

# 第08期-编写单元测试与集成测试

2020年7月24日 13:17

## 所用到的开源库

单元测试驱动框架: <https://github.com/xunit/xunit>

链式断言库: <https://github.com/shouldly/shouldly>

Mock模拟库: <https://github.com/nsubstitute/nsubstitute>

## 添加测试数据

```
public class BookStoreTestDataSeedContributor : IDataSeedContributor,
ITransientDependency
{
    private readonly IRepository<Book, Guid> _bookRepository;
    private readonly IGuidGenerator _guidGenerator;

    public BookStoreTestDataSeedContributor(IRepository<Book, Guid>
bookRepository, IGuidGenerator guidGenerator)
    {
        _bookRepository = bookRepository;
        _guidGenerator = guidGenerator;
    }

    public async Task SeedAsync(DataSeedContext context)
    {
        await _bookRepository.InsertAsync(
            new Book(id: _guidGenerator.Create(),
                name: "Test book 1",
                type: BookType.Fantastic,
                publishDate: new DateTime(2015, 05, 24),
                price: 21
            )
        );

        await _bookRepository.InsertAsync(
            new Book(id: _guidGenerator.Create(),
                name: "Test book 2",
                type: BookType.Science,
                publishDate: new DateTime(2014, 02, 11),
                price: 15
            )
        );
    }
}
```

## 测试 BookAppService 应用服务

```
public class BookAppService_Tests : BookStoreApplicationTestBase
{
    private readonly IBookAppService _bookAppService;

    public BookAppService_Tests()
    {
        _bookAppService = GetRequiredService<IBookAppService>();
    }

    [Fact]
    public async Task Should_Get_List_Of_Books()
    {
        //Act
        var result = await _bookAppService.GetListAsync(new
PagedAndSortedResultRequestDto());

        //Assert
        result.TotalCount.ShouldBeGreaterThan(0);

        result.Items.ShouldContain(b => b.Name == "Test book 1");
    }
}
```

## 测试创建一个合法 Book 实体的场景

```
[Fact]
public async Task Should_Create_A_Valid_Book()
{
    //Act
    var result = await _bookAppService.CreateAsync(
        new CreateUpdateBookDto
        {
            Name = "New test book 42",
            Price = 10,
            PublishDate = DateTime.Now,
            Type = BookType.ScienceFiction
        }
    );

    //Assert
    result.Id.ShouldNotBe(Guid.Empty);
    result.Name.ShouldBe("New test book 42");
}
```

## 测试创建一个非法 Book 实体的场景

```
[Fact]
public async Task Should_Not_Create_A_Book_Without_Name()
{
    var exception = await
Assert.ThrowsAsync<Volo.Abp.Validation.AbpValidationException>(async () =>
    {
        await _bookAppService.CreateAsync(
            new CreateUpdateBookDto
            {
                Name = "",
                Price = 10,
                PublishDate = DateTime.Now,
                Type = BookType.ScienceFiction
            }
        );
    });

    exception.ValidationErrors.ShouldContain(err => err.MemberNames.Any(mem => mem
== "Name"));
}
```

## 使用 Substitute 模拟库单元测试

```
[Fact]
public async Task Should_Get_List_Of_Books_For_Substitute()
{
    IBookAppService _bookAppService = Substitute.For<IBookAppService>();

    var bookDtoList = new List<BookDto>
    {
        new BookDto{ Id=Guid.NewGuid(), Name="EfCore",
CreationTime=DateTime.Now },
        new BookDto{ Id=Guid.NewGuid(), Name="AspNetCore",
CreationTime=DateTime.Now }
    };

    var bookDtoPagedList = new PagedResultDto<BookDto>(bookDtoList.Count,
bookDtoList);

    var pagedRequestDto = new PagedAndSortedResultRequestDto();

    _bookAppService.GetListAsync(pagedRequestDto).Returns(Task.FromResult(bookDtoP
agedList));

    //Act
    var result = await _bookAppService.GetListAsync(pagedRequestDto);
}
```

```

//Assert
result.TotalCount.ShouldBe(2);

result.Items.ShouldContain(b => b.Name == "AspNetCore");
}

```

## 创建 UI 测试与 TestServer 说明

```

[Fact]
public async Task BookList()
{
    string cultureString = "en";

    string cookieValue =
WebUtility.UrlEncode($"c={cultureString}|uic={cultureString}");

    Cookie clientCookie = new Cookie(".AspNetCore.Culture", cookieValue);

    Client.DefaultRequestHeaders.Add("Cookie", clientCookie.ToString());

    var response = await GetResponseAsStringAsync("/books");

    var context = BrowsingContext.New(Configuration.Default);
    var parser = context.GetService<IHtmlParser>();
    var document = parser.ParseDocument(response);

    var trElement = document.QuerySelectorAll("th");

    trElement.Count().ShouldBe(6);

    trElement.ShouldContain(e => e.TextContent == "Actions");
}

```

UI集成测试可用的库： Selenium、 Html Agility Pack 和 **AngleSharp** 框架

第36期 集成测试&远程调试&源码跟踪