Sugar Rush

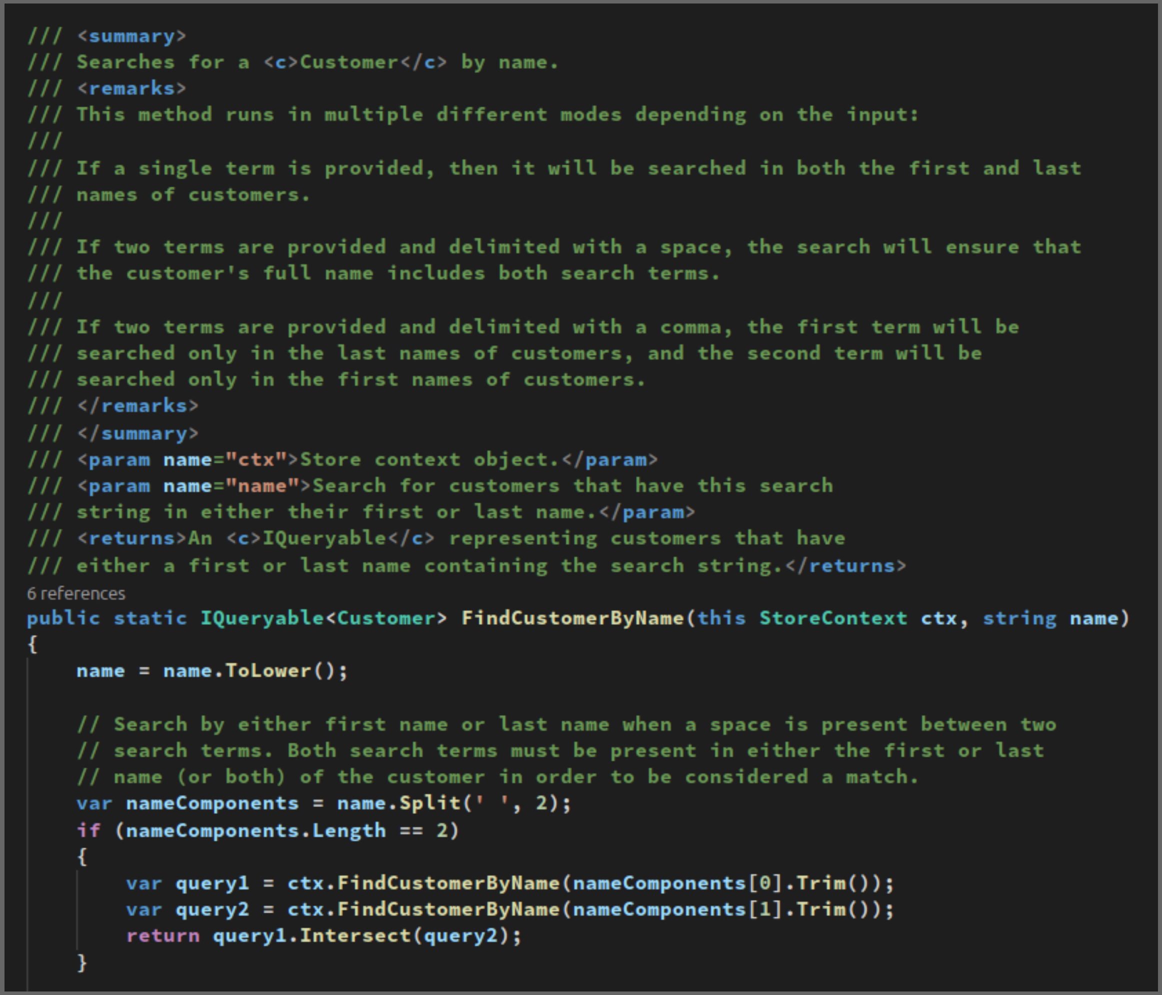
CLI Storefront

PRESENTED BY:
Jayson Lennon

Technologies & Tools

- C# Programming Language
- Visual Studio Code & OmniSharp
- .NET Core
- Entity Framework Core
- SQLite

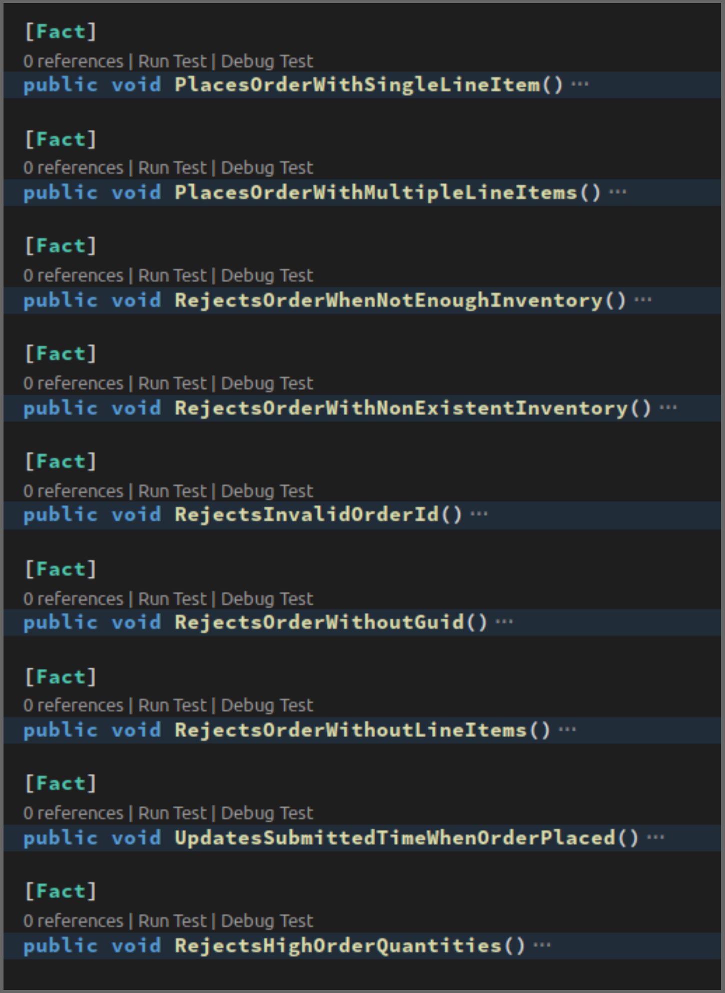
Documentation



```
/// <summary>
/// Searches for a <c>Customer</c> by name.
/// <remarks>
/// This method runs in multiple different modes depending on the input:
///
/// If a single term is provided, then it will be searched in both the first and last
/// names of customers.
///
/// If two terms are provided and delimited with a space, the search will ensure that
/// the customer's full name includes both search terms.
///
/// If two terms are provided and delimited with a comma, the first term will be
/// searched only in the last names of customers, and the second term will be
/// searched only in the first names of customers.
/// </remarks>
/// </summary>
/// <param name="ctx">Store context object.</param>
/// <param name="name">Search for customers that have this search
/// string in either their first or last name.</param>
/// <returns>An <c>IQueryable</c> representing customers that have
/// either a first or last name containing the search string.</returns>
6 references
public static IQueryable<Customer> FindCustomerByName(this StoreContext ctx, string name)
{
    name = name.ToLower();

    // Search by either first name or last name when a space is present between two
    // search terms. Both search terms must be present in either the first or last
    // name (or both) of the customer in order to be considered a match.
    var nameComponents = name.Split(' ', 2);
    if (nameComponents.Length == 2)
    {
        var query1 = ctx.FindCustomerByName(nameComponents[0].Trim());
        var query2 = ctx.FindCustomerByName(nameComponents[1].Trim());
        return query1.Intersect(query2);
    }
```

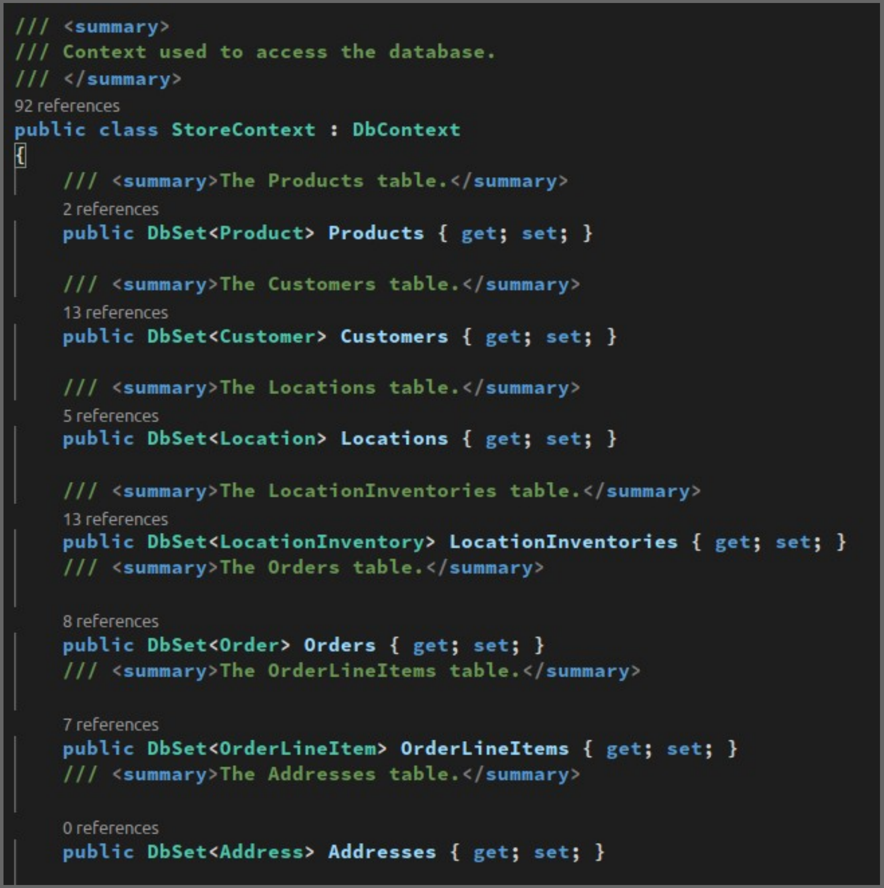
Testing

- Coverage is mostly over order management and customer queries
- 9 tests related to order handling
- 7 tests related to customer data


```
[Fact]
0 references | Run Test | Debug Test
public void PlacesOrderWithSingleLineItem() ⋯

[Fact]
0 references | Run Test | Debug Test
public void PlacesOrderWithMultipleLineItems() ⋯

[Fact]
0 references | Run Test | Debug Test
public void RejectsOrderWhenNotEnoughInventory() ⋯

[Fact]
0 references | Run Test | Debug Test
public void RejectsOrderWithNonExistentInventory() ⋯

[Fact]
0 references | Run Test | Debug Test
public void RejectsInvalidOrderId() ⋯

[Fact]
0 references | Run Test | Debug Test
public void RejectsOrderWithoutGuid() ⋯

[Fact]
0 references | Run Test | Debug Test
public void RejectsOrderWithoutLineItems() ⋯

[Fact]
0 references | Run Test | Debug Test
public void UpdatesSubmittedTimeWhenOrderPlaced() ⋯

[Fact]
0 references | Run Test | Debug Test
public void RejectsHighOrderQuantities() ⋯
```

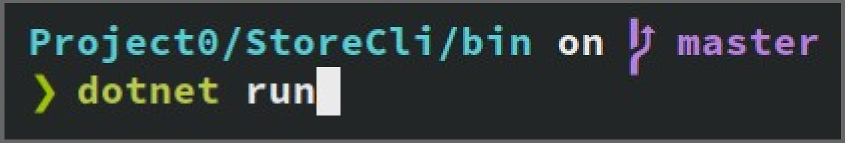
Data Persistence & Management

- Data is saved into a SQLite database
- Database is in 3NF
- Managed by EF Core using a code-first methodology
- Data is accessed through a class library in a separate project which contains all domain classes



```
/// <summary>
/// Context used to access the database.
/// </summary>
92 references
public class StoreContext : DbContext
{
    /// <summary>The Products table.</summary>
    2 references
    public DbSet<Product> Products { get; set; }

    /// <summary>The Customers table.</summary>
    13 references
    public DbSet<Customer> Customers { get; set; }

    /// <summary>The Locations table.</summary>
    5 references
    public DbSet<Location> Locations { get; set; }

    /// <summary>The LocationInventories table.</summary>
    13 references
    public DbSet<LocationInventory> LocationInventories { get; set; }
    /// <summary>The Orders table.</summary>

    8 references
    public DbSet<Order> Orders { get; set; }
    /// <summary>The OrderLineItems table.</summary>

    7 references
    public DbSet<OrderLineItem> OrderLineItems { get; set; }
    /// <summary>The Addresses table.</summary>

    0 references
    public DbSet<Address> Addresses { get; set; }
```

Demo



```
Project0/StoreCli/bin on  master
❯ dotnet run
```

Questions

?