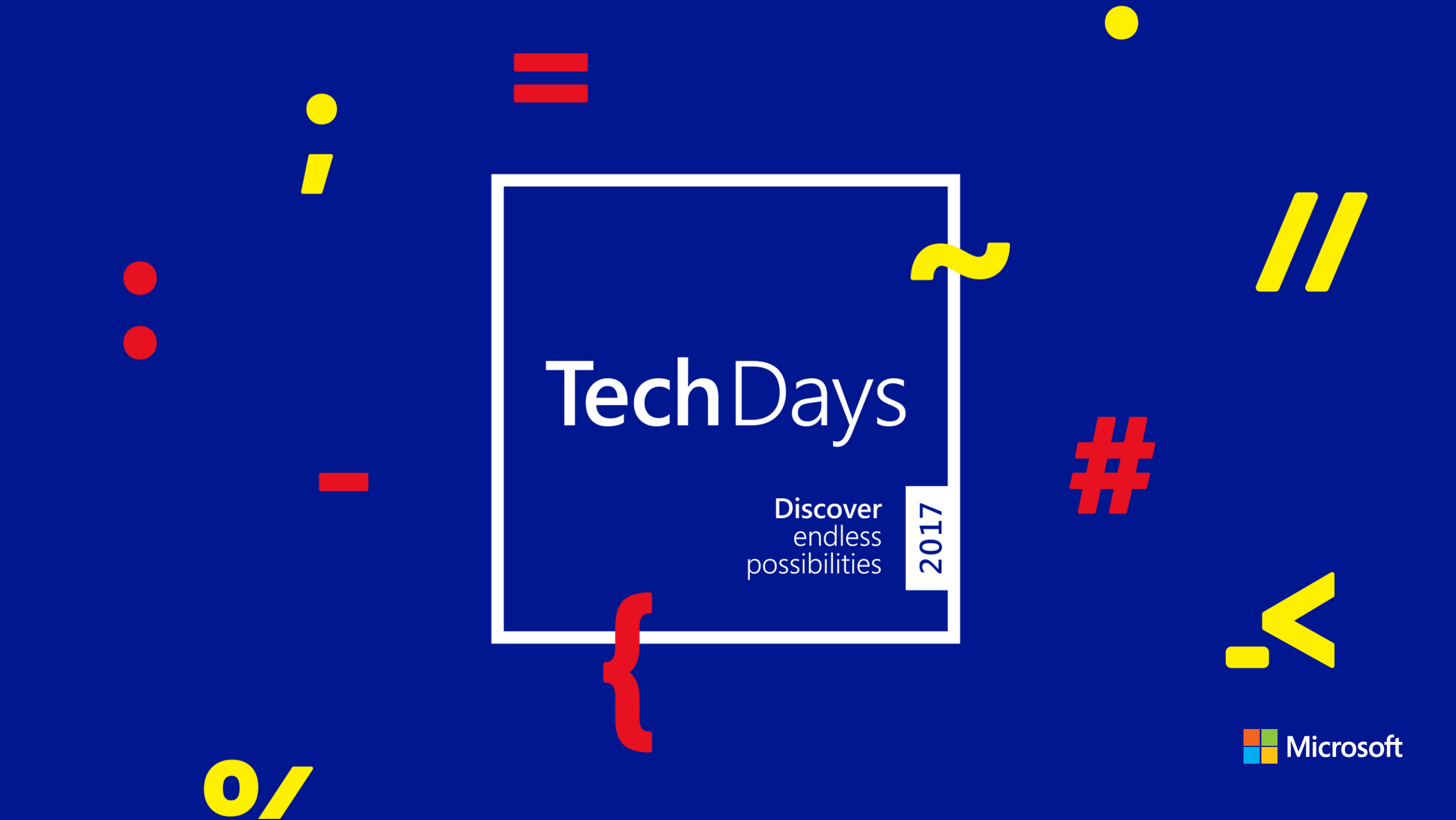

TechDays
Discover
endless
possibilities
2017
Microsoft

#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

Building (secure) applications is hard!

https://xkcd.com/1349/

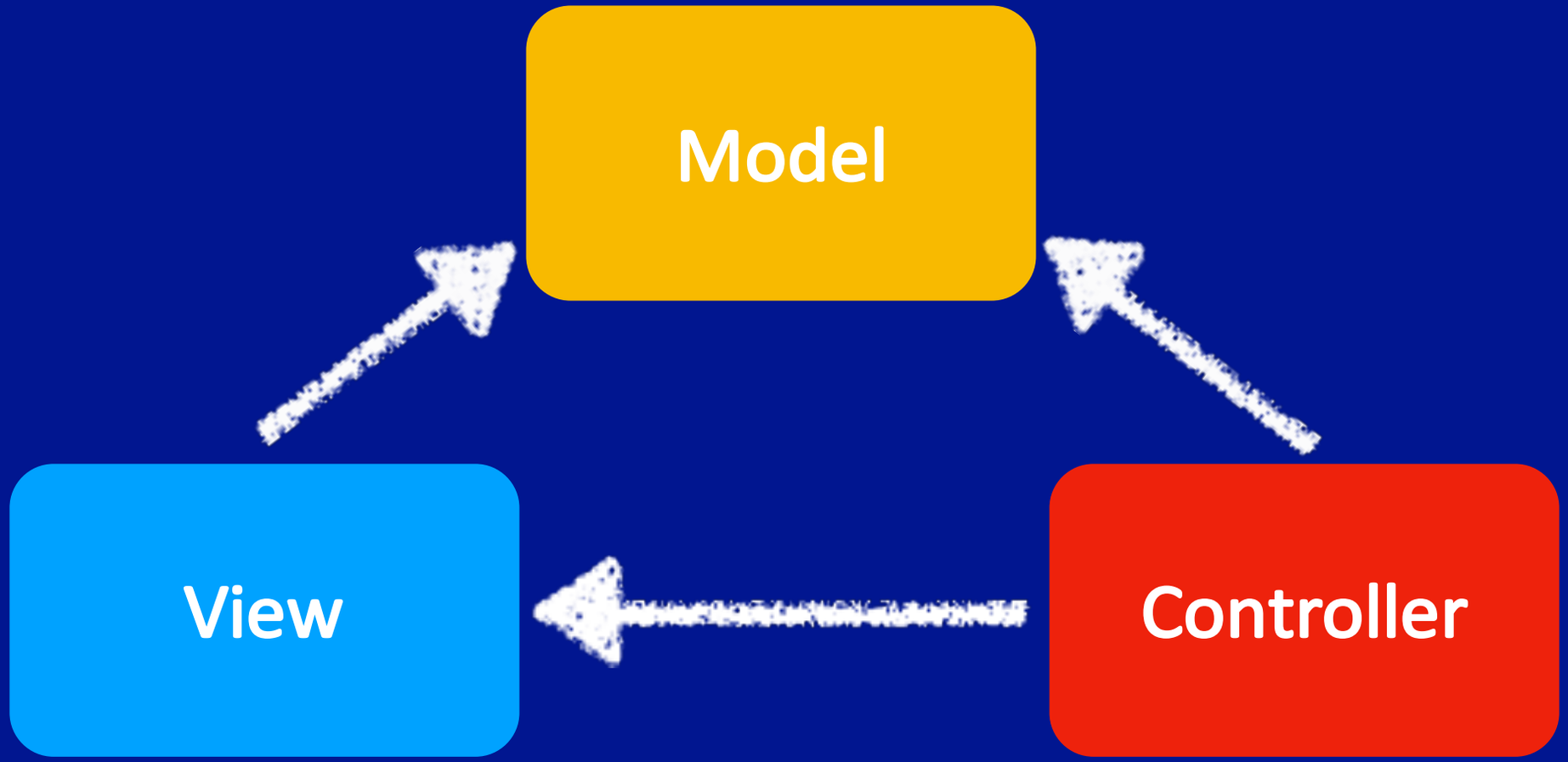
Agenda

- 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; }
    [Required, 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

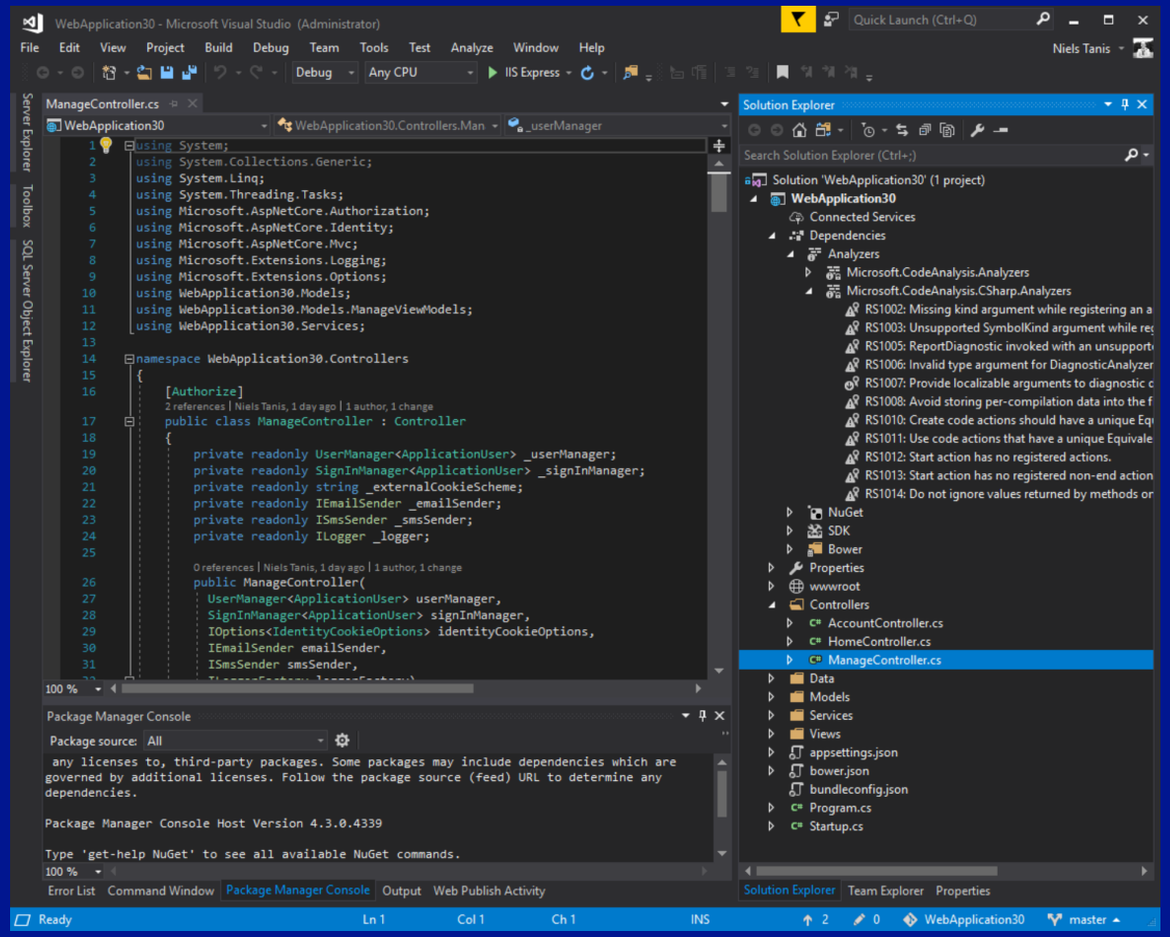
```
public void ConfigureServices(IServiceCollection services)
{
        …
        services.AddDbContext<OrderDataContext>(options =>
           options.UseSqlServer(
              Configuration.GetConnectionString(“OrderConnection")));

        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



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 explicit 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