

Simplified Testing with EF Core's InMemory Provider



Julie Lerman

MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK

@julielerman thedatafarm.com



Overview



Testing with the InMemory provider in a full .NET solution

Dynamically switching providers and connections

Testing in a .NET Core solution

Using the InMemory provider to test logic that uses EF Core



Testing with the New InMemory Provider



Database Integration Testing with EF Core

Test project should have a connection string pointing to a test database

Create fresh database each time tests are run

EF Core methods `DbContext.Database.EnsureDeleted` + `EnsureCreated`



Refactoring the WPF Solution to Dynamically Use the Correct Provider



Testing in the ASP.NET Core Solution



Using InMemory to Test Methods That Use EF Core




Review

Full .NET

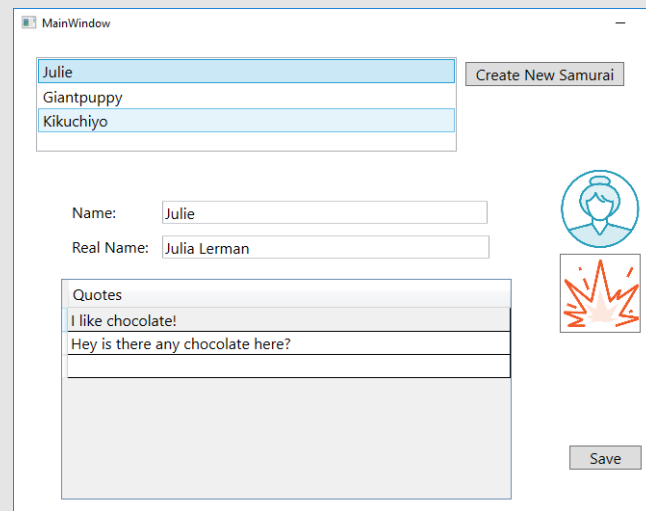
```
[TestMethod]
public void CanInsertSamuraiWithSaveChanges() {
    var optionsBuilder=new DbContextOptionsBuilder();
    optionsBuilder.UseInMemoryDatabase();
    using (var context = new SamuraiContext(optionsBuilder.Options)) {
        var samurai = new Samurai();
        samurai.Name = "Julie";
        context.Samurais.Add(samurai);
        context.SaveChanges();
    }
    using (var context2 = new SamuraiContext(optionsBuilder.Options)) {
        Assert.AreEqual(1, context2.Samurais.Count());
    }
}
```

Test Name: CanInsertSamuraiIntoDatabase

Test Outcome:  Passed

Standard Output

Debug Trace:
Default Samurai Id 0
EF Default Samurai Id -2147482647
DB assigned Samurai Id 1



.NET Core

```
{
  "version": "1.0.0-*",
  "testRunner": "mstest",
  "dependencies": {
    "Microsoft.NETCore.App": {
      "version": "1.1.0",
      "type": "platform"
    },
    "SamuraiAppCore.Data": "2.0.0-*",
    "SamuraiAppCore.Domain": "2.0.0-*",
    "Microsoft.EntityFrameworkCore.InMemory": "1.1.0",
    "MSTest.TestFramework": "1.0.8-rc",
    "dotnet-test-mstest": "1.1.2-preview"
  },
  "frameworks": {
    "netcoreapp1.0": {
      "imports": "dnxcore50"
    }
  }
}
```

```
public SamuraiModelTests() {
    _options = new DbContextOptionsBuilder<SamuraiContext>()
        .UseInMemoryDatabase().Options;
    SeedInMemoryStore();
}

[TestMethod]
public void CanRetrieveListOfSamuraiValues() {
    var context = new SamuraiContext(_options);
    var repo = new DisconnectedData(context);
    var list = repo.GetSamuraiReferenceList();
    Assert.IsInstanceOfType(list,
        typeof(List<KeyValuePair<int, string>>));
}
```



Resources

Entity Framework Core on GitHub github.com/aspnet/entityframework

EF Core Documentation docs.microsoft.com/ef

Unit testing with MSTest & .NET Core on Windows, using Visual Studio 2015
docs.microsoft.com/en-us/dotnet/articles/core/testing/using-mstest-on-windows

Relevant Pluralsight Courses

Automated Testing for Fraidy Cats Like Me bit.ly/PS-Testing

Entity Framework in the Enterprise bit.ly/PS_EFEnt



Simplified Testing with EF Core's InMemory Provider



Julie Lerman

MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK

@julielerman thedatafarm.com

