Setting Up Your C# Pit of Success

with AL Rodriguez

Duende

About Me

- @ProgrammerAL
- https://ProgrammerAL.com
- Customer Success Engineer at Duende

Duende

Why are we here?

- Discuss a C# Pit of Success
  - Enforce code quality
- Present ***Recommendations*** for ***Concepts***
  - Please limit your yelling
- All Free*

Session Topics

- Warnings as Errors
- Nullable Reference Types
- Static Code Analysis
- Codify Code Patterns
- Secure Supply Chain
- Continuous Testing

Recommendation:

Treat Warnings as Errors

- "Strict Mode"
- Don't compile if warning found
    - Ex: Not using method return
- Single Line in `*.csproj` file
- Later recommendations built on this

How to Enable Warnings as Error in CsProj

```xml
<PropertyGroup>
  <TreatWarningsAsErrors>true</TreatWarningsAsErrors>
<PropertyGroup>
```

Recommendation:

Embrace Nullable References Types (NRTs)

- Stop yelling at me!
    - They're good! I swear.
- Not as hard as Rust!
- A lot of code hassle can be mitigated

What are NRTs?

- Nullable Reference Types
- Compiler check for null value at Compile Time
- Compiler Warning when compiler thinks something ***can be*** null
- https://learn.microsoft.com/en-us/dotnet/csharp/nullable-references

```
<PropertyGroup>
  <Nullable>enable</Nullable>
<PropertyGroup>
```

Duende

Sample NRT Usage

```
public void SomeMethod()
{
  var userId = LoadUserId(_context);
  Console.WriteLine(userId);//Generates Compiler Warning
}

private string? LoadUserId(HttpContext? context)
{
  if(context is null)
  {
    return null;
  }

  return context.ParseUserId();
}
```

Duende

Use Attributes with NRTs for Runtime Help

- Attributes to help compiler for runtime checks
  - `NotNullWhenAttribute`
  - `MemberNotNullAttribute`
- Useful when deserializing objects
- Or, when things are weird

Check for Null at "Ingress of the App"

- Check all entities coming into your code
  - HTTP Requests
  - External Service Calls
  - Database Calls*
- Require data should be default
  - Not always possible, more thinking now (shift-left)

Nullable References Types Finds Bugs

- See Title

Recommendation:

Static Code Analysis

- Checks for common code issues
    - Sometimes fixes
- Enforce styling rules

External Tools You Can Install

- SonarSource / SonarQube / SonarCloud
- CodeMaid
- FxCop
- Rider / Resharper
- Visual Studio

Roslyn Analyzers

- Scan code at compile time
    - Built-in ones
    - Add with NuGets
    - Build your own
- May include code fixes

Built-In Roslyn Analyzers

```xml
<PropertyGroup>
  <EnableNETAnalyzers>true</EnableNETAnalyzers>
  <EnforceCodeStyleInBuild>true</EnforceCodeStyleInBuild>
  <EnableRequestDelegateGenerator>true</EnableRequestDelegateGenerator>
  <EnableConfigurationBindingGenerator>true</EnableConfigurationBindingGenerator>
</PropertyGroup>
```

- Note: `Microsoft.CodeAnalysis.NetAnalyzers` Package or `<EnableNETAnalyzers>` replaced FxCop

Add Analyzers with NuGet Packages

- `*.Analyzers` package
  - `XUnit.Analyzers`
  - `Roslynator.Analyzers`
  - `Microsoft.Azure.Functions.Worker.Sdk.Analyzers`
  - `SonarAnalyzer.CSharp`

Recommendation:

Manage all Static Rules in One Place

- .editorconfig file

`.editorconfig` File

- Extensible / Open Standard / Configurable / etc
- Single file checked into source control
- Different languages have different levels of support for it
    - C# support is really good
- https://editorconfig.org

```
# C# files
[*.cs]

#### Core EditorConfig Options ####

# Indentation and spacing
indent_size = 4
indent_style = space
tab_width = 4

# New line preferences
end_of_line = crlf
insert_final_newline = true

# Parentheses preferences
dotnet_style_parentheses_in_arithmetic_binary_operators = always_for_clarity:error
dotnet_style_parentheses_in_other_operators = never_if_unnecessary:silent

# Built-In Error Codes
csharp_style_deconstructed_variable_declaration = true:suggestion
dotnet_style_coalesce_expression = true:suggestion

# Specific Error Codes
dotnet_diagnostic.S3267.severity = suggestion
dotnet_diagnostic.SA1403.severity=error
```

It's okay to suppress rules in specific cases

- `GlobalSuppressions.cs` file
- Individual `SuppressMessage` attribute

Recommendation:

Codify Code Patterns

- Don't rely on humans to always do things right
- PRs only go so far

Custom Roslyn Analyzers

- Make your own
    - Specific to your projects
- Example:
    - All ASP.NET Controller Endpoints must include `[Authorize]` or `[AllowAnonymous]` attribute

Source Generators

- Add new code at compile time
- Already used by the .NET Team heavily for AoT stuff

Common Interface Example

```
public interface IUserManager
{
  ValueTask UpdateNameAsync(string name);
}

public class UserManager : IUserManager
{
  public async ValueTask UpdateNameAsync(string name)
  {
    await Task.CompletedTask;
  }
}

//Register with IoC Container
builder.Services.AddScoped<IUserManager, UserManager>();
```

Generated Interface Example

```
[GenerateSimpleInterface]
public class UserManager : IUserManager
{
  public async ValueTask UpdateNameAsync(string name)
  {
    await Task.CompletedTask;
  }
}

//Register with IoC Container
builder.Services.AddScoped<IUserManager, UserManager>();
```

Shameless Self Promotion

- Other ConFoo Session
  - Coding C# with C#
  - Friday: 11 AM

Recommendation:

Secure your Software Supply Chain

- Update Dependencies
  - Update nugets each sprint

Protect Against NuGet CVEs

```
<PropertyGroup>
  <NuGetAudit>true</NuGetAudit>
  <NuGetAuditMode>all</NuGetAuditMode>
  <NuGetAuditLevel>low</NuGetAuditLevel>
</PropertyGroup>
```

- Adds build warnings
    - https://learn.microsoft.com/en-us/nuget/concepts/auditing-packages

Recommendation:

Continuous Testing

- Continuous Testing Tools
    - NCrunch
    - Resharper*/Rider*
    - Visual Studio*
    - `dotnet watch`

* = certain licenses of that product

Review

- Warnings as Errors
- Nullable Reference Types
- Static Code Analysis
- Codify Patterns
- Secure Supply Chain
- Continuous Testing

In Depth Videos

- YouTube Series of these concepts, more in-depth
  - https://programmeral.com/posts/20250603_OpinionatedAspNetVideos
  - https://www.youtube.com/playlist?list=PLywNgcEGt3ofUBJsuriVX9A-93JPDi48A

Duende

Content

Feedback