Unit Testing for an Exception

Sometimes, a unit test needs to ensure a particular exception type occurs.

Here’s how to check that a particular exception and message occurs.

The catch is that, when executing code inside a unit test framework, such as MSUnit, any exception thrown by code is wrapped in an outer exception by the unit framework.

So, the desired exception must be unwrapped, for interrogation.

The following is a method of ensuring a particular exception type occurs, with a particular message:

```
try
{
    // Execute the code that will return the exception...
    var pl =
NETCore_GenericRepository_Base.QueryHelpers.PaginatedList<GenericRepositoryEF_Tests.TestClass>
                .CreateAsync(query, 1, 0).Result;

    Assert.Fail("An exception should have been thrown.");
}
catch (Exception e)
{
    // Since we are running inside unit testing, our exception type will be an inner exception of the exception we
receive.

    // Get the inner exception...
    var ee = e.InnerException;

    // Check its type...
    var etn = ee.GetType().Name;
    if (etn != "BusinessRuleBrokenException")
        Assert.Fail(string.Format("Unexpected exception of type {0} caught: {1}",
                e.GetType(), e.Message));

    // Check the exception message...
```

```
    if(ee.Message != "pageSize invalid. Must be positive.")
        Assert.Fail("Incorrect exception messag.");
}
```

Revision #1
Created 11 May 2025 06:44:16 by glwhite
Updated 11 May 2025 06:47:01 by glwhite