ASP 5 Workshop

Getting Started

AmmonsOnline.com

Who Am I?

- Jeff Ammons
  - Principal Architect
  - Sage Software

- President
  - GGMUG
  - Gwinnett, Georgia, Microsoft User Group
  - 2nd Thursday
  - Gwinnett Technical College
  - http://GGMUG.com

http://ammonsonline.com

Why is ASP Changing?

- Market Share/Mind Share
  - Web developers choose:
    - Mac for development
    - Linux for deployment
  - Dynamic languages popular
- Cross Platform
  - See above…
- Technical Deficits
  - .Net built for desktop apps
  - Not optimized for server apps

http://ammonsonline.com

Cynical History of ASP

- ASP (classic)
  - Hey, look, we have a web stack!
  - Better than Perl/CGI!
- ASP.Net
  - Pretend the web is Windows!
- ASP.Net MVC
  - We can do what Ruby on Rails does!
- ASP.Net Web API
  - We have to use WCF, damnit!
- Where'd everybody go?
- ASP 5
  - We can do what Node does!
  - We'll let you work on Mac!
  - We'll let you deploy to Linux!
  - You don't *have* to buy Visual Studio!

http://ammonsonline.com

What's Staying The Same

- MVC
  - Controllers
  - Views
  - Models
  - Routes*
  - Razor*
- Web API
  - Controllers
  - Routes*

http://ammonsonline.com

* A few changes to Routes and Razor

What's New?

- Open Source
- Cross Platform
- Modular
- OPT*

- *Other People's Tools*

http://ammonsonline.com

What's New?

- .Net Core
  - Deployable CLR
- Nuget is first class citizen
- Dependency Injection by default
- wwwroot
  - Lock down files that will be served
- Node.js
  - NPM: Node Package Manager
  - Grunt/Gulp for automation
  - Bower for *client* package management

http://ammonsonline.com

What's Different?

- Project file
  - project.json instead of web.config
- Configuration
  - config.json instead of web.config
  - Environment vars instead of web.confg
- Minification and Bundling
  - grunt/gulp instead of ASP.net
  - Bundle/minify at *compile time* not runtime

http://ammonsonline.com

Current Timeline

- Beta 6: End of July 2015
- Beta 7: End of August 2015
- Beta 8: End of September 2015
- RC: ~November (Go Live License?)
- RTM: Early 2016

http://ammonsonline.com

ASP 5 Cheat Sheet

- project.json
  - Replaces web.config
  - Define webroot directory
  - Register dependencies
    - Nuget packages
  - Register commands
  - Define which frameworks are valid (.Net full, .Net Core)
  - Exclude files & directories from project
- wwwroot
  - Web accessible files ~ think content dir in older mvc
    - Images, scripts, stylesheets, etc.
  - Easier to isolate the files you want to serve
- config.json
  - Replaces web.config for appsettings & connection strings
  - Default option, you can also use xml, ini, or env vars
  - Expected path: config.json for local dev, env vars for servers
- bower.json
  - Manages *client* side dependencies
  - Like Nuget for javascript libs
- gulpfile.js
  - Define build and publish tasks
    - Copy files, minify css & javascript, etc.
  - Task Runner Explorer
- Startup.cs
  - Entry point for app
  - Add “Middleware” to pipeline
  - Dependency Injection
  - Configure Routes

http://ammonsonline.com

Resources

- https://github.com/aspnet/Home
- http://docs.asp.net/en/latest/index.html
- https://www.microsoftvirtualacademy.com/en-us/training-courses/what-s-new-with-asp-net-5-8478
- https://azure.microsoft.com/en-us/documentation/articles/web-sites-dotnet-deploy-aspnet-mvc-app-membership-oauth-sql-database/#add-a-database-to-the-application
- http://ef.readthedocs.org/en/latest/getting-started/aspnet5.html
- http://www.codeproject.com/Tips/988763/Database-Migration-in-Entity-Framework
- http://www.bricelam.net/2014/09/14/migrations-on-k.html
- http://stephenwalther.com/archive/2015/02/24/top-10-changes-in-asp-net-5-and-mvc-6

http://ammonsonline.com

Contact Info

- Twitter: jeffa00
- Google Plus: jeffa00
- Linked-In: jeffammons
- Blog: http://ammonsonline.com
- YouTube: http://youtube.com/ammonsonline
- SciFi/Humor: http://galacticbeacon.com
- User Group: http://ggmug.com

Workshop

- Windows Phone Spotter App
  - Log every time you see a Windows Phone in the Wild!
  - Goal: Highlight the familiar, introduce a bit of the new
    - Show that lots of the new changes are optional
- Warning:
  - BETA BETA BETA
    - Feels more like ALPHA ALPHA ALPHA
  - Some things can be
    - Broken Broken Broken
    - YMMV: Your Mileage May Vary. Wildly.
  - Let’s keep things simple!

http://ammonsonline.com

Workshop

- Feel free to work at your own speed or follow along
- Feel free to skip the optional Prequel section tonight and try that later
- My Samples:
  - https://github.com/jeffa00/WinPhoneSpotter01
  - http://winphonespotter01.azurewebsites.net/
    - Note: They will likely change in the future

http://ammonsonline.com

Why Did Visual Studio Stop Working???

- Updating ASP 5 from command line borks VS
- `dnvm upgrade`
  - Pulls latest version AND sets it to be used and makes it the “default”
- Visual Studio uses whichever version is defined assigned to the Alias “default”
- Fix by setting the default alias to the version Visual Studio needs
- `dnvm alias default <semver> [-x86][-x64] [-svr50][-svrc50]`

http://ammonsonline.com

Why Did Visual Studio Stop Working???

- The best part is that VS doesn't set the path for you
- If you need to use Entity Framework, you currently HAVE to use the command line
- That means you need to do the upgrade, set default dance at least once
- If you set the path manually, then you will point to the wrong command line tool after you do upgrade…

http://ammonsonline.com

User Stories

- Prequel:
  - Create project
  - Initial setup
  - Optional Bits
    - GitHub
    - Azure
- Story 1: User wants to log in
- Story 2: User wants to see list of sightings
- Story 3: User wants to log a phone spotting

http://ammonsonline.com

Prequel: Create Project

- File
  - New
  - Project…
  - Web
  - ASP.NET Web Application
  - ASP.NET 5 Preview Templates
  - Web Sites

  - Wait…

http://ammonsonline.com

Prequel: Change Site Title

- Open config.json
- Change AppSettings: SiteTitle value to
  - Windows Phone Spotter Dev

http://ammonsonline.com

Prequel (Optional): Commit in Git

- Team Explorer
- Changes
- Enter commit message: “initial commit”
- Click “Commit”

http://ammonsonline.com

Prequel (Optional): Add to Git Server

- GitHub with 2015 GitHub explorer
- Sync
- Make sure you are connected to your GitHub acct
- Either:
  - New Repo
    - Enter new Repo Name
    - Publish
  - OR
  - Existing Repo
    - Enter URL (https version) of existing Repo
    - Publish

http://ammonsonline.com

Prequel (Optional): Create Azure WebApp

- If you plan to go this route I'd do it early in the process so you are testing the fewest of *your* changes
- NOTE: You will have to work through the EF migrations if you test this way.

- Using Azure Portal
- Create a new Web App
- Link to your GitHub repo
- …
- Profit!

http://ammonsonline.com

Prequel (Optional): Change Azure Site Title

- Application Settings
- App Settings
  - Key: AppSettings:SiteTitle
  - Value: Windows Phone Spotter

http://ammonsonline.com

Prequel (Optional): Create Azure SQL Db

- Use Portal to create a new DB
- Or use an existing one

http://ammonsonline.com

Prequel (Optional): Add AzureSQL Connection String

- Get connection string from portal
- Go to Web App in Portal
  - Application Settings
    - Show Connection Strings
    - Add
      - Key: DefaultConnection
      - Value: Paste in conn string

http://ammonsonline.com

Prequel (Optional): Test Login

- At this point if things are swinging your way:
  - Register
  - Add email address and pwd
  - Should create DB/tables/add new user
  - …
  - More Profit!

http://ammonsonline.com

Story 1: User wants to log in

- For simplicity we'll use the default identity
- We'll also take the simple route for dev/prod
  - Dev/local in config.json
    - Don't change it – leave it local and trusted/Never check in credentials…
      - Windows security, not SQL Server Login: No user/pwd
  - Prod from environment variable
- SecretManager is nearly usable
  - Works sometimes
  - For now I'd stick with the local/ trusted_connection for dev

http://ammonsonline.com

Story 2:
User wants to see list of sightings

- Create basic Model Objects
  - PhoneSpotting
    - PhoneSpottingId
    - SpotTime
    - PhoneManufacturer
    - PhoneModel
    - Latitude
    - Longitude
    - Notes
  - PhoneSpottingsViewModel
    - List<PhoneSpotting>

http://ammonsonline.com

Story 2:
User wants to see list of sightings

- Add PhoneSpottingContext
- Register in Startup.cs
  - In ConfigureServices after ApplicationDbContext

```
services.AddEntityFramework()
  .AddDbContext<PhoneSpottingContext>(options =>
    options.UseSqlServer(Configuration["Data:DefaultConnection:ConnectionString"]));
```

- Generate EF Migration
  - Command line (likely added to VS in the future)
  - You'll need to have done the "upgrade, set default" dance to have dnx on your path
  - cd to src directory
  - Run
    - `dnx . ef migration add PhoneSpotterCreate –c PhoneSpottingContext`

http://ammonsonline.com

Story 2:
User wants to see list of sightings

- Create PhoneSpottingController
- Submit list of spottings to View in Index
  - `return View(new PhoneSpottingsViewModel { Spottings = _spottingContext.Spottings.ToList() });`
- Create View
  - PhoneSpotting/Index.cshtml

http://ammonsonline.com

Story 3:
User wants to log a phone spotting

- Create Create method(s)

```
// GET: /<controller>/Create
        [Authorize]
        public IActionResult Create()
        {
            return View();
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        [Authorize]
        public IActionResult Create(PhoneSpotting newSpotting)
        {
            if (ModelState.IsValid)
            {
                _spottingContext.Spottings.Add(newSpotting);
                _spottingContext.SaveChanges();
                return RedirectToAction("Index");
            }

            return View(newSpotting);
        }
```

http://ammonsonline.com

Story 3:
User wants to log a phone spotting

- Create View
  - Try out some tag-helpers

```
<form asp-controller="PhoneSpotting" asp-action="Create" method="post" class="form-horizontal">
```

  - For fun you can prefill lat/long from javascript
    - The crux of it is this:

```
navigator.geolocation.getCurrentPosition(function(position){
       $("#Longitude").val(position.coords.longitude);
       $("#Latitude").val(position.coords.latitude);

       var currentTime = new Date();
       $('#SpotTime').val(currentTime.toLocaleString());
       });
```

http://ammonsonline.com