



Module 2 Day 7

Integration Testing

DAO, continued

- AppSettings.json
 - DAO often relies on outside resources (DBMS), thus *integration*
- Re-factor of Country DAO
- Inserting, updating and deleting rows



Let's
Code

Types of Testing

- Unit Testing
 - Verifies parts of an application independently from other parts (isolated)
 - Narrow scope (focused)
 - Tests many cases (deep)
 - Fast!
- Integration Testing
 - Verifies interactions between multiple components
 - Broader scope
 - More difficult to test thoroughly
- End-to-end Testing (E2E)
 - Verifies interaction from end-user through a transaction and back to end-user
 - Broadest scope
 - Most difficult to test all possible combinations of scenarios
 - Slowest to run (for both setup and execution)

DAO Integration Testing

- DAO testing is an example of an Integration Test
 - DAO often relies on outside resources (DBMS), thus *integration*
- Any type of test should be:
 - *Repeatable*: If the test passes/fails on first execution, it should pass/fail on second execution if no code has changed.
 - *Independent*: A test should be able to be run on its own, independent of other tests.
 - *Obvious*: When a test fails, it should be as obvious as possible why it failed.

Test Data Approaches

- *Shared Test Database (remote)*
 - *Advantages:*
 - Easy setup, often already exists
 - Production-like software and (possibly) hardware
 - *Disadvantages:*
 - Unreliable and brittle
 - Lack of test isolation
 - Temptation to rely on existing data (which can change)
- *Local Test Database*
 - *Advantages:*
 - Production-like software
 - Reliable (local control)
 - Isolation
 - *Disadvantages:*
 - Requires local hardware resources
 - RDBMS needs to be installed and managed

Testing DAOs

- Test Select, Insert, Update and Delete to verify the expected change
 - Existence or non-existence of rows
 - RowsAffected meets expectations
- Insert a set of test data so we know our starting state
- Next test should run with clean, known data
- This can be done in a few ways
 - We are going to use Transactions for easy cleanup



Let's
Code