

Lab 6 Encapsulate Collections and Protect Aggregates

Goal

Don't allow client code to directly manipulate entries without going through parent Guestbook entity.

Topics Used

Aggregate, Entity Framework Core 1.1

Requirements

The Guestbook currently exposes a List. This means client code can be written that will add new entries (without creating associated events), or even clear the entire collection. Encapsulate access to the Guestbook aggregate's child collection by using an IEnumerable property.

Note: You must be using Entity Framework Core 1.1 or higher. Update your project files if required.

Detailed Steps

- Ensure your projects are using `Microsoft.EntityFrameworkCore` version **1.1.0** or higher.
- Modify `Guestbook` to have a private `List<GuestbookEntry>` field called `_entries` and a public `IEnumerable<Guestbook>` that returns `_entries.ToList()`.
- You'll need to update `AppDbContext` to support this by adding the following:

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    var navigation = modelBuilder.Entity<Guestbook>()
        .Metadata.FindNavigation(nameof(Guestbook.Entries));

    navigation.SetPropertyAccessMode(PropertyAccessMode.Field);
}
```

With this in place, you should be able to run the application and add entries. However, some tests are likely failing, because the modified `Guestbook` class cannot be deserialized from the JSON that is returned from the API methods.

- Create DTOs for `Guestbook` and `GuestbookEntry`
 - Expose at least `Id` and `Entries` for `GuestbookDTO`
 - Expose `Id`, `GuestbookId`, `EmailAddress`, `Message`, and `DateTimeCreated` for `GuestbookEntryDTO`
- Update integration test methods to deserialize to `GuestbookDTO`

At this point the tests should pass, despite the fact the APIs still actually return domain model types.

Nonetheless, it's better to return DTOs rather than domain model instances.

- Update `GuestbookController` to return DTO types.
- Update `GuestbookController` to accept DTO type as input for `NewEntry`
- Add any additional properties to the DTOs necessary to represent the underlying model types