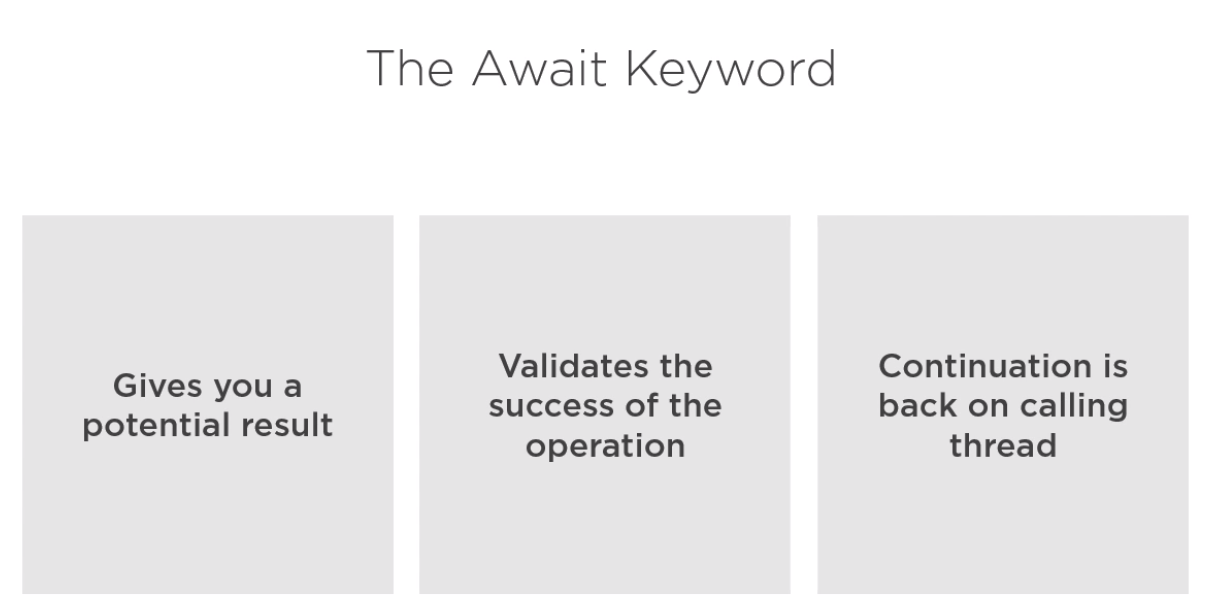
Async -Await

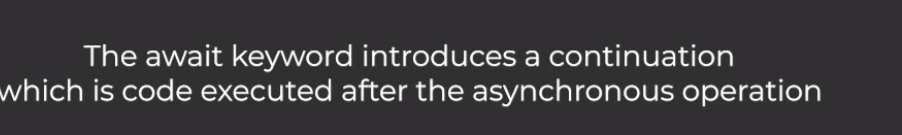
Await

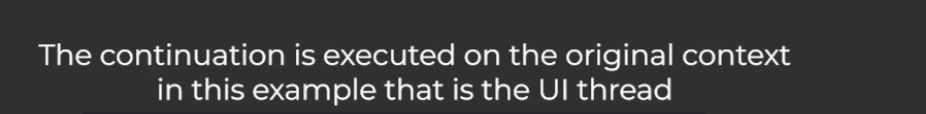


Continuation

The code execution after asynchronous operation complete the await

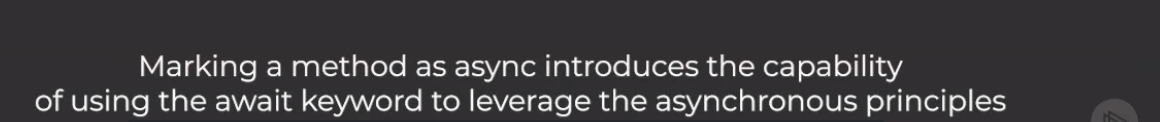


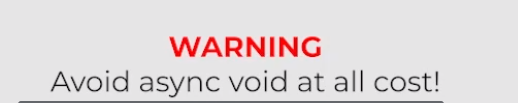




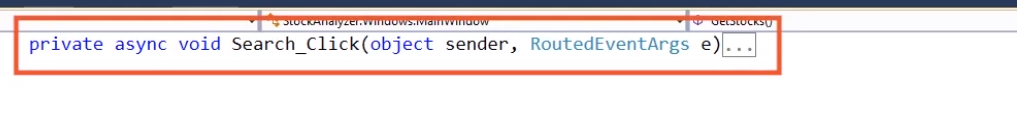


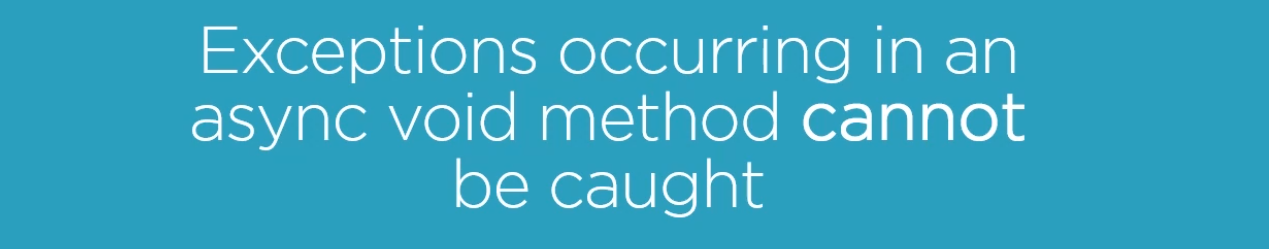
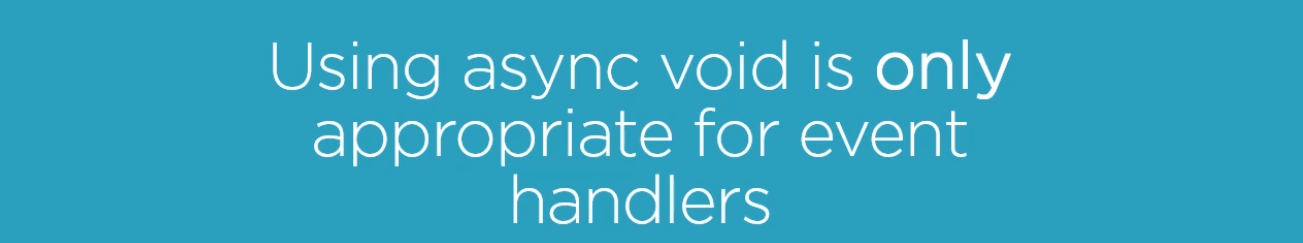
Async



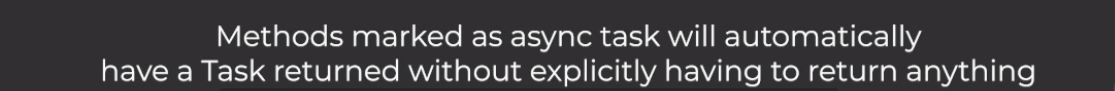


It only use for event handler





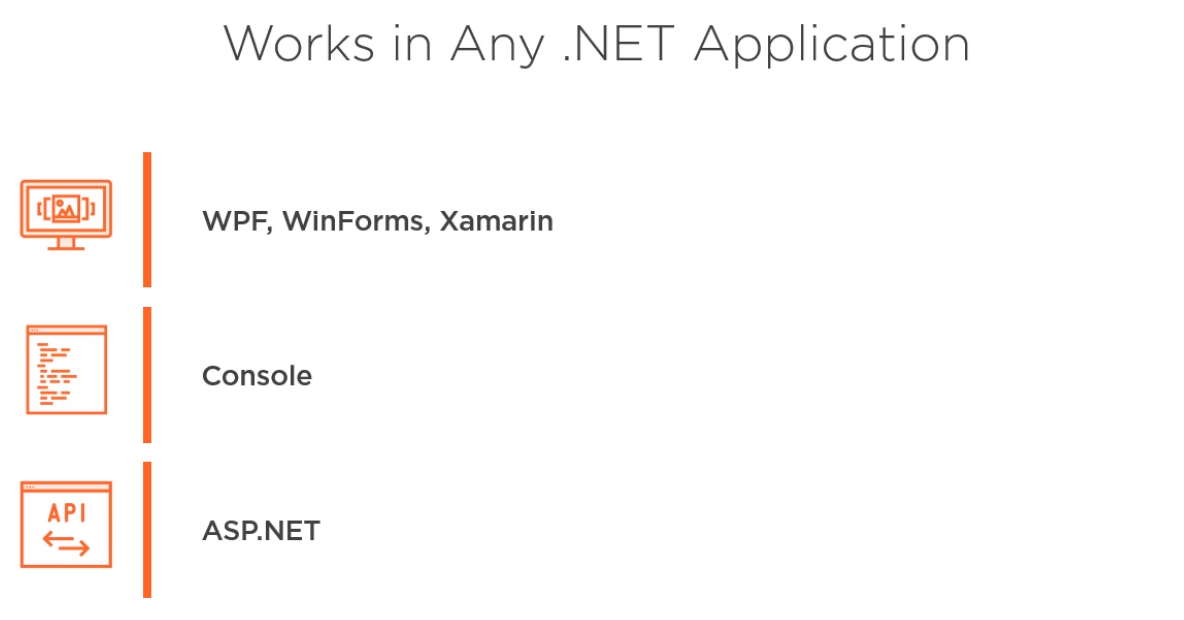
Return Task

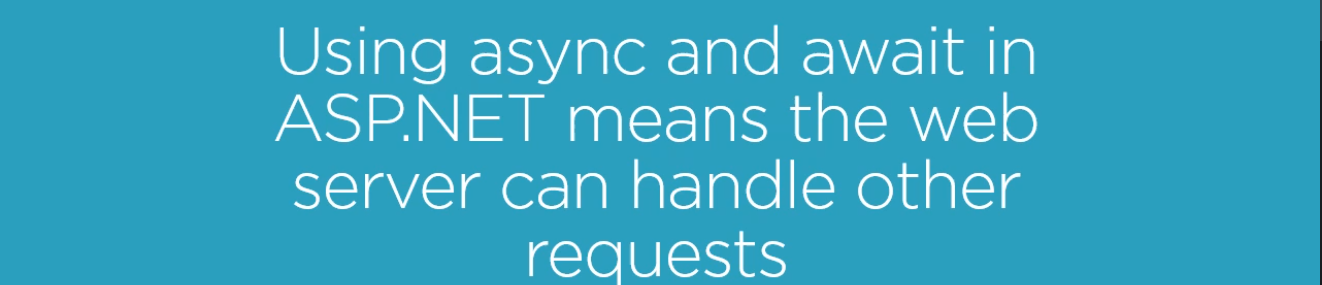


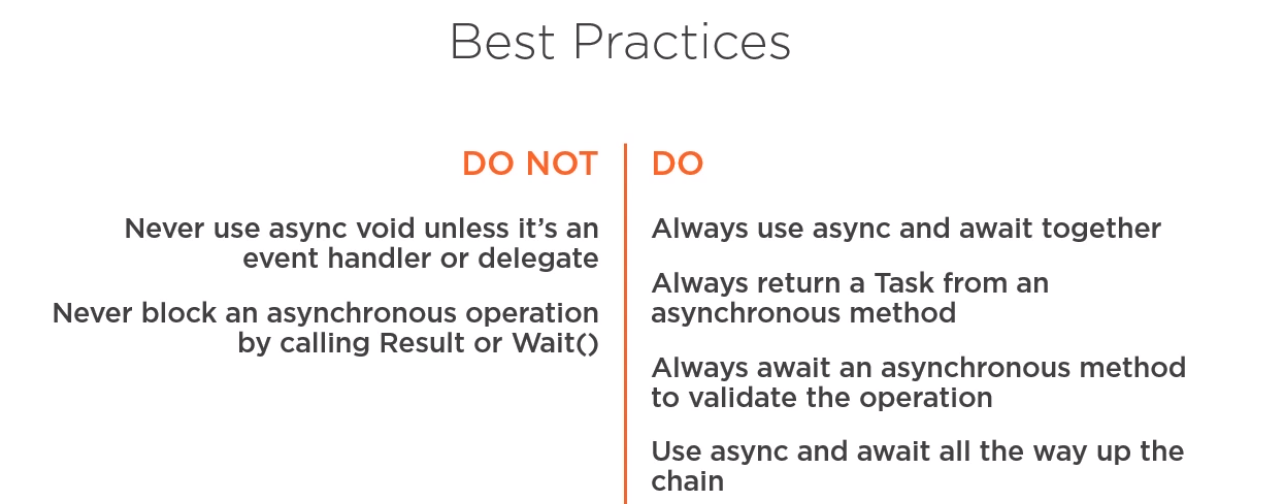
Compiler generate the return

It getting a reference of ongoing task

BEST Practice

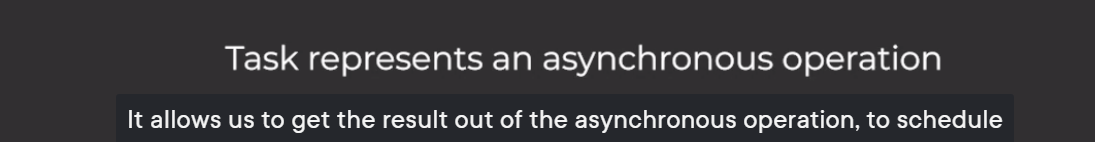




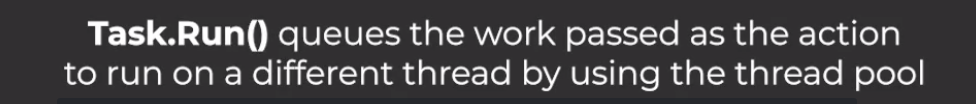


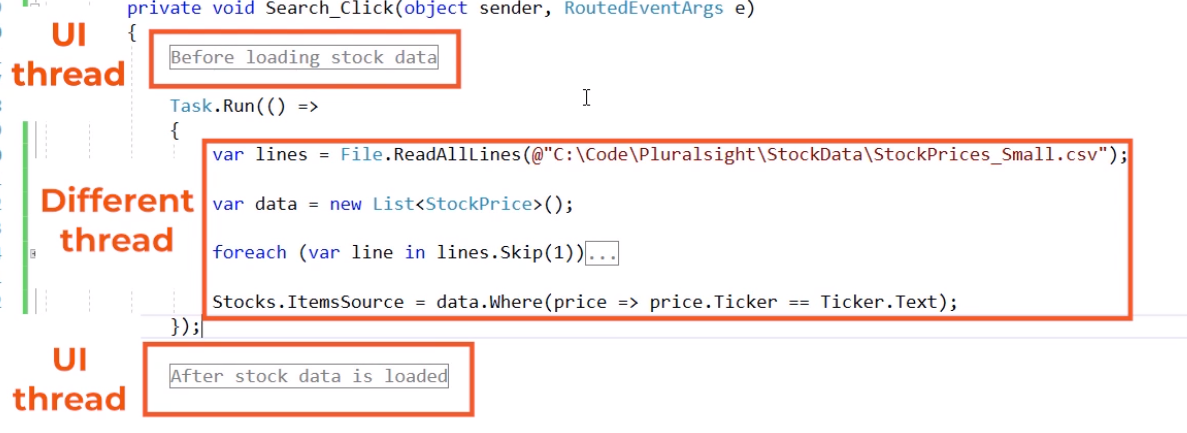
Using the Task Parallel Library in .NET

Task:

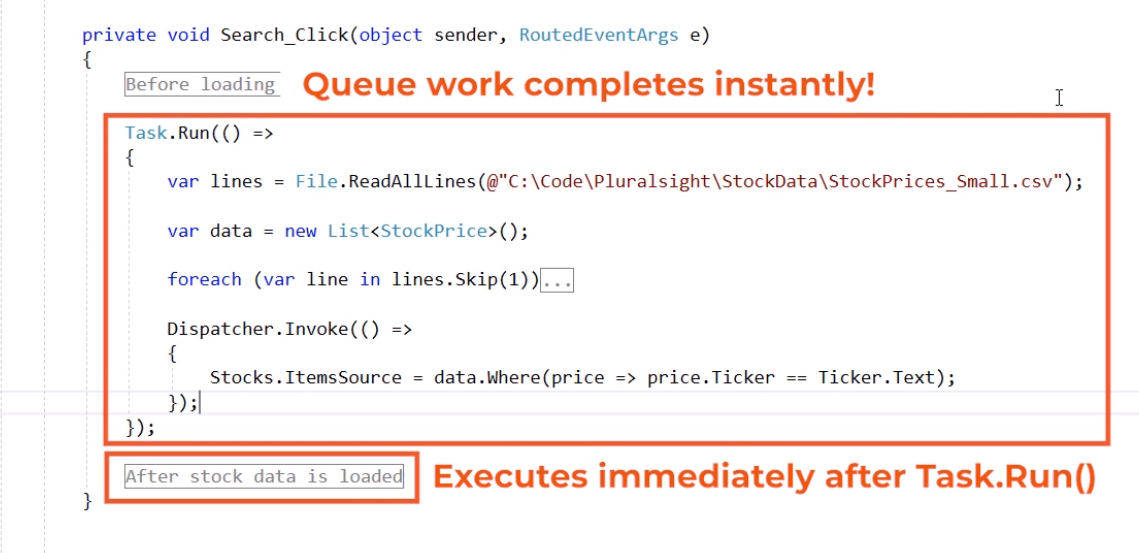


Use to schedule work one the asynchronies operation done and if there is problem it cancel or continue.

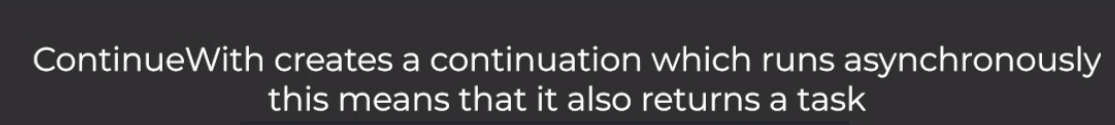


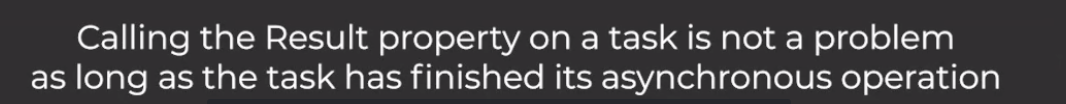


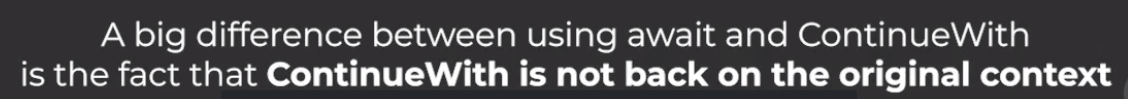
ACCESS recourse in different thread



ContinueWith

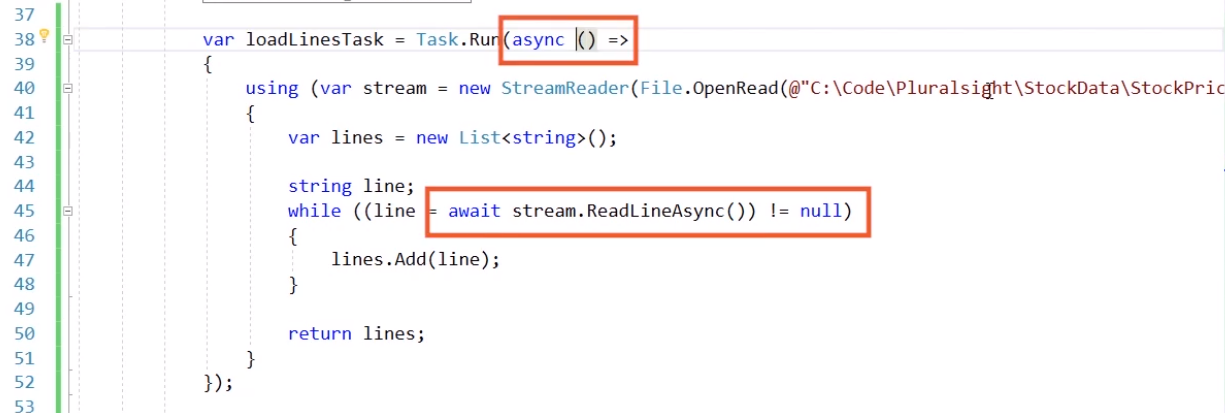




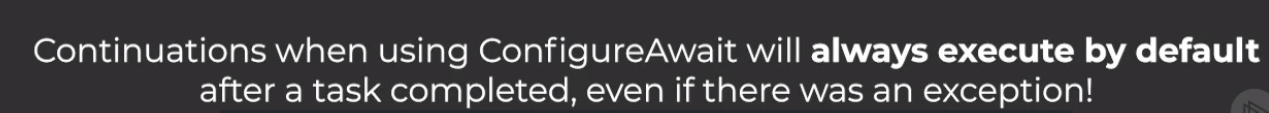


Async & Await inside task

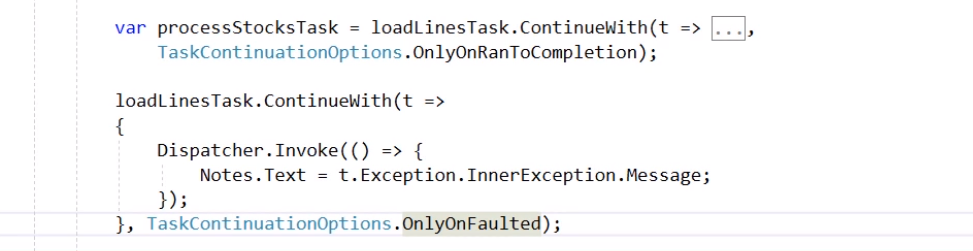
Can use async and await with task

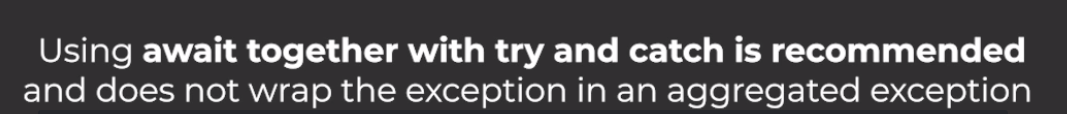


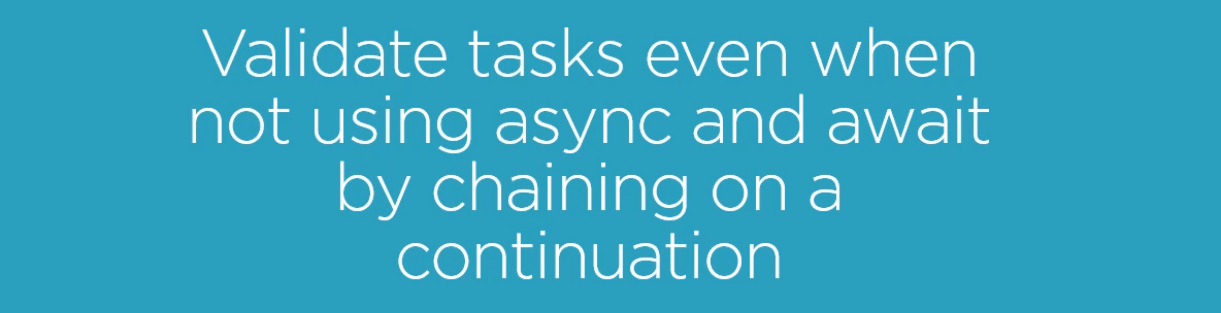
Error handle in Task & continueWith



Using TaskContinuationOptions we can chose when this contieation execute







Cancel task