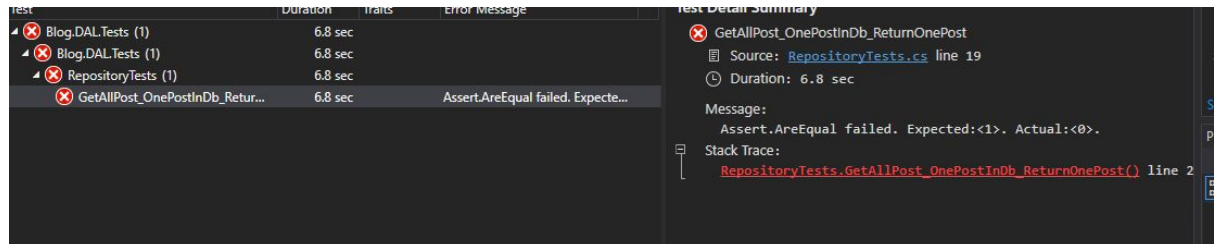


Testy wstępne

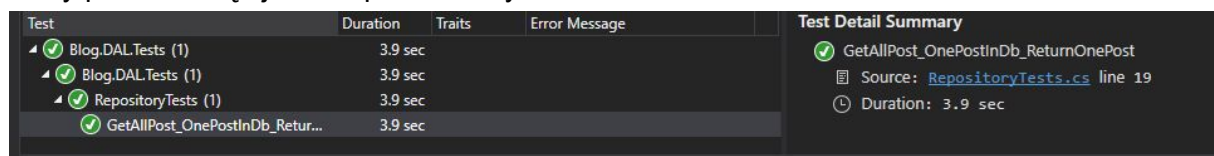
Pierwsze odpalenie test - test nie przechodzi

Jest to spodziewane.



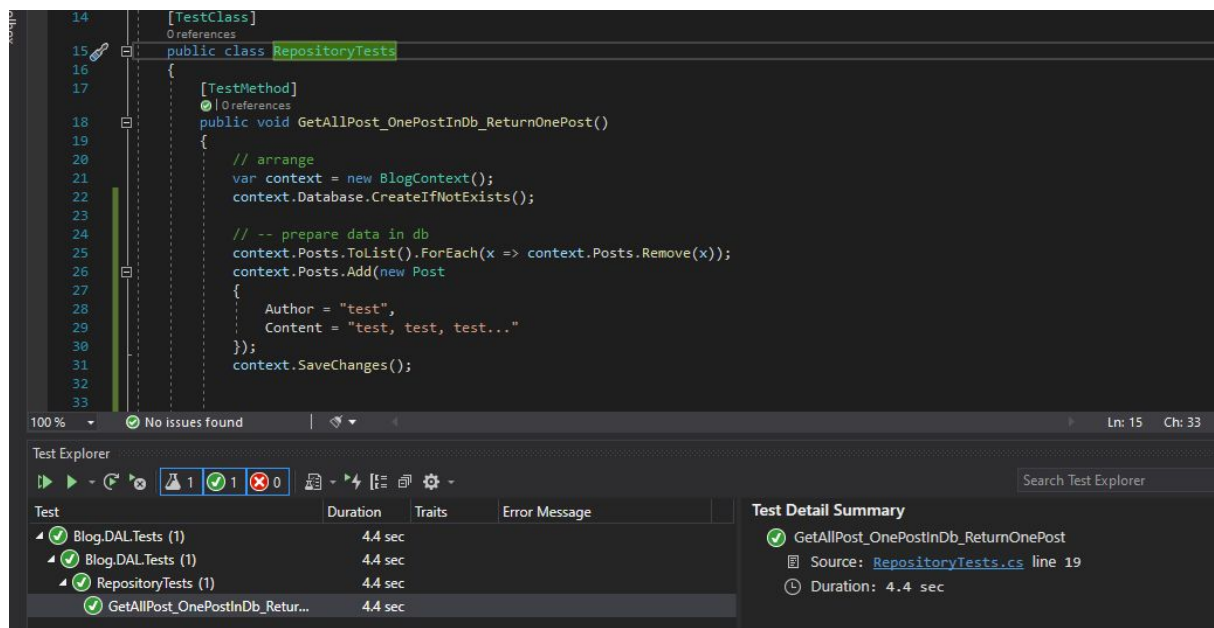
Manualne dodanie danych do bazy

Testy przechodzą - jest to spodziewany rezultat.



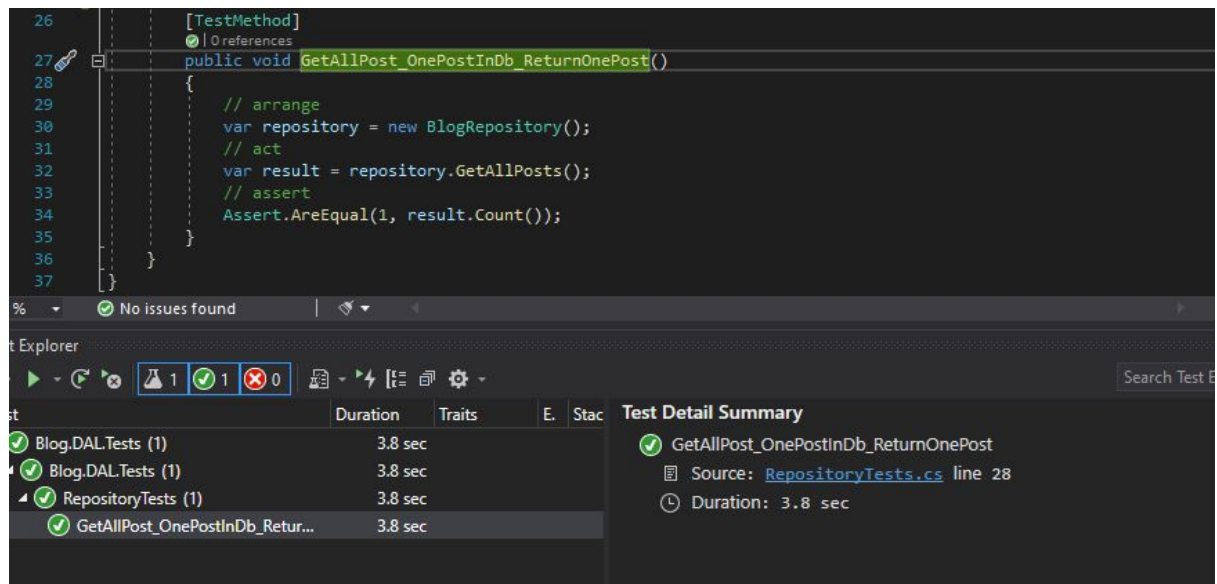
Dodanie kodu uzupełniającego bazę w sekcji arrange

Test ponownie przeszedł.



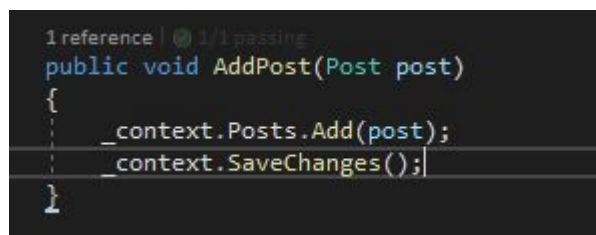
Dodanie klas Fixtures

Test również przechodzi



Właściwe zadania

Dodanie metody do dodawania postów



Dodanie pola Required, oraz utworzenie odpowiedniego testu

```
5 references
public class Post
{
    [Key]
    1 reference | 1/1 passing
    public long Id { get; set; }

    [Required]
    0 references
    public string Content { get; set; }

    1 reference | 1/1 passing
    public string Author { get; set; }
}
```

```
[TestMethod]
[ExpectedException(typeof(DbEntityValidationException))]
1/0 references
public void AddPost_NullContent_ThrowsException()
{
    // arrange
    var repository = new BlogRepository();
    // act
    repository.AddPost(new Post() { Id = 10, Author = "Autor" });
    // assert
}
```

Dodanie nowej tabeli Comments

Klasa komentarza

```
namespace Blog.DAL.Model
{
    1 reference
    public class Comment
    {
        [Key]
        0 references
        public long Id { get; set; }

        0 references
        public string Content { get; set; }

        0 references
        public long PostId { get; set; }

        0 references
        public Post Post { get; set; }
    }
}
```

Aktualizacja kontekstu bazy danych

```
4 references
public class BlogContext : DbContext
{
    3 references
    public DbSet<Post> Posts { get; set; }
    2 references
    public DbSet<Comment> Comments { get; set; }

    1 reference
    public BlogContext() : base("Blog")
    {
    }
}
```

Dodanie nowego pola do klasy z Fixtures

```
1 reference
public class BlogFixturesModel
{
    0 references
    public FixtureTable<Post> Posts { get; set; }
    0 references
    public FixtureTable<Comment> Comments { get; set; }
}
```

Aktualizacja pliku yaml

```
Posts:
- test-post:
  Id: 1
  Author: test
  Content: test, test, test...
- test-post-2:
  Id: 2
  Author: test2
  Content: test2, test2, test2...

Comments:
- comment-1:
  PostId: 1
  Content: ha ha ha
- comment-2:
  PostId: 1
  Content: ha ha ha2
- comment-3:
  PostId: 2
  Content: ha ha ha3
```

Nowe funkcjonalności w repozytorium

```
1 reference | 1/1 passing
public void AddComment(Comment comment)
{
    _context.Comments.Add(comment);
    _context.SaveChanges();
}

3 references | 2/2 passing
public IEnumerable<Comment> GetAllCommentsForPost(Post post)
{
    return _context.Comments.Include(comment => comment.Post).Where(comment => comment.PostId == post.Id);
}
}
```

Testy nowych funkcjonalności

```
[TestMethod]
0 references
public void GetAllCommentsForPost_TwoCommentForOnePostAndOneCommentForSecondPost_ReturnAllTheComments()
{
    // arrange
    var repository = new BlogRepository();
    // act
    var posts = repository.GetAllPosts();

    var commentsForFirstPost = repository.GetAllCommentsForPost(posts.First());
    var commentsForSecondPost = repository.GetAllCommentsForPost(posts.Last());
    // assert
    Assert.AreEqual(2, commentsForFirstPost.Count());
    Assert.AreEqual(1, commentsForSecondPost.Count());
}
}
```

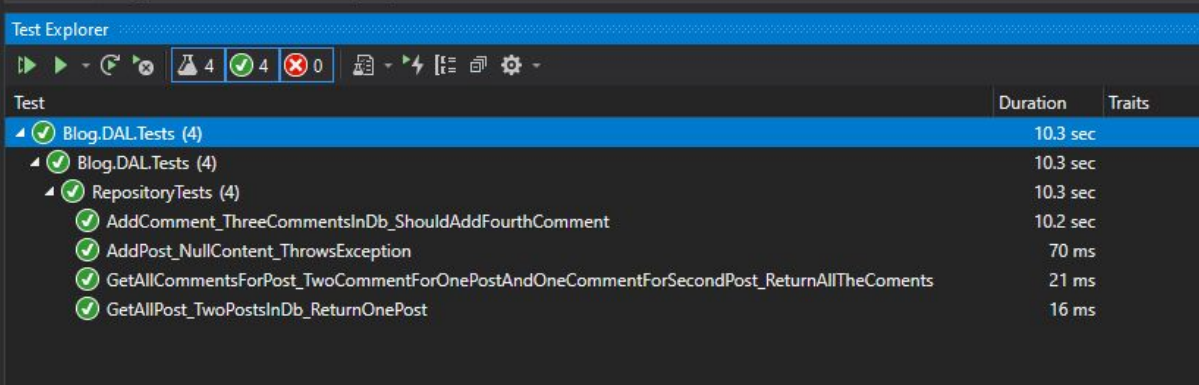
```
[TestMethod]
0 references
public void AddComment_ThreeCommentsInDb_ShouldAddFourthComment()
{
    // arrange
    var repository = new BlogRepository();
    // act
    var posts = repository.GetAllPosts();
    repository.AddComment(new Comment() { Post = posts.First(), Content = "some comment" });


    var commentsForPost = repository.GetAllCommentsForPost(posts.First());

    // assert
    Assert.AreEqual(3, commentsForPost.Count());
}
}
```

Finalne uruchomienie wszystkich testów

Wszystkie testy prawidłowo przechodzą



Test Explorer		
		
Test	Duration	Traits
✓ Blog.DAL.Tests (4)	10.3 sec	
✓ Blog.DAL.Tests (4)	10.3 sec	
✓ RepositoryTests (4)	10.3 sec	
✓ AddComment_ThreeCommentsInDb_ShouldAddFourthComment	10.2 sec	
✓ AddPost_NullContent_ThrowsException	70 ms	
✓ GetAllCommentsForPost_TwoCommentForOnePostAndOneCommentForSecondPost_ReturnAllTheComents	21 ms	
✓ GetAllPost_TwoPostsInDb_ReturnOnePost	16 ms	