib": ["es2015.promise", "dom", "es5"],  
    "paths": {  
        "*": [ "*", "Client/script/*" ]  
    }  
},  
"compileOnSave": false,  
"exclude": [  
    "node_modules",  
    "wwwroot"  
]
```

Use @types (no tsd, no typings)

```
"devDependencies": {  
  "@types/jasmine": "^2.5.38",  
  "@types/react": "^0.14.50",  
  "@types/react-bootstrap": "0.0.37",  
  "@types/react-dom": "^0.14.19",
```

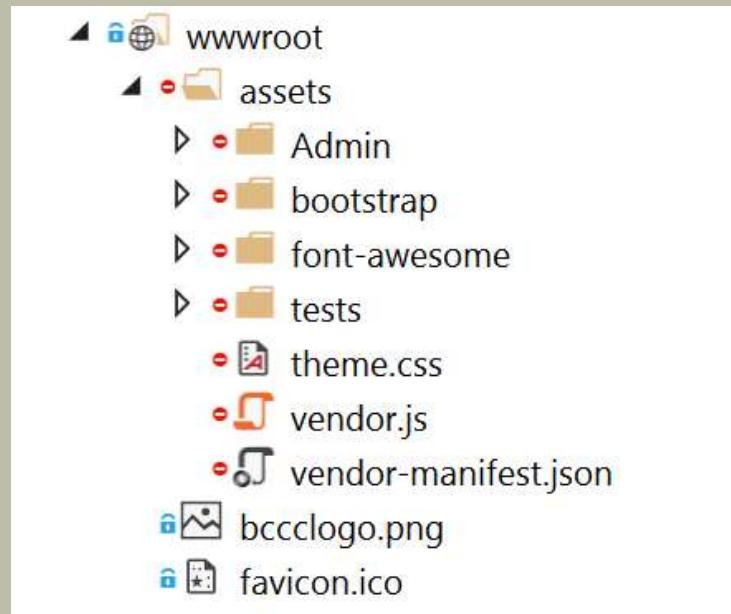
Use Visual Studio Code for TypeScript and ES6+



The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** EditMatchSet.tsx - BeaverLeague.Web - Visual Studio Code
- File Menu:** File Edit Selection View Go Help
- Toolbar:** Standard VS Code icons for file operations.
- Explorer:** Shows the project structure:
 - OPEN EDITORS: EditMatchSet.tsx (highlighted), EditMatchSetStore.specs.ts, EditMatchSetStore.tsx, api.ts, theme.css
 - BEAVERLEAGUE.WEB:
 - .vscode: launch.json, settings.json, tasks.json
 - bin
 - Client:
 - css: theme.css
 - script:
 - components: EditMatchSet.tsx (highlighted)
 - index.tsx
 - Toggle.tsx
 - models:
 - EditMatchSetStore.specs.ts
 - EditMatchSetStore.tsx
 - index.ts
 - services:
 - api.ts
 - error.ts
 - index.ts
 - parameters.specs.ts
 - parameters.ts
- Editor:** The main editor pane displays the code for the `EditMatchSet.tsx` file. The code uses React components (`<Table>`, `<tbody>`, `<tr>`, `<td>`) and prop types defined in `IInactiveListProps` and `IMatchListProps`. The code is numbered from 42 to 68.
- Bottom Status Bar:** master* 0 0 0 0 63 | Ln 120, Col 33 Spaces: 4 UTF-8 CRLF TypeScript React 2.1.4 😊
- Footer:** Ode to Code logo

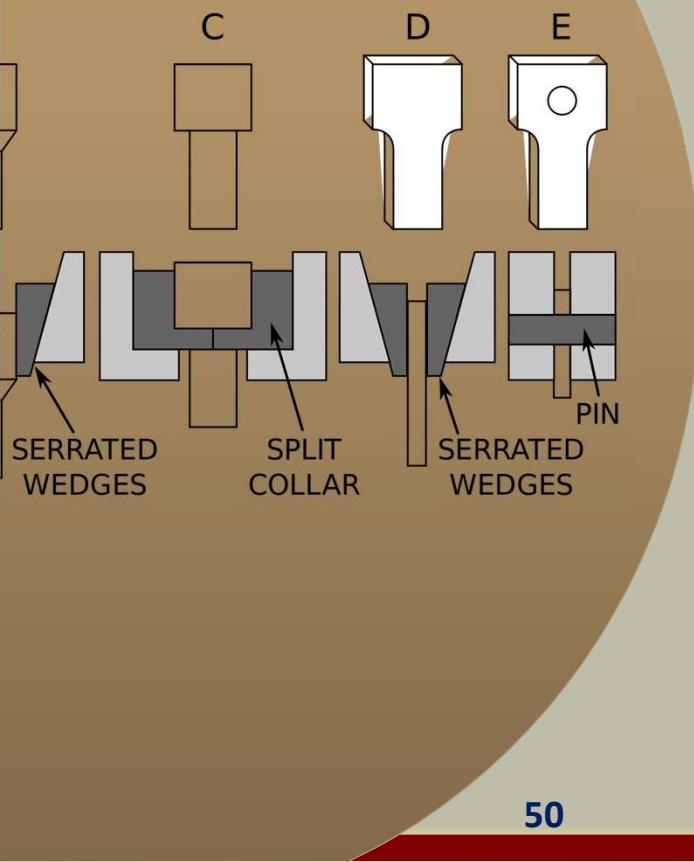
wwwroot is your bin\ folder



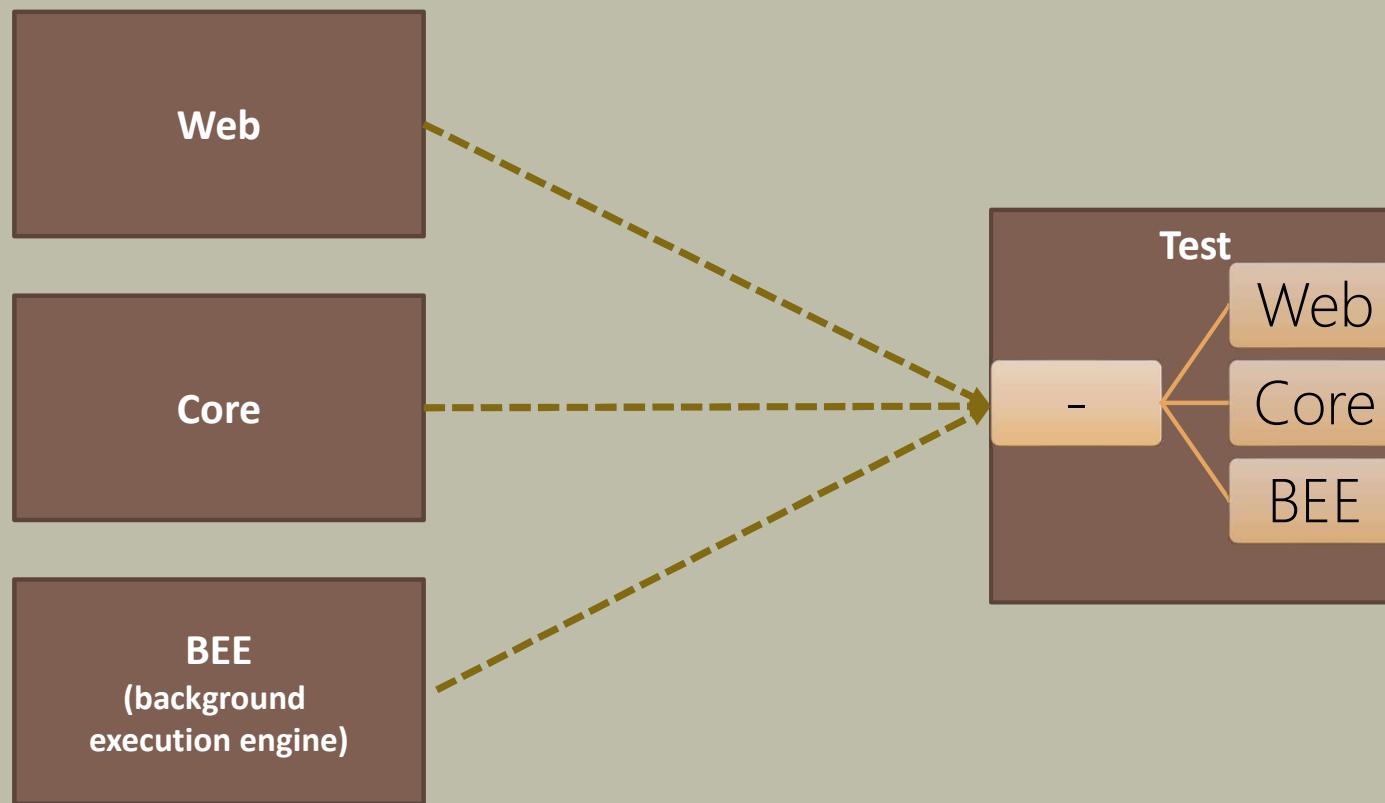
Kick Off Client Components With Parameters

```
<div id="react-creatematchset"
      data-parameters="@Json.Serialize(Model)">
</div>
```

Testing



A Single Test Project Is Not A Bad Idea



Use xUnit



- Extensible, flexible, open source
- Low ceremony
- Good vocabulary

```
[Fact]
public async Task CanCreateSeason()
{
    var controller = new ManageSeasonController(_mediator);
    var inputModel = new CreateSeasonCommand() {Name = "2017"};

    var result = await controller.Create(inputModel);

    var verifySeason = await _verifyContext.Sessions.SingleOrDefaultAsync(s => s.Name == "2017");

    IsType<RedirectToActionResult>(result);
    Equal(verifySeason.Name, "2017");
}
```

Try Hard to Avoid Mocks

≡ DESTROY ALL SOFTWARE

Test Isolation Is About Avoiding Mocks

Stop Using Mocks

23 Mar 2011

Mock yourself, not your tests

Published on Mon 27 July 2015, under [Programming](#), tagged with [testing](#), [python](#), [mocks](#) and [rants](#).

tl;dr: While sometimes mocks are needed, they should not be the first tool you use to write tests. It is a pity to be mocking your code when you could be writing meaningful tests at *almost* [All Your Mocks are Evil!!](#)

[8 Replies](#)

In unit testing, all mocks are evil! Now there's a controversial statement to start a blog post with, but let me explain. I am writing this from my own experience as a software developer and a leader of software developers. This is the sort of thing that software religious wars are made of, so if you agree, or disagree, I would love for you to share your thoughts in the comments.

Testing
Tuesday, I

By Andrew

This articl
download

Making a Mockery of TDD

I made this gnomic remark on Twitter the other day:

To be a successful mockist, you must dislike mocks.

re Dead. Long
tists.

[Testing »](#) [Approaches »](#)

Ode to Code

Use the ASP.NET Core Test Host

Microsoft.AspNetCore.TestHost 1.1.0

ASP.NET Core web server for writing and running tests.

```
[Fact]
public async Task AjaxLoginRedirectToReturnUrlTurnsInto200WithLocationHeader()
{
    var server = CreateServer(new CookieAuthenticationOptions
    {
        AutomaticChallenge = true,
        LoginPath = "/login"
    });

    var transaction = await SendAsync(server, "http://example.com/protected?X-Requested-With=XMLHttpRequest");
    Assert.Equal(HttpStatusCode.Unauthorized, transaction.Response.StatusCode);
    var responded = transaction.Response.Headers.GetValues("Location");
    Assert.Equal(1, responded.Count());
    Assert.True(responded.Single().StartsWith("http://example.com/login"));
}
```

```
[Fact]
public async Task NormalRequestPassesThrough()
{
    var server = CreateServer(new CookieAuthenticationOptions());
    var response = await server.CreateClient().GetAsync("http://example.com/normal");
    Assert.Equal(HttpStatusCode.OK, response.StatusCode);
}
```

Use the EF Core In-Memory Provider

```
[Fact]
public void CanAddGolfer()
{
    using (var leagueData = _db.NewContext())
    {
        leagueData.Golfers.Add(new Golfer());
        leagueData.SaveChanges();
    }

    using (var leagueData = _db.NewContext())
    {
        Equal(1, leagueData.Golfers.Count());
    }
}

readonly DbInstance<LeagueDb> _db = new DbInstance<LeagueDb>(
    options => new LeagueDb(options));
```

Use the EF Core In-Memory Provider

```
public class DbInstance<T> where T : DbContext
{
    public DbInstance(Func<DbContextOptions<T>, T> factory)
    {
        var provider = new ServiceCollection()
            .AddEntityFrameworkInMemoryDatabase()
            .BuildServiceProvider();

        var builder = new DbContextOptionsBuilder<T>();
        builder.UseInMemoryDatabase()
            .UseInternalServiceProvider(provider);

        _options = builder.Options;
        _factory = factory;
    }

    public T NewContext()
    {
        return _factory(_options);
    }

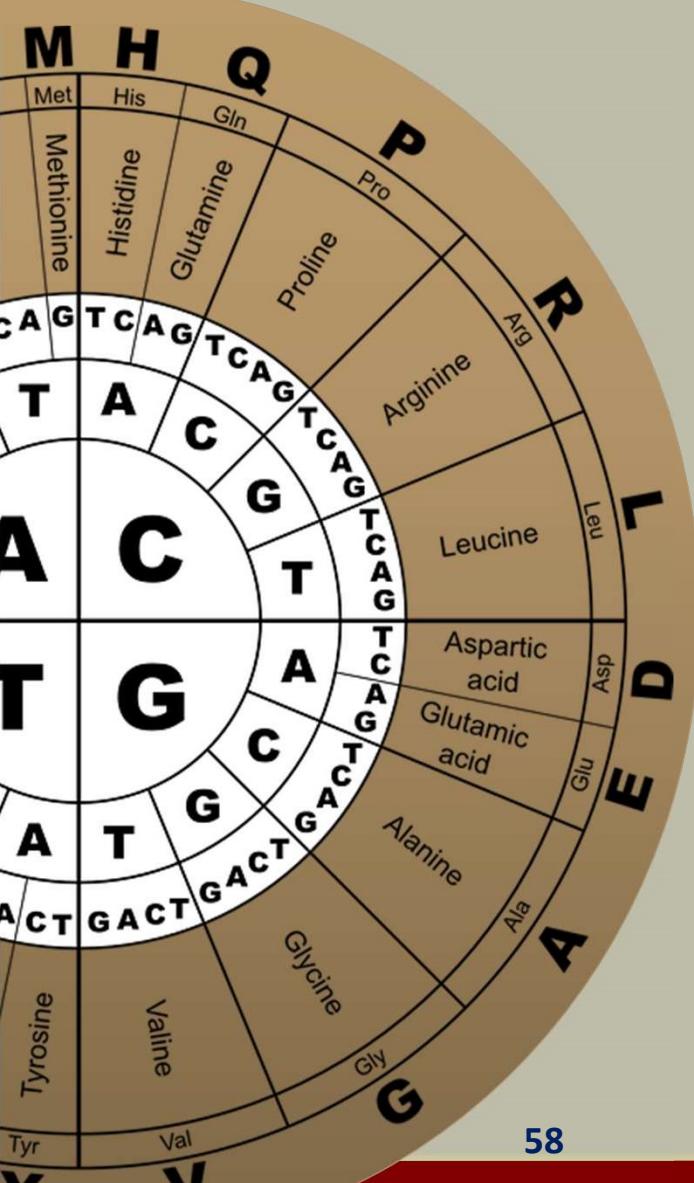
    private readonly DbContextOptions<T> _options;
    private readonly Func<DbContextOptions<T>, T> _factory;
}
```

More Integration Tests & Fewer Unit Tests

```
[Fact]
public async Task CanFailOnDuplicateSeasonNames()
{
    var controller = new ManageSeasonController(_mediator);
    var inputModel = new CreateSeasonCommand { Name = "2017" };

    _testContext.Seasons.Add(new Season {Name = "2017"});
    _testContext.SaveChanges();

    var result = await controller.Create(inputModel) as ViewResult;
    NotNull(result);
    Equal(1, result.ViewData.ModelState.ErrorCount);
}
```



58

C# Code

Use Environment.NewLine

Environment.NewLine Property

.NET Framework (current version)

Other Versions ▾

Gets the newline string defined for this environment.

Use Path.Combine

Path.Combine Method (String, String)

.NET Framework (current version) | [Other Versions ▾](#)

Combines two strings into a path.

Be Aware of Environment Variables

```
var homeDir = Environment.GetEnvironmentVariable("USERPROFILE")
    ?? Environment.GetEnvironmentVariable("HOME");
```

Don't Use the Async Suffix

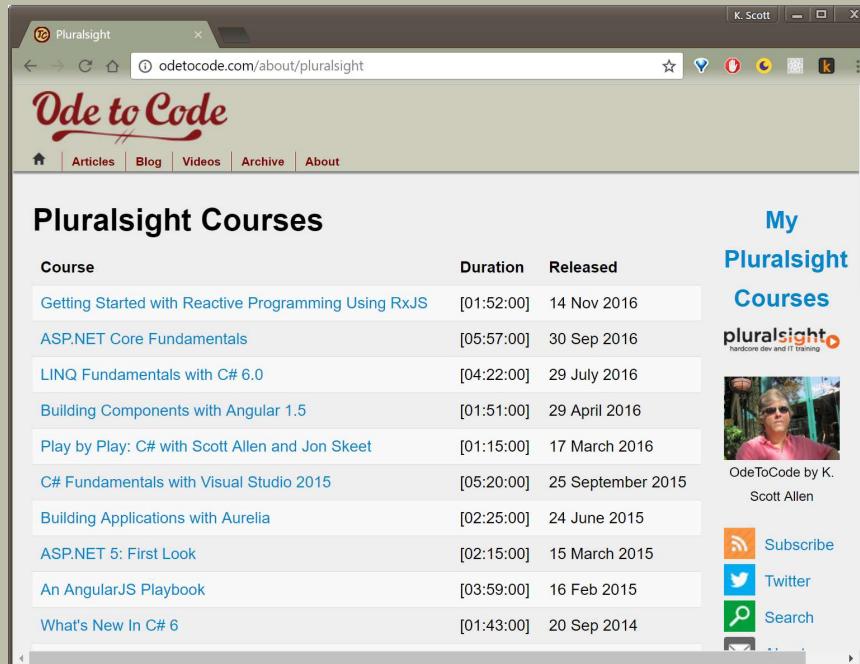
```
public async Task<bool> SignOutTenantUser()
{
    await _contextAccessor.HttpContext.Authentication.SignOutAsync(SchemeName);
    return true;
}
```

One of my favorite insights about syntax design appeared in a [retrospective on C++](#) by Bjarne Stroustrup:

- For new features, people insist on **LOUD** explicit syntax.
- For established features, people want terse notation.

Thanks!

scott@OdeToCode.com



The screenshot shows a web browser window with the URL odetocode.com/about/pluralsight. The page displays a list of Pluralsight courses taught by Scott Allen. The courses listed are:

| Course | Duration | Released |
|--|------------|-------------------|
| Getting Started with Reactive Programming Using RxJS | [01:52:00] | 14 Nov 2016 |
| ASP.NET Core Fundamentals | [05:57:00] | 30 Sep 2016 |
| LINQ Fundamentals with C# 6.0 | [04:22:00] | 29 July 2016 |
| Building Components with Angular 1.5 | [01:51:00] | 29 April 2016 |
| Play by Play: C# with Scott Allen and Jon Skeet | [01:15:00] | 17 March 2016 |
| C# Fundamentals with Visual Studio 2015 | [05:20:00] | 25 September 2015 |
| Building Applications with Aurelia | [02:25:00] | 24 June 2015 |
| ASP.NET 5: First Look | [02:15:00] | 15 March 2015 |
| An AngularJS Playbook | [03:59:00] | 16 Feb 2015 |
| What's New In C# 6 | [01:43:00] | 20 Sep 2014 |

To the right of the course list, there is a sidebar titled "My Pluralsight Courses" which includes a photo of Scott Allen, his name, and links to "Subscribe", "Twitter", and "Search".

Ode to Code