

## ZADANIA w tle (job)

Zadania w tle (jobs) w PowerShell pozwalają na wykonywanie długotrwałych lub asynchronicznych operacji w tle, bez blokowania wiersza poleceń. Zadania te są wykonywane w osobnym wątku, co pozwala na kontynuowanie pracy w sesji PowerShell podczas ich działania.

Komenda **Start-Job** jest używana do uruchamiania zadań w tle. Przyjmuje blok skryptu (**ScriptBlock**), który określa operacje do wykonania w tle. Przykład:

```
$job = Start-Job -ScriptBlock { ... }  
  
ScriptBlock {Write-Host "zadanie w tle" }  
  
ScriptBlock { param($tekst) Write-Host $tekst} -ArgumentList "ALA ma"
```

**Get-Job** wyświetla listę wszystkich zadań w tle, umożliwiając sprawdzenie ich stanu (czy są w toku, zakończone itp.).

**Stop-Job** służy do zatrzymywania działających zadań w tle.

```
Stop-Job -Id <IdentyfikatorZadania>
```

**Remove-Job** usuwa zadanie z listy zadań. Można go użyć po zakończeniu lub zatrzymaniu zadania, aby zwolnić zasoby.

```
Remove-Job -Id <IdentyfikatorZadania>
```

**Receive-Job** pobiera wyniki z wykonanego zadania. Wyniki są buforowane i mogą być pobierane w dowolnym momencie po zakończeniu zadania.

```
Receive-Job -Id <IdentyfikatorZadania>
```

**Wait-Job** wstrzymuje wykonywanie bieżącego skryptu do momentu zakończenia wskazanego zadania w tle. Jest to użyteczne, gdy skrypt zależy od wyników zadania w tle i musi poczekać na jego zakończenie przed kontynuacją.

```
Wait-Job -Id <IdentyfikatorZadania>
```

**Suspend-Job** pozwala tymczasowo wstrzymać wykonanie zadania w tle. Zadanie pozostaje w pamięci, ale jego wykonanie jest zatrzymane, dopóki nie zostanie wznowione. Jest to przydatne, gdy potrzebujesz zwolnić zasoby systemowe na inny proces bez całkowitego zatrzymywania i usuwania zadania. Użycie:

```
Suspend-Job -Id <IdentyfikatorZadania>
```

**Resume-Job** jest używany do wznowienia wykonania zadania, które zostało wcześniej wstrzymane za pomocą **Suspend-Job**. Ta komenda pozwala na kontynuację procesu od miejsca, w którym został wstrzymany. Użycie:

```
Resume-Job -Id <IdentyfikatorZadania>
```

## Przykłady Użycia

```
$job1 = Start-Job -ScriptBlock { 1..10 | Measure-Object -Sum }
$job3 = Start-Job -ScriptBlock { Start-Sleep 10; "Zakończono" }

# Utworzenie zadania w tle
$job = Start-Job -ScriptBlock
{
    Start-Sleep -Seconds 5
    "Zadanie zakończone"
}

# Wyświetlenie informacji, że główny skrypt czeka na zakończenie zadania
Write-Host "Czekam na zakończenie zadania w tle..."

# Użycie Wait-Job do wstrzymania skryptu do czasu zakończenia zadania
Wait-Job -Job $job

# Odbiór i wyświetlenie wyników zadania
$result = Receive-Job -Job $job
Write-Host "Wynik zadania w tle: $result"

# Usunięcie zadania
Remove-Job -Job $job
```