

DbContext for SQL Trace

 Entity-ORM SQL

```
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Logging; //Nuget .
using Microsoft.Extensions.Logging.Console;
using Microsoft.Extensions.Logging.Debug;

namespace accountapi.Config
{
    public class LogSettings
    {
        public static readonly LoggerFactory ConsoleLogger = new LoggerFactory(new[] {
            new ConsoleLoggerProvider((category, level) => category == DbLoggerCategory.Database.Command.Name
                && level == LogLevel.Debug, true)
        });

        public static readonly LoggerFactory DebugLogger = new LoggerFactory(new[] {
            new DebugLoggerProvider(_ , LogLevel) => LogLevel >= LogLevel.Debug)
        });
    }
}
```

Microsoft.Extensions.Logging
ConsoleLogger DebugLogger Xunit DebugLogger .
Xunit DebugLogger , APP ConsoleLogger

DbContext

```
var builder = new DbContextOptionsBuilder<AccountContent>()
    .UseLoggerFactory(LogSettings.DebugLogger) //Entity ORM
    .UseMySql(AccountControllerTest.ConnectionString);
var context = new AccountContent(builder.Options);
```

DbContext SQL .

모두 실행 | 실행... | 재생 목록: 모든 테스트

accountapi 테스트 7개

- ✓ accountapi_test (7)
 - ✓ accountapi_test (7)
 - ✓ AccountControllerTest (6)
 - Actor_CRUDTest
 - ConcurrencyTest
 - LocalStoage
 - TestUserActive
 - TestUserCntChk
 - TokenTest1
- ✓ DBInit (1)
 - ✓ PrepareTestData

선택한 테스트 실행(S) 선택한 테스트 디버그(D) 테스트 사례 연결(A) 모두 확장 Ctrl+오른쪽 화살표

accountapi_test.DBInit Source: DBInit.cs 줄 9

```

4  namespace accountapi_test
5  {
6      public class DBInit
7      {
8          [Fact]
9          public void PrepareTestData()
10         {
11             int count = AccountControllerTest.PrepareTestData();
12             Assert.Equal(10, count);
13         }
14     }
15 }
16
17

```

109 % 출력

작성자: 테스트 오전 12:23:52 Informational] [xUnit.net 00:00:01.4390374] Starting: accountapi_test 오전 12:23:55 Informational] [xUnit.net 00:00:03.7861804] Finished: accountapi_test 오전 12:23:55 Informational] ===== 테스트 실행 완료: 1개 실행(0:00:04.4544773) =====

ORM DB

SQL

모두 실행 | 실행... | 재생 목록: 모든 테스트

accountapi 테스트 7개

- ✓ accountapi_test (7)
 - ✓ accountapi_test (7)
 - ✓ AccountControllerTest (6)
 - Actor_CRUDTest
 - ✓ ConcurrencyTest 2초
 - LocalStoage
 - TestUserActive
 - TestUserCntChk
 - TokenTest1
- ✓ DBInit (1)
 - ✓ PrepareTestData 2초

accountapi_test.AccountCon... 모두 복사 Source: AccountControllerTest.cs 줄 97

✓ accountapi_test.AccountControllerTest.ConcurrencyCheck Elapsed time: 0:00:02.315

109 % 출력

```

103 Person person1 = new Person();
104 person1.LastName = "PS";
105 person1.FirstName = "MON";
106 _testContext.Pessoas.Add(person1);
107 _testContext.SaveChanges();
108
109 Person edit1 = _testContext2.Pessoas.FirstOrDefault(e => e.FirstName);
110 Assert.Equal("PS", edit1.LastName);
111
112 for (int i = 0; i < 10; i++)...
113
114 }
115
116 catch(DbUpdateConcurrencyException e)
117 {
118 }

```

Microsoft.EntityFrameworkCore.Database.Transaction:Debug: Beginning transaction with isolation level 'Unspecified'.
Microsoft.EntityFrameworkCore.Database.Command:Debug: Executing DbCommand [Parameters=@p1='?']
UPDATE `test_person` SET `LastName` = @p0
WHERE `PersonId` = @p1 AND `LastName` = @p2;
SELECT ROW_COUNT();
Microsoft.EntityFrameworkCore.Database.Command:Information: Executed DbCommand (3ms) [Parameters=@p1='?']
UPDATE `test_person` SET `LastName` = @p0
WHERE `PersonId` = @p1 AND `LastName` = @p2;
SELECT ROW_COUNT();
Microsoft.EntityFrameworkCore.Database.Command:Debug: A data reader was disposed.
Microsoft.EntityFrameworkCore.Database.Transaction:Debug: Disposing transaction.
Microsoft.EntityFrameworkCore.Database.Connection:Debug: Closing connection to database 'db_test' on server '(local)'.
Microsoft.EntityFrameworkCore.Database.Connection:Debug: Closed connection to database 'db_test' on server '(local)'.
Microsoft.EntityFrameworkCore.Update:Error: An exception occurred in the database while saving changes for entity type 'Person'.
Microsoft.EntityFrameworkCore.DbUpdateConcurrencyException: Database operation expected to affect 1 rows.

Entity-ORM ,

DbUpdateConcurrencyException .

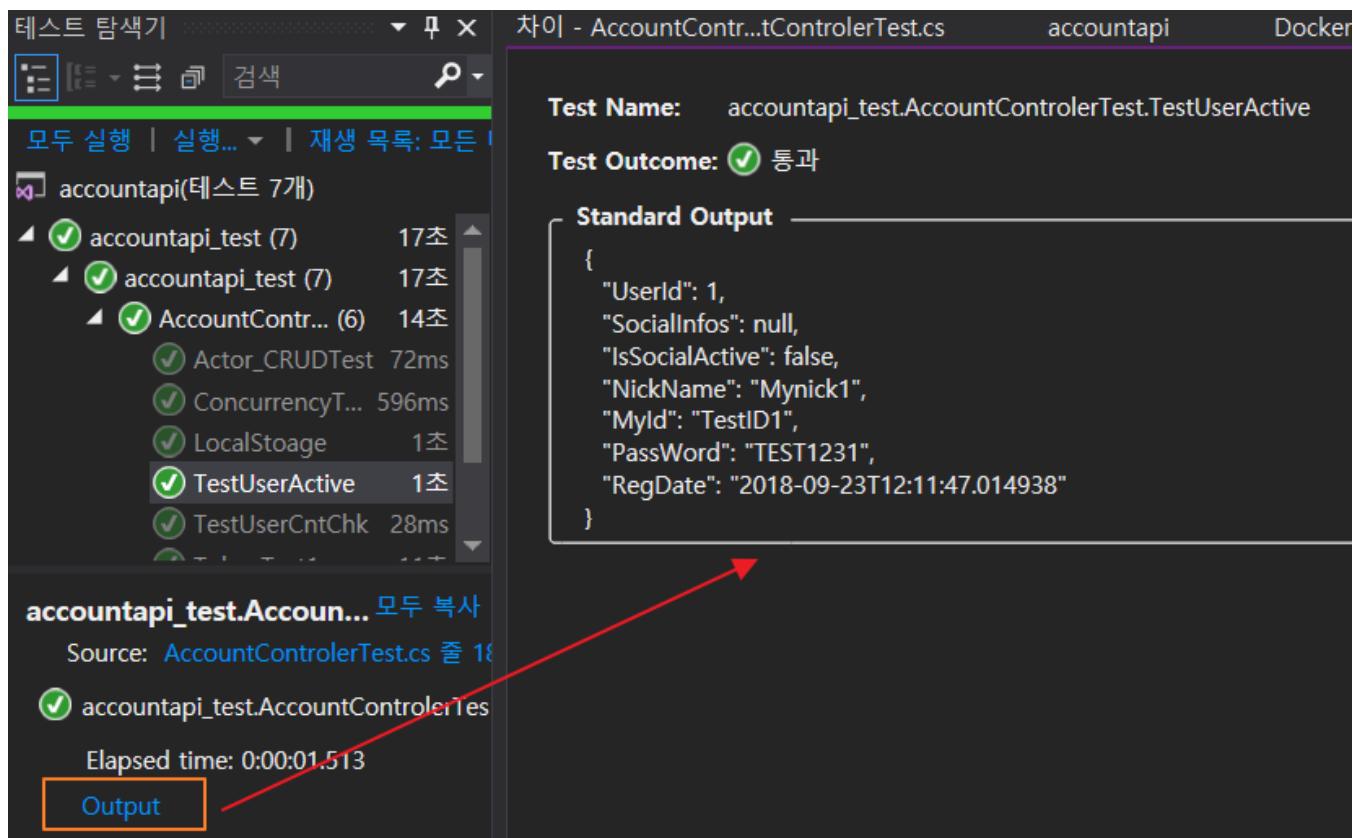
ORM SQL , SQL .

Entity-ORM ConcurrencyCheck .

```
[ConcurrencyCheck]
public string LastName { get; set; }
```

, SQL

SQL OOP DB Concurrency Write .



Json OutputJson .

ITestOutputHelper xUnit .

```
public class AccountControllerTest
{
    private readonly ITestOutputHelper output;

    public AccountControllerTest(ITestOutputHelper output)
    {
        this.output = output;
        InitContext();
    }

    protected void OutPutJson(Object obj)
    {
        output.WriteLine("{0}", JsonConvert.SerializeObject(obj, Formatting.Indented));
    }

    [Fact]
    public void TestUserActive()
    {
        bool expected = false;
        var controller = new AccountController(_accountService);
        User result = controller.GetUserByid(1); // 
        OutPutJson(result); // Print
        Assert.Equal(expected, result.IsSocialActive); // 
    }
}
```

ORM .

DB / / 3 .

- MSDN : <https://docs.microsoft.com/ko-kr/ef/core/saving/concurrency>
- :