Sela.

AsyncWS

Beyond Async and Await

college@sela.co.il 03-6176666

Beyond Async and Await

AsyncWS - Version: 1

2 days Course

Description:

This is a 2-days workshop focusing on async and await theory and practice. You will gain deep understanding of async methods, how to use them right, best practices, and useful patterns. Although they seem easy on the surface, async methods and the await keyword are full of tricks and pitfalls that you need to remember in order to write efficient asynchronous code. This day is designed for developers currently using async and await for their systems, and also for developers who have experience with asynchronous programming using the TPL or other threading libraries.

Intended audience:

Experienced .NET Developers, Team Leader and Architects

Prerequisites:

Developers / Team Leaders who have some experience with asynchronous programming using TPL.

Objectives:

Getting familiar with advance async design pattern and be aware of the pitfall.
Enable developers and team leader getting more from async programming.

Topics:

Intro

- Amdahl’s Law

    - Thread vs ThreadPool

Task in deep (from different angle)

- Task as data structure
- Semantical Task

    - Thread Pool and Task

Demystify async and await

- async syntax
- What does it do? (and don’t)

º Chaining await

º Simultaneous await

º When All (fork join pattern)

º void async?

º Async Download

º Implicit Parallelism (pitfall)

Await loop

- Sequential

- Await LINQ Pattern
- TPL Dataflow Patterns

º Task.Run vs Task.Factory.StartNew

Error Handling

- Catching aggregate exceptions
- Read-able exceptions
- Debugger tricks

Cancellation

- Timeouts
- Other Cancellation APIs
- Lazy Cancellation

Useful Extensions

- Deadlock detection
- Cancellation (safe)

º IO Completion Port

TAP: Task-based Asynchronous Pattern

- TAP with WCF
- TAP with Web API

ConfigureAwait

- Dead-locks (ASP.NET, WPF)
- Parallel-forking
- Scheduler

º Static Code Analyzers

API Design

- Initialization pitfall
- Initialization Pattern
- API Design
- Common Service Mistakes
- Open Files Mistakes

º Async Call Context

º Async Critical Section

TPL Dataflow

- Play with Fork Join (Transform Block)
- Batch Block (non-greedy)

Behind async await

- Awaitable

º C# 8 new async features

º Summary

º Exercise

Appendix

- Complex Async Routing
- Design API (delegates)