Testing and Debugging Effectively

Mark Heath
SOFTWARE DEVELOPER

@mark_heath   www.markheath.net

Overview

What happens when things go wrong?

How can we debug LINQ pipelines?

How can we unit test LINQ pipelines?

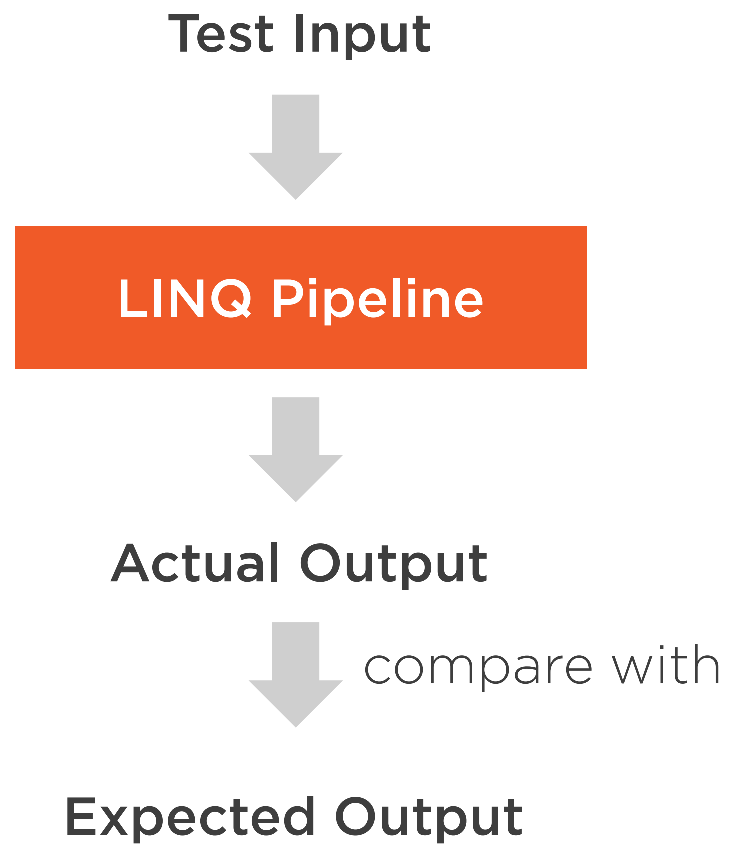
How can we handle exceptions?

- Can we suppress them and carry on?

Debugging LINQ Queries

Testing LINQ Queries

Unit Testing LINQ Pipelines



Testing with Side-effects

“Pure” functions have no “side effects”

- e.g. accessing disk, network, global state
- Same input always results in same output

How can we test LINQ pipelines with side effects?

- Make use of “mocks”

Testing LINQ to Entities

Test against a “real” database or in-memory?

- It is possible to mock a DbSet<T>

LINQ to Entities != LINQ to Objects

- e.g. String.Contains case sensitivity
- e.g. “Include” method

Consider creating “integration” tests

- Run against a real pre-seeded database
- Can be used for performance testing

Exception Handling in LINQ Queries

Demo

Suppressing Errors in LINQ Pipelines

Summary

Debugging LINQ

- Visual Studio lets you step into lambdas
- Use a “Peek” extension to inspect pipeline at intermediate stages

Testing LINQ

- Mock out methods with side effects
- Test against a real database

Handling Exceptions

Up Next: Functional Programming