I’ll Get Back to You

Task, Await, and Asynchronous Programming in C#

Jeremy Clark
Developer Betterer
jeremybytes.com

Level: Introductory / Intermediate

Session Survey

- Your feedback is very important to us
- Please take a moment to complete the session survey found in the mobile app
- Use the QR code or search for “Converge360 Events” in your app store
- Find this session on the Agenda tab
- Click “Session Evaluation”
- Thank you!

30th ANNIVERSARY VSLive! 1993 - 2023

Asynchronous Patterns

- Asynchronous Programming Model (APM)
- Event Asynchronous Pattern (EAP)
- Task Asynchronous Pattern (TAP)

Asynchronous Programming Model (APM)

- Method-Based
- Methods

```
BeginGetData()
EndGetData()
```

- IAsyncResult

Event Asynchronous Pattern (EAP)

- Method/Event-Based
- Method

  `GetDataAsync()`
- Event

  `GetDataCompleted`

  Results in EventArgs

Task Asynchronous Pattern (TAP)

- Task-Based
- Method Returns a Task

```
Task<T> GetDataAsync()
```

- Task
  - Represents a concurrent operation
  - May or may not operate on a separate thread
  - Can be chained and combined

async & await

- Syntactic Wrapper Around Task
  - “await” pauses the current method until Task is complete.
  - Looks like a blocking operation
  - Does not block current thread
- “async” is just a Hint
  - Does not make a method run asynchronously
  - Tells the compiler to treat “await” as noted above

Task Properties (.NET 6 / 7)

- Task Properties
  - IsFaulted
  - IsCanceled
  - IsCompleted*
  - IsCompletedSuccessfully

- IsCompletedSuccessfully
  - .NET 6 / 7
  - .NET Standard 2.1
  - NOT .NET Standard 2.0
  - NOT .NET Framework

*Note: Means “no longer running” not “completed successfully”

Task Properties (.NET Framework)

- Task Properties
  - IsFaulted
  - IsCanceled
  - IsCompleted*
  - Status

*Note: Means “no longer running” not “completed successfully”

- TaskStatus
  - ***Canceled***
  - Created
  - ***Faulted***
  - ***RanToCompletion***
  - Running
  - WaitingForActivation
  - WaitingForChildrenToComplete
  - WaitingToRun

Exception Handling

- AggregateException
  - Tree structure

- Flatten()
  - Flattens the tree structure to a single level

Cancellation

- CancellationToken is ReadOnly

```
new CancellationToken(false)
new CancellationToken(true)
```

- CancellationTokenSource

```
var cts = new CancellationTokenSource()
var token = cts.Token
cts.Cancel()
```

Thank You!

Jeremy Clark

- www.jeremybytes.com
- jeremy@jeremybytes.com
- @jeremybytes

https://github.com/jeremybytes/vslive2023-microsofthq

Session Survey

- Your feedback is very important to us
- Please take a moment to complete the session survey found in the mobile app
- Use the QR code or search for “Converge360 Events” in your app store
- Find this session on the Agenda tab
- Click “Session Evaluation”
- Thank you!