PowerShell Conference Europe

Pester 5 vs Pester 4

Jakub Jareš

@nohwnd

20 - 23 June 2022

Raffle

https://forms.office.com/r/TZM7h7JWn7

@nohwnd

PSCONF.EU VIENNA 20 - 23 June 2022

Many thanks to our sponsors:

Jakub Jareš

PSCONF.EU VIENNA 20 - 23 June 2022

Pester owner and maintainer. Senior software engineer, developing VSTest and MSTest at Microsoft. All opinions are mine.

@nohwnd
@pspester
me@jakubjares.com

Please consider sponsoring my open-source development:

[Sponsor @nohwnd on GitHub Sponsors]

Why stop using Pester 4?

@nohwnd

Why stop using Pester 4?

- It hasn't seen release for few years (4.10.0, 07/2020).
- There won't be any new development.
- It is a bit slow.
- Code coverage is really slow.
- Interactive UI for tests in VSCode won't work with Pester 4.

Why Pester 5?

Why Pester 5?

- It has active development and support.
  - Twitter: @FrodeFlaten
  - Github: fflaten
- It has new features.
- It is optimized for speed.
- It has new faster code coverage.
- It works with interactive test UI in VS code.

Raffle

https://forms.office.com/r/TZM7h7JWn7

@nohwnd

The biggest change.

Discovery, and run

@nohwnd

```
BeforeAll {
  Import-Module $PSScriptRoot\Planets.psd1
}
Describe "Get-Planets" {
  BeforeAll {
    Update-PlanetHelp
  }
  It "Get first planet returns Mars" {
     (Get-Planet)[0] | Should -Be "Mars"
  }
  AfterAll {
    Clean-PlanetHelpCache
  }
}
```

Discovery

```
BeforeAll {
  Import-Module $PSScriptRoot\Planets.psd1
}
Describe "Get-Planets" {
  BeforeAll {
    Update-PlanetHelp
  }
  It "Get first planet returns Mars" {
     (Get-Planet)[0] | Should -Be "Mars"
  }
  AfterAll {
    Clean-PlanetHelpCache
  }
}
```

```
BeforeAll {
  Import-Module $PSScriptRoot\Planets.psd1
}
Describe "Get-Planets" {
  BeforeAll {
    Update-PlanetHelp
  }
  It "Get first planet returns Mars" {
     (Get-Planet)[0] | Should -Be "Mars"
  }
  AfterAll {
    Clean-PlanetHelpCache
  }
}
```

Run

```
"Get-Planets"
"Get first planet returns Mars"
```

```
{
  Import-Module $PSScriptRoot\Planets.psd1
}
{
  Update-PlanetHelp
}
{
  (Get-Planet)[0] | Should -Be "Mars"
}
{
  Clean-PlanetHelpCache
}
```

Discovery

Run

What exactly runs in Discovery?

- All code in Discovery, that is outside of BeforeAll, BeforeEach, It, AfterEach and AfterAll.
- This includes:
  - TestCases
  - variables in test names
  - all code that you place into Describe

What exactly runs in Run?

- Code in:
  - BeforeAll
  - BeforeEach
  - It
  - AfterEach
  - AfterAll

- Run happens in a new **empty** script scope.

Demo 1

Discovery and Run

PSCONF.EU VIENNA 20 - 23 June 2022

@nohwnd

Patterns & common problems

@nohwnd

BeforeAll

for script setup

@nohwnd

NO:

```
. $PSScriptRoot\Get-Emoji.ps1
$script:name = "avocado"

Describe "Get-Emoji" {
   ...
}
```

YES:

```
BeforeAll {
   . $PSScriptRoot\Get-Emoji.ps1
   $script:name = "avocado"
}

Describe "Get-Emoji" {
   ...
}
```

NO:

```
$here = Split-Path $MyInvocation.MyCommand.Path
$sut = (Split-Path -Leaf
$MyInvocation.MyCommand.Path).Replace(".Tests.", ".")
. "$here\$sut"
```

YES:

```
. $PSCommandPath.Replace(".Tests.", ".")


. $PSScriptRoot\Get-Avocado.ps1
```

Demo 2

Importing scripts

in BeforeAll

@nohwnd

BeforeDiscovery

@nohwnd

NO:

```
$testCases = Import-Csv testcases1.csv
Describe "Get-Avocado" {
  It "Lists the correct emoji" -TestCases $testCases {
  }
}
```

YES:

```
BeforeDiscovery {
  $testCases = Import-Csv testcases1.csv
}
Describe "Get-Avocado" {
  It "Lists the correct emoji" -TestCases $testCases {
  }
}
```

TestCases

NO:

```
BeforeAll {
  $testCases = Import-Csv testcases1.csv
}
Describe "Get-Avocado" {
  It "Lists the correct emoji" -TestCases $testCases { }
}
```

YES:

```
BeforeDiscovery {
  $testCases = Import-Csv testcases1.csv
}
Describe "Get-Avocado" {
  It "Lists the correct emoji" -TestCases $testCases { }
}
```

PSCONF.EU
VIENNA
20 - 23 June 2022

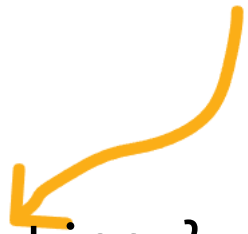
Variables and scoping

@nohwnd

NO:

```
$fileLocation = "C:\data\input.csv"
Describe "Get-Avocado" {
  It "Gets and avocado" {
      $content = Import-Csv $fileLocation
  }
}
```

TestCases on Describe



YES:

```
BeforeDiscovery { $fileLocation = "C:\data\input.csv" }
Describe "Get-Avocado" -ForEach @{ fileLocation = $fileLocation } {
  It "Gets and avocado" {
      $content = Import-Csv $fileLocation
  }
}
```

PSCONF.EU
VIENNA
20 - 23 June 2022

Variables in names

And advanced <> templates

@nohwnd

NO:

```
BeforeAll { $fileLocation = "C:\data\input.csv" }
Describe "Get-Avocado" {
  It "Gets and avocado from $fileLocation" {
      $content = Import-Csv $fileLocation
  }
}
```

YES:

```
BeforeAll { $fileLocation = "C:\data\input.csv" }
Describe "Get-Avocado" {
  It "Gets and avocado from <fileLocation>" {
      $content = Import-Csv $fileLocation
  }
}
```

```
BeforeAll {
  $user = [PsCustomObject]@{
    Name = "Jakub"
    Age  = 34
  }
}
Describe "d" {
  It "User: <user.name>, <user.age> " {

  }
}
```

```
[+] User: Jakub, 34
```

InModuleScope

@nohwnd

NO:

```
InModuleScope Avocado {
  Describe "Get-AvocadoInternal" {
    It "Gets and avocado" {
      $avocado = Get-AvocadoInternal
    }
} }
```

YES:

```
Describe "Get-AvocadoInternal" {
  It "Gets and avocado" {
    InModuleScope Avocado {
      $avocado = Get-AvocadoInternal
    }
} }
```

PSCONF.EU
VIENNA
20 - 23 June 2022

Foreach

@nohwnd

NO:

```
$testCases = Import-Csv testcases1.csv
foreach ($_ in $testCases) {
  It "test case: $_ $($_.name)" { $_ }
}
```

YES:

```
$testCases = Import-Csv testcases1.csv
It "test case: <_> <_.name>" -ForEach $testCases { $_ }
```

```
[+] @{Name=Jakub} Jakub
```

Parameters & Configuration

v4 compatibility parameter set

```
Invoke-Pester
  -Path "."
  -TagFilter -ExcludeTagFilter
  -FullNameFilter
  -EnableExit
  -PassThru
  -CodeCoverage -CodeCoverageOutputFile etc.
  -OutputFile
  -Show
  -Quiet
```

Simple parameter set

```
Invoke-Pester
  -Path "." -ExcludePath
  -TagFilter -ExcludeTagFilter
  -FullNameFilter -ExcludeFullNameFilter
  -Output Diagnostic, Detailed, Normal, Minimal, None
  -PassThru
  -CI (output xml, coverage, and exit on error)
  -Container (provide scriptblock, or ps1 + data)
```

Advanced parameter set

```
$configuration = New-PesterConfiguration
-Configuration $configuration
```

DEPRECATED:

```
Invoke-Pester -Path . -OutputFile tests.xml -Show All -EnableExit
```

YES:

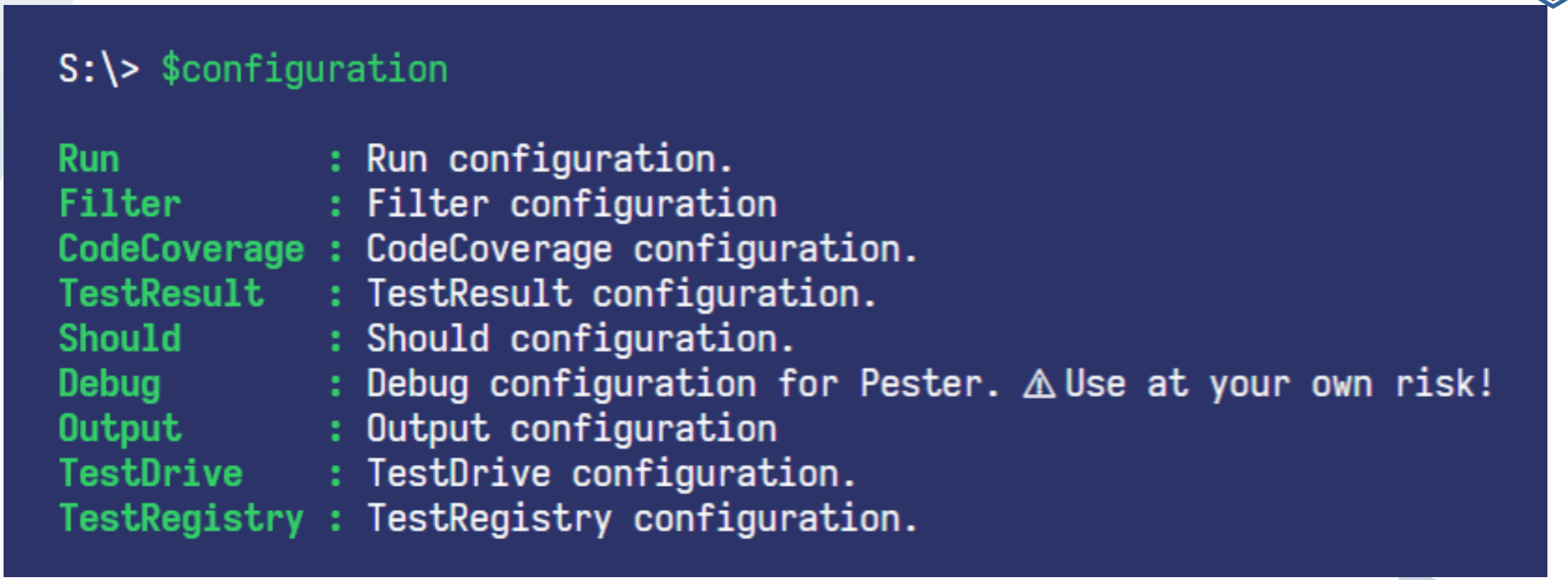
```
$configuration = New-PesterConfiguration
$configuration.Run.Path = "."
$configuration.Run.Exit = $true

$configuration.TestResult.Enabled = $true
$configuration.TestResult.OutputPath = "tests.xml"

$configuration.Output.Verbosity = "Detailed"

Invoke-Pester -Configuration $configuration
```

$configuration = New-PesterConfiguration



Demos

Configuration object

(if there is time).

Summary

- Understand the difference of Discovery and Run.
- Pass data between them using ForEach (TestCases).

Pick a winner!

Q&A

15 minutes

Ask me **here** about Pester.

Ask me **outside** about:

Pester, testing, C#, .NET, dotnet, dotnet test, MSTest, Profiler, PerfView, coffee, beer etc.

@nohwnd