TechDays

Discover
endless
possibilities

2017

Microsoft

#techdays2017

Building Secure ASP.NET Core MVC Applications

Niels Tanis
CA | Veracode

Microsoft

About me

- Niels Tanis
- Security Researcher
- Background in:
  - .NET development
  - Pentesting
  - Security Consultancy
  - CSSLP

ca technologies

VERACODE

Building (secure) applications is hard!

WHAT I'M TRYING TO DO IS REALLY SIMPLE.
IT SHOULDN'T BE HARD.
ALL COMPUTERS ARE JUST CAREFULLY ORGANIZED SAND. *EVERYTHING* IS HARD UNTIL SOMEONE MAKES IT EASY.
MAYBE I SHOULD TURN THIS ONE *BACK* INTO SAND.
I'LL FIND A BLOWTORCH.

https://xkcd.com/1349/

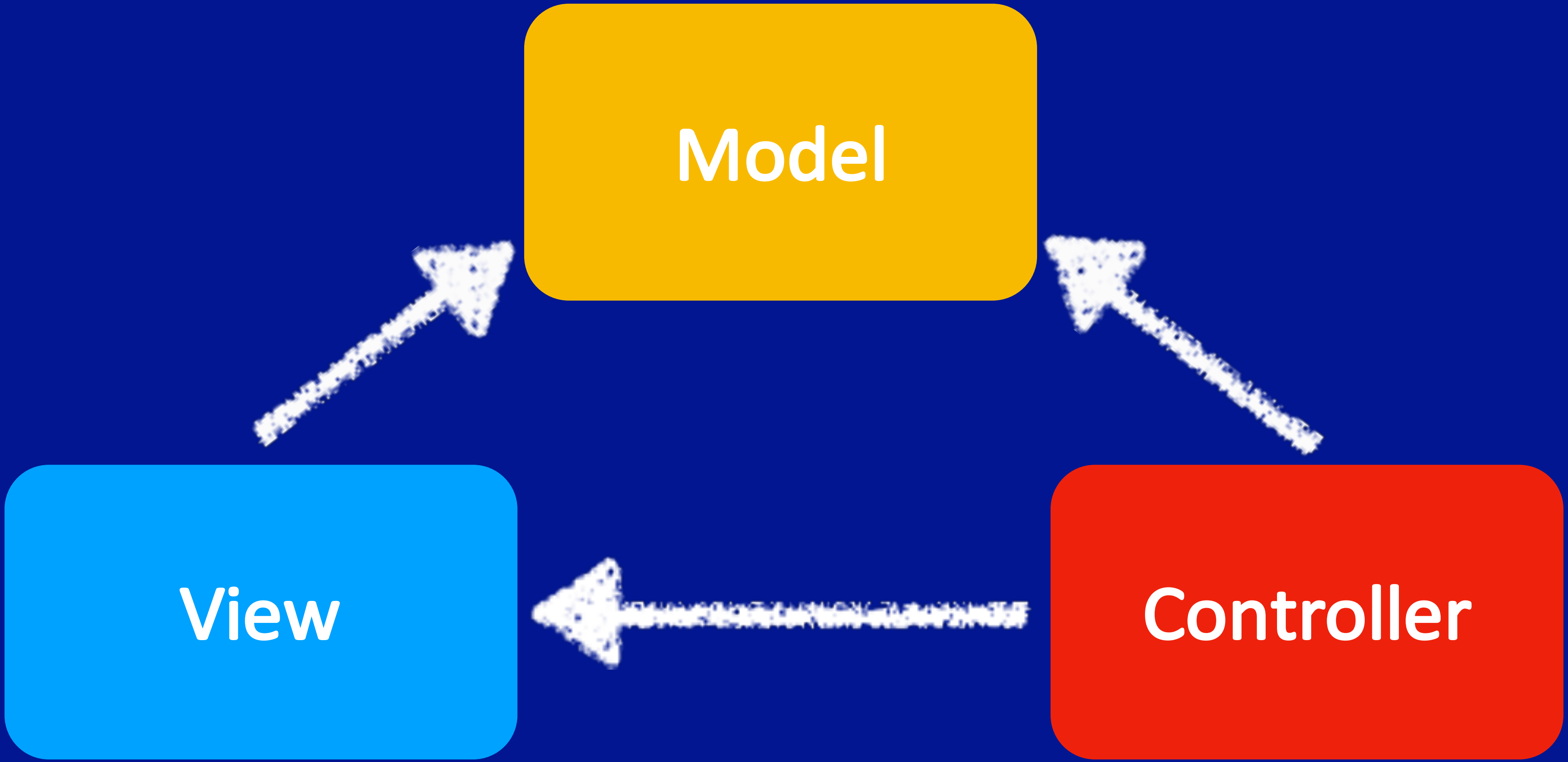
Agenda

- Introduction
- Building Secure ASP.NET Core MVC Applications
  - Processing data
  - Returning data
  - Adapt web standards
  - Analysing existing solutions
- Conclusion
- Q&A

.NET Core and ASP.NET Core MVC

- .NET Core
  - Open Source
  - Modular based
  - Multi platform (Windows, MacOSX, Linux)
- ASP.NET Core MVC
  - Complete rewrite of ASP.NET MVC
  - Runs on .NET Core and .NET Framework

Model-View-Controller



Program

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

            host.Run();
        }
    }
}
```

Controller

```
namespace WebApplication
{
    public class DefaultController
    {
           public string Index()
           {
                 return $"Hello from {this.GetType().ToString()}!”;
           }
    }
}
```

Controller

- Convention based resolving: ”…Controller”
- Any referenced assembly can expose controllers!
- ConfigureServices
    - ApplicationPartsManager
    - ControllerFeatureProvider

ApplicationPartsManager

```
public void ConfigureServices(IServiceCollection services)
{

        …
        services.AddMvc().ConfigureApplicationPartManager(p =>
        {
                var libsToExclude = p.ApplicationParts.Where(x =>
                !x.Name.StartsWith(this.GetType().Namespace) &&
                !x.Name.StartsWith("Microsoft.AspNetCore")).ToList();


                libsToExclude.ForEach(l =>
                p.ApplicationParts.Remove(l));
        });
        …
}
```

MVC Routing in Startup ConfigureServices

```
app.UseMvc(routes =>
{
    routes.MapRoute(
        name: "default",
        template: "{controller=Home}/{action=Index}/{id?}");
});
```

https://localhost/webapp/info/edit?input=data

https://localhost/webapp/info/delete/2

Model Binding

- Model binding will fetch data from:
  - Form Inputs
  - Route Parameters
  - Query string

Processing Data

```
public class OrdersController : Controller
{
    private readonly OrderDataContext _context;

    public OrdersController(OrderDataContext context)
    {
        _context = context;
    }

    public async Task<IActionResult> CreateNew(Order order)
    {
        _context.Add(order);
        await _context.SaveChangesAsync();
        return RedirectToAction("Index");
    }
}
```

Processing Data

```
[HttpPost]
public async Task<IActionResult> CreateNew(Order order)
{
    _context.Add(order);
    await _context.SaveChangesAsync();
    return RedirectToAction("Index");
}
```

Processing Data

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> CreateNew(Order order)
{
    _context.Add(order);
    await _context.SaveChangesAsync();
    return RedirectToAction("Index");
}
```

Processing Data

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> CreateNew(Order order)
{
    if (ModelState.IsValid)
    {
        _context.Add(order);
        await _context.SaveChangesAsync();
        return RedirectToAction("Index");
    }
    return View(order);
}
```

Processing Data

```
public class Order
{
    public int ID { get; set; }
    [EmailAddress]
    public string Email { get; set; }
    [MaxLength(255)]
    public string Description { get; set; }
    public decimal TotalPrice { get; set; }
    public IEnumerable<OrderDetail> Details { get; set; }
}
```

Processing Data

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> CreateNew(
                      [Bind("Description,Email")]Order order)
{
    if (ModelState.IsValid)
    {
        _context.Add(order);
        await _context.SaveChangesAsync();
        return RedirectToAction("Index");
    }
    return View(order);
}
```

Processing Data

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> EditOrderDetail(int? id) {
    var detail = await _context.Details
         .SingleOrDefaultAsync(c => c.ID == id);
    if (await TryUpdateModelAsync(orderDetail)) {
         try {
              await _context.SaveChangesAsync();
         }
         catch (DbUpdateException /* ex */) {
              ModelState.AddModelError("ERR", ”Not saved.”);
         }
         return RedirectToAction("Index");
    }
    return View(orderDetail);
}
```

Processing Data

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> EditOrderDetail(int? id) {
      var detail = await _context.Details
            .SingleOrDefaultAsync(c => c.ID == id);
      if (await TryUpdateModelAsync(orderDetail, x => x.Ammount)) {
            try {
                  await _context.SaveChangesAsync();
            }
            catch (DbUpdateException /* ex */) {
                  ModelState.AddModelError("ERR", ”Not saved.”);
            }
            return RedirectToAction("Index");
      }
      return View(orderDetail);
}
```

Processing Data

- HTTP GET should never change internal state.
- CSRF protection with the use of [ValidateForgeryToken] or AutoValidateForgeryTokenFilter configuring MVC
- New taghelpers ASP.NET Core 2.0

ConfigureServices

```
public void ConfigureServices(IServiceCollection services)
{
      …
      services.AddMvc(options => options.Filters.Add(
                      new AutoValidateAntiforgeryTokenAttribute()));
      …
}
```

Processing Data

- Explicit model validation and binding
    - Data Annotations, ModelState.IsValid
    - [FromHeader], [FromQuery], [FromRoute], [FromForm] & [FromBody]
- Overposting data
    - Bind[(….)]
    - TryUpdateModelAsync<T>(obj, ””, x=> x.Property)
    - Specific ViewModels

Presenting Data

```
public class InfoController
{
    public string Index(string name)
    {
        return $"Hello {name}!";
    }
}
```

Presenting Data

```
public class InfoController
{
      [Produces("text/html")]
      public string Index(string name)
      {
            return $"Hello {name}!";
      }
}
```

Presenting Data

```
public class InfoController
{
    public ContentResult Index(string name)
    {
        return new ContentResult
        {
            Content = $"Hello {name}!"
        };
    }
}
```

Presenting Data

```
public class InfoController
{
      public ContentResult Index(string name)
      {
            return new ContentResult
            {
                  Content = $"Hello {name}!”,
                  ContentType = "text/html"
            };
      }
}
```

Presenting Data

```
public class InfoController
{
    readonly HtmlEncoder _htmlEncoder;
    public InfoController(HtmlEncoder htmlEncoder)
    {
        _htmlEncoder = htmlEncoder;
    }

      public ContentResult Index(string name)
      {
           return new ContentResult
           {
                Content = $"Hello {_htmlEncoder.Encode(name)}!",
                ContentType = "text/html"
           };
      }
}
```

Razor View – Presenting Data

```
@model IEnumerable<WebApplication.Models.Order>
…
<table class=“table”>
@foreach (var item in Model) {
        <tr>
           <td>@item.Email</td>
           <td>@Html.Raw(item.Description)</td>
           <td>@Html.DisplayFor(modelItem => item.TotalPrice)</td>
           <td><a asp-action="Edit" asp-route-id="@item.ID">Edit</a>
           <a asp-action="Delete" asp-route-id=“@item.ID">Delete</a>
           </td>
        </tr>
}
</table>
```

Presenting Data

- Input validation and context specific output encoding
- Default encoders
  - HtmlEncoder, JavascriptEncoder & UrlEncoder
- Taghelpers do a good job!
- @Html.Raw(...) → HtmlString

Extending ASP.NET Core MVC

- SameSite Cookie
  - https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site-00
- Lax and Strict Policy
- Mitigating control for CSRF
- Timing and Information leagage resource size attack
  - https://tom.vg

Override CookieManager in Startup

```
public void ConfigureServices(IServiceCollection services)
{
      …
      services.AddDbContext<OrderDataContext>(options =>
        options.UseSqlServer(
           Configuration.GetConnectionString(“OrderConnection")));

      services.AddMvc();
      services.Configure<CookieAuthenticationOptions>(opts =>
      {
           opts.CookieManager = new SameSiteCookieManager();
      });
      …
}
```

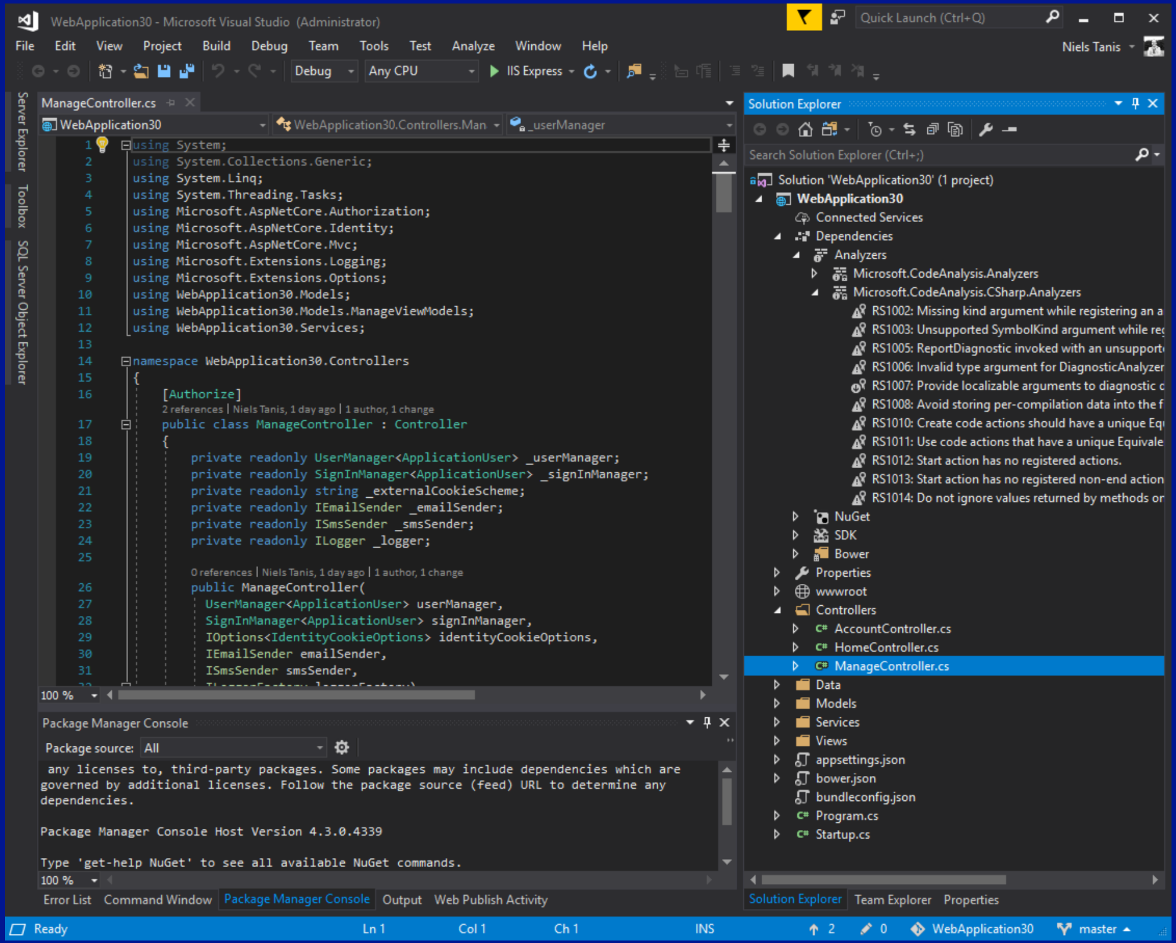
ASP.NET Core 2.0 - CookiePolicyMiddleware

```
namespace Microsoft.AspNetCore.Builder
{
     public class CookiePolicyOptions
     {

           …
           /// <summary>
           /// Affects the cookie's same site attribute.
           /// </summary>
           public SameSiteMode MinimumSameSitePolicy { get; set; }
                                                  = SameSiteMode.Lax;
           …
     }
}
```

Analyse Existing Solutions

- Roslyn compiler
- Microsoft.CodeAnalysis



```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.Options;
using WebApplication30.Models;
using WebApplication30.Models.ManageViewModels;
using WebApplication30.Services;

namespace WebApplication30.Controllers
{
    [Authorize]
    2 references | Niels Tanis, 1 day ago | 1 author, 1 change
    public class ManageController : Controller
    {
        private readonly UserManager<ApplicationUser> _userManager;
        private readonly SignInManager<ApplicationUser> _signInManager;
        private readonly string _externalCookieScheme;
        private readonly IEmailSender _emailSender;
        private readonly ISmsSender _smsSender;
        private readonly ILogger _logger;

        0 references | Niels Tanis, 1 day ago | 1 author, 1 change
        public ManageController(
          UserManager<ApplicationUser> userManager,
          SignInManager<ApplicationUser> signInManager,
          IOptions<IdentityCookieOptions> identityCookieOptions,
          IEmailSender emailSender,
          ISmsSender smsSender,
```

Solution Explorer

Search Solution Explorer (Ctrl+;)

```
Solution 'WebApplication30' (1 project)
  WebApplication30
    Connected Services
    Dependencies
      Analyzers
        Microsoft.CodeAnalysis.Analyzers
        Microsoft.CodeAnalysis.CSharp.Analyzers
          RS1002: Missing kind argument while registering an a
          RS1003: Unsupported SymbolKind argument while reg
          RS1005: ReportDiagnostic invoked with an unsupport
          RS1006: Invalid type argument for DiagnosticAnalyzer
          RS1007: Provide localizable arguments to diagnostic d
          RS1008: Avoid storing per-compilation data into the f
          RS1010: Create code actions should have a unique Eq
          RS1011: Use code actions that have a unique Equivale
          RS1012: Start action has no registered actions.
          RS1013: Start action has no registered non-end action
          RS1014: Do not ignore values returned by methods or
      NuGet
      SDK
      Bower
    Properties
    wwwroot
    Controllers
      AccountController.cs
      HomeController.cs
      ManageController.cs
    Data
    Models
    Services
    Views
    appsettings.json
    bower.json
    bundleconfig.json
    Program.cs
    Startup.cs
```

```
 any licenses to, third-party packages. Some packages may include dependencies which are
governed by additional licenses. Follow the package source (feed) URL to determine any
dependencies.

Package Manager Console Host Version 4.3.0.4339

Type 'get-help NuGet' to see all available NuGet commands.
```

Microsoft.CodeAnalysis

```
SyntaxTree tree; //Contains syntaxtree controller
var root = (CompilationUnitSyntax)tree.GetRoot();
var publicMethods =
root.DescendantNodes().OfType<MethodDeclarationSyntax>()
      .Where(x => x.Modifiers.Any(SyntaxKind.PublicKeyword));

foreach (var method in publicMethods) {
      var attributes = method.AttributeLists
                             .SelectMany(x => x.Attributes);
            if (attributes.Any(x => x.Name.ToString() == “HttpPost"))
            { //Validate that ValidateForgeryToken   }
            var invocations = method.DescendantNodes()
            .OfType<MemberAccessExpressionSyntax>().Where(x =>
            x.Name.ToFullString() == "ModelState.IsValid").ToList();
            //If no found flag method for not checking modelstate.
}
```

Conclusion

- Be aware of attack surface and basic rules like validation and proper output encoding!
- Compared to ASP.NET MVC defaults of Core MVC are better!
- API’s give good guidance and explict changes needed to be ‘unsafe’
- Quick release cycle allows lot of innovation and change

Questions?

Thanks!

- ntanis at veracode.com
- @nielstanis
- https://github.com/nielstanis/techdays2017

Microsoft