**May 24-2025: This did not build**

# ASP.NET Core (.NET Framework)

# Build and test ASP.NET Core projects targeting the full .NET Framework.

# Add steps that publish symbols, save build artifacts, and more:

# https://docs.microsoft.com/azure/devops/pipelines/languages/dotnet-core

trigger:

branches:

include:

- master # Set to 'master' as your main branch for triggering pipeline

pool:

vmImage: 'windows-2022' # Use Windows-based build agent (Windows Server 2022)

variables:

solution: '\*\*/\*.sln' # Glob pattern to locate your solution files

buildPlatform: 'Any CPU'

buildConfiguration: 'Release'

stages:

- stage: Restore

displayName: 'Restore Dependencies'

jobs:

- job: Restore

displayName: 'Restore Dependencies'

steps:

- task: UseDotNet@2

inputs:

packageType: 'sdk'

version: '9.x' # Ensure this matches your target .NET SDK version (e.g., 6.x for .NET 6)

installationPath: $(Agent.ToolsDirectory)/dotnet

- task: DotNetCoreCLI@2

inputs:

command: 'restore'

restoreSolution: $(solution) # Restore dependencies for both the Web API and test project

- stage: Build

displayName: 'Build Projects'

jobs:

- job: Build

displayName: 'Build Projects'

steps:

- task: UseDotNet@2

inputs:

packageType: 'sdk'

version: '9.x'

installationPath: $(Agent.ToolsDirectory)/dotnet

- task: DotNetCoreCLI@2

inputs:

command: 'build'

projects: April3\_2025WEBAPI/April3\_2025WEBAPI.csproj # Build your Web API project

tests/April3\_2025WEB\_API\_TDD/April3\_2025WEB\_API\_TDD.csproj # Build your test project

- stage: Test

displayName: 'Run Tests'

jobs:

- job: Test

displayName: 'Run Unit Tests'

steps:

- task: UseDotNet@2

inputs:

packageType: 'sdk'

version: '9.x'

installationPath: $(Agent.ToolsDirectory)/dotnet

- task: DotNetCoreCLI@2

inputs:

command: 'restore'

projects: 'tests/April3\_2025WEB\_API\_TDD/April3\_2025WEB\_API\_TDD.csproj' # Ensure the test project dependencies are restored

- task: DotNetCoreCLI@2

inputs:

command: 'build'

projects: 'tests/April3\_2025WEB\_API\_TDD/April3\_2025WEB\_API\_TDD.csproj' # Build the test project

- task: DotNetCoreCLI@2

inputs:

command: 'test'

projects: 'tests/April3\_2025WEB\_API\_TDD/April3\_2025WEB\_API\_TDD.csproj' # Run tests from the MsTest project

arguments: '--configuration $(buildConfiguration) --no-build --logger trx' # --no-build skips rebuilding and --logger trx is for test results in TRX format

- stage: Publish

displayName: 'Publish Artifacts'

jobs:

- job: Publish

displayName: 'Publish Artifacts'

steps:

- task: DotNetCoreCLI@2

inputs:

command: 'publish'

projects: 'April3\_2025WEBAPI/April3\_2025WEBAPI.csproj' # Publish your Web API project (optional)

arguments: '--configuration $(buildConfiguration) --output $(Build.ArtifactStagingDirectory)'

- task: PublishBuildArtifacts@1

inputs:

PathtoPublish: $(Build.ArtifactStagingDirectory)

ArtifactName: 'drop'

publishLocation: 'Container'

|  |
| --- |
| **May 24-2025: This built but not sure if the tests ran** |
| # ASP.NET Core (.NET Framework)  # Build and test ASP.NET Core projects targeting the full .NET Framework.  # Add steps that publish symbols, save build artifacts, and more:  # https://docs.microsoft.com/azure/devops/pipelines/languages/dotnet-core  trigger:  branches:  include:  - master # Set to 'master' as your main branch for triggering pipeline  pool:  vmImage: 'windows-latest' # Use Windows-based build agent (Windows Server 2022)  variables:  solution: '\*\*/April3\_2025WEBAPI.sln' # Glob pattern to locate your solution files  project: '\*\*/April3\_2025WEBAPI.csproj'  tests: '\*\*/April3\_2025WEBAPI\_TDD.csproj'  buildPlatform: 'Any CPU'  buildConfiguration: 'Release'  steps:  - task: NuGetToolInstaller@1  name: 'NugetToolInstaller'  - task: NuGetCommand@2  name: 'RestoreNuget'  inputs:  command: 'restore'  restoreSolution: '$(solution)'  feedsToUse: select'    - task: DotNetCoreCLI@2  name: 'TDDTest'  inputs:  command: 'test'  projects: '$(tests)'  arguments: '--configuration $(buildConfiguration)'  - task: DotNetCoreCLI@2  name: 'buildArtifact'  inputs:  command: 'publish'  publishWebProjects: false  projects: '$(project)'  arguments: '--configuration $(buildConfiguration) --output $(Build.ArtifactStagingDirectory)'  zipAfterPublish: true  - task: PublishBuildArtifacts@1  name: 'publishArtifact'  inputs:  PathtoPublish: '$(Build.ArtifactStagingDirectory)'  ArtifactName: 'April3\_2025WEBAPI\_artifact'  publishLocation: 'Container' |