Querying and Saving Related Data

Julie Lerman
MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK

@julielerman thedatafarm.com

Overview

Solution changes since previous module

Inserting related data

Querying related data

Using related data to filter query results

Persisting disconnected graphs

- Using DbSet methods
- Using DbContext.Entry
- Using the new TrackGraph method

Changes to Solution Since Previous Module

Inserting Related Data - Basics

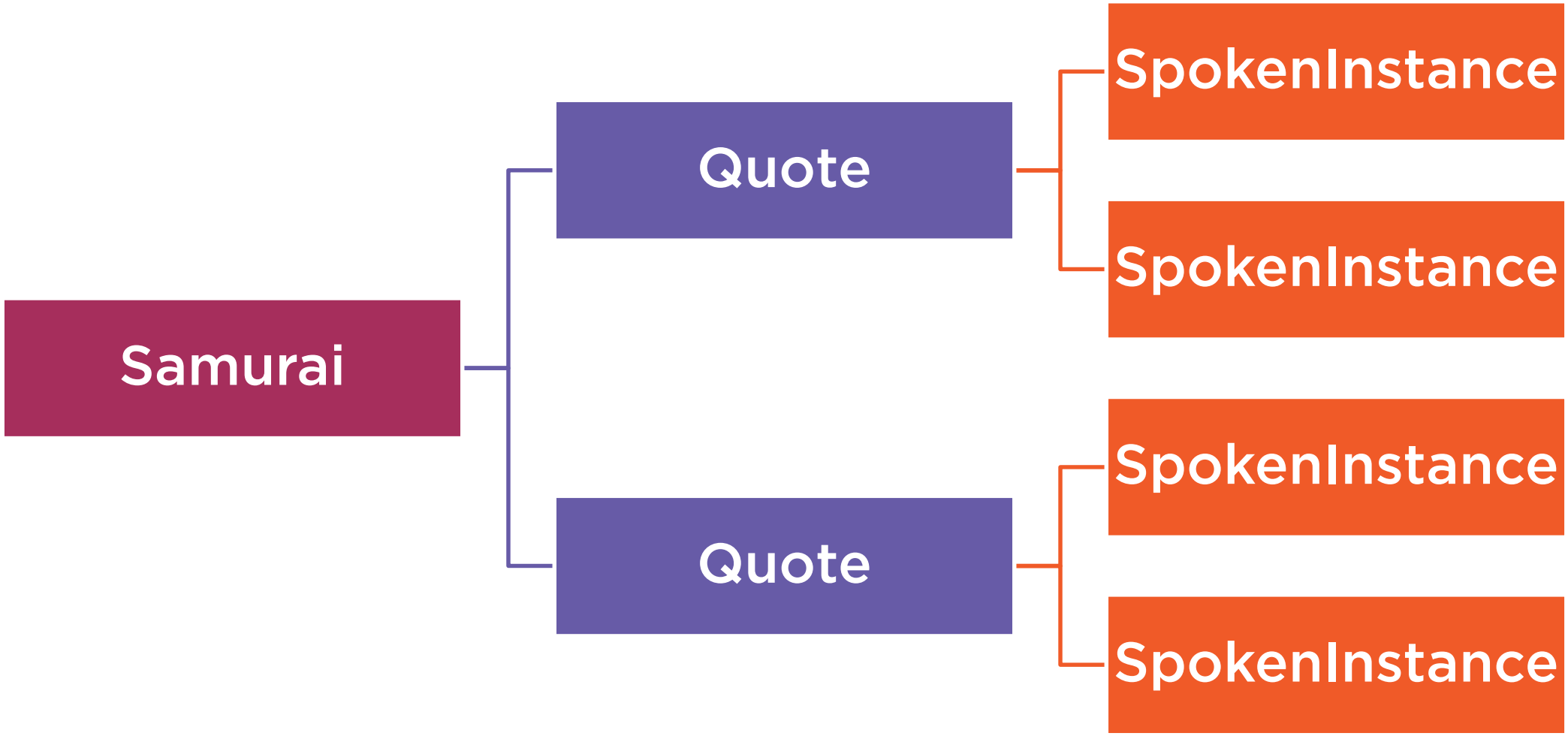
Querying Related Data: Eager Loading

Loading Related Data

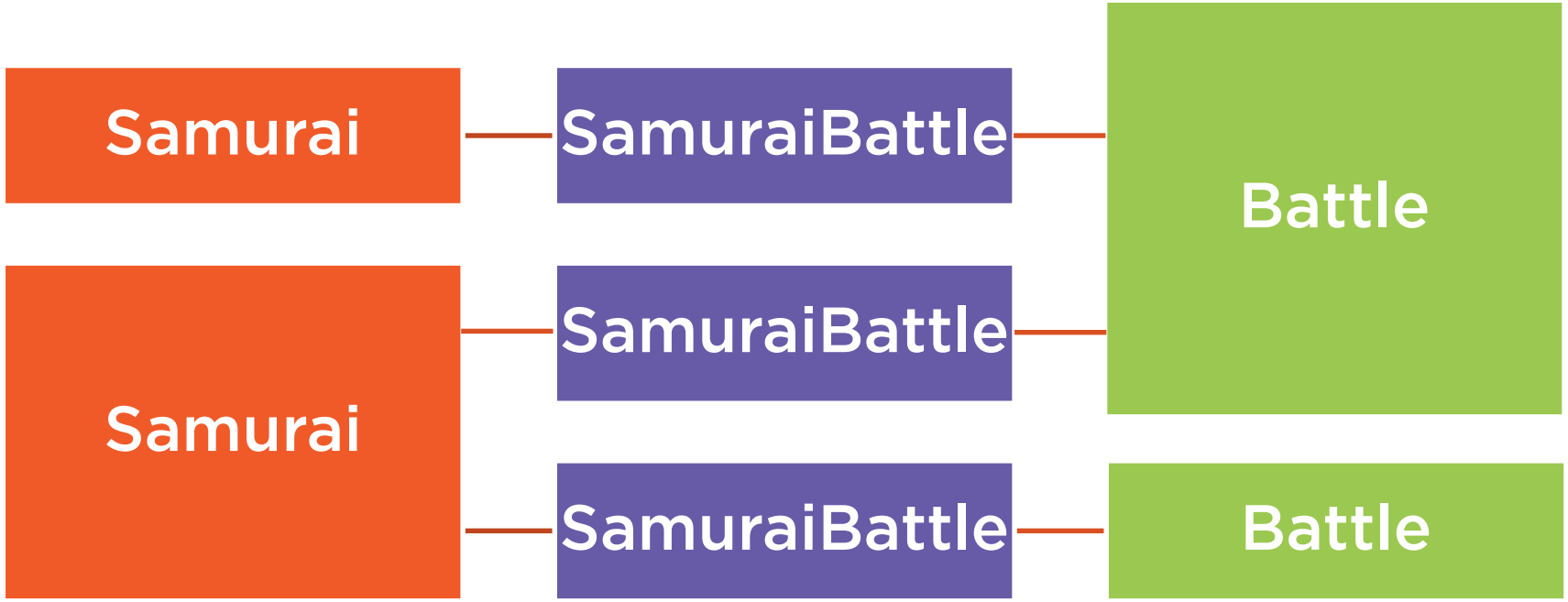
	EF -> EF6	EF Core 1.1
Eager via Include()	✓	✓
Eager via projection	✓	Not quite
Explicit Load	✓	✓
Lazy Loading	✓	Future release

Include loads the entire set of related objects

Multi-Level Include



Multi-Level Include



Retrieving Related Data Using Projections

Loading Related Data

	EF -> EF6	EF Core 1.1
Eager via Include()	✓	✓
Eager via projection	✓	Not quite
Explicit Load	✓	✓
Lazy Loading	✓	Future release

Projections in EF Core 1.1

	SQL	Results
Single Type	Simple	As expected
Scalar values across relationships	N+1	As expected
Full related objects	N+1	Not tracked
Full related objects, filtered	Requires ToList or other LINQ exec method	Not tracked

Explicitly Loading Related Data

Loading Related Data

	EF -> EF6	EF Core 1.1
Eager via Include()	✓	✓
Eager via projection	✓	Not quite
Explicit Load	✓	✓
Lazy Loading	✓	Future release

Explicit loading allows you to filter the related data

```
_context.Entry(object).Collection(o=>o.property)
                      .Query()
                      .LINQMethod(lambda)
                      .Load()

_context.Entry(object).Reference(o=>o.property)
                      .Query()
                      .LINQMethod(lambda)
                      .Load()
```

Explicit Load with Filter

Loading Related Data Pros & Cons

	Eager via Include	Eager via Projection	Post-Query Explicit Load
Code	Single Query	Single Query Requires a hack	Separate Method
Limitations	No filtering, sorting, etc.	1.1 bug prevents tracking	None that I can think of
Database trips	Reference: 1 Collection: N+1	Reference: 1 Collection: N+1	Initial + Load method

Using Related Data to Filter Objects

EF Core's Disconnected Graph Behaviors

Different Change Tracker Behaviors

Completely Disconnected

Connecting entities when
none
are being tracked

Partially Disconnected

Connecting entities when
some
are already tracked

Behavior has changed
from EF6.

Pay attention!

What I Mean Is...

DbContext.Add
DbContext.Attach
DbContext.Update
DbContext.Remove

DbContext.AddRange
DbContext.AttachRange
DbContext.UpdateRange
DbContext.RemoveRange

Further Reading on Handling Full Graphs

https://blog.oneunicorn.com/2016/11/17/add-attach-update-and-remove-methods-in-ef-core-1-1/

Nov 17

Add, Attach, Update, and Remove methods in EF Core 1.1

by Arthur Vickers

EF Core provides a variety of ways to start tracking entities or change their state. This post gives a brief overview of the different approaches.

Tracking queries

Queries will automatically track returned entities unless tracking has been turned off. If you want to track entities from a query, then just use this automatic tracking. Do not use a no-tracking query and then try to attach entities–doing so is slower and can require special handling for shadow properties.

Entity states

Tracked entities can be in one of four states. The state of an entity determines how it is processed when SaveChanges is called.

- Added: The entity does not yet exist in the database. SaveChanges should insert it.
- Unchanged: The entity exists in the database and has not been modified on the client. SaveChanges should ignore it.
- Modified: The entity exists in the database and has been modified on the client. SaveChanges should send updates for it.
- Deleted: The entity exists in the database but should be deleted. SaveChanges should delete it.

Follow me on Twitter

Tweets by @ajcvickers

Arthur Vickers
@ajcvickers
Blogged: Add, Attach, Update, and Remove methods in EF Core 1.1 #efcore blog.oneunicorn.com/2016/11/17/add...

Add, Attach, ...
EF Core prov...
blog.oneunic...

18 Nov

Microsoft | Developer Network

magazine | Issues and downloads | Subscribe | Submit article

Issues and downloads / 2016 / August 2016 / Data Points - EF Core Change-Tracking Behavior: Unchanged, Modified and Added

AUGUST 2016

Data Points - EF Core Change-Tracking Behavior: Unchanged, Modified and Added

By Julie Lerman | August 2016 | Get the Code

Did you see what I did there, in this column's title? You may have recognized Unchanged, Modified and Added as enums for EntityState in Entity Framework (EF). They also help me describe the behaviors of change tracking in EF Core compared to earlier versions of Entity Framework. Change tracking has become more consistent in EF Core so you can be more confident in knowing what to expect when you're working with disconnected data.

Keep in mind that while EF Core attempts to keep the paradigms and much of the syntax of earlier versions of Entity Framework, EF Core is a new set of APIs—a completely new code base written from scratch. Therefore, it's important not to presume that everything will behave exactly as it did in the past. The change tracker is a critical case in point.

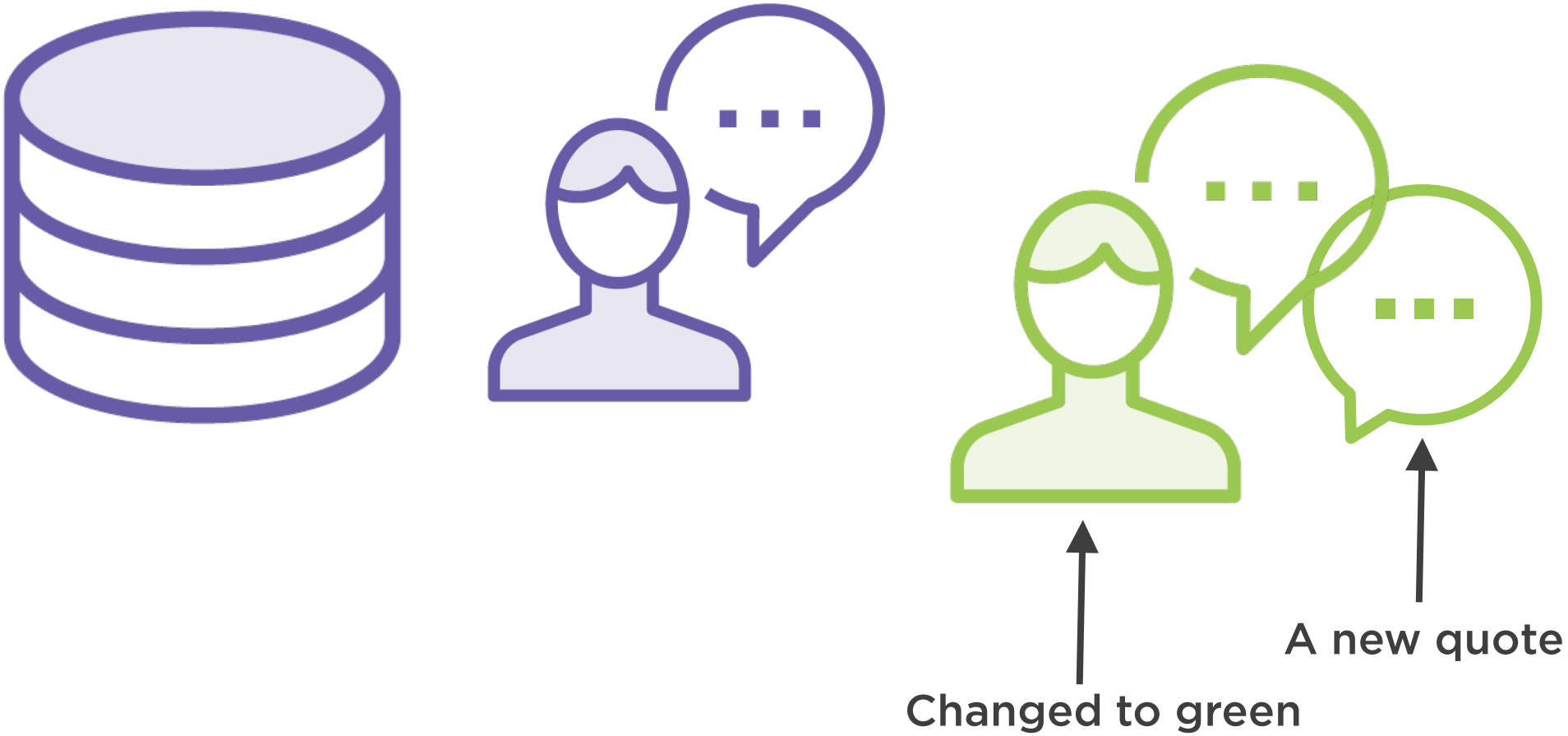
Because the first iteration of EF Core is targeted to align with ASP.NET Core, a lot of the work focused on disconnected state, that is, making sure Entity Framework can handle the state of objects coming out of band, so to speak, where EF hasn't been keeping track of those objects. A typical scenario is a Web API that's accepting objects from a client application to be persisted to the database.

DbSet/DbContext Methods Behavior with Untracked Graphs

Add AddRange	Attach AttachRange	Update UpdateRange	Remove RemoveRange
All objects marked **Added** Regardless of key values	Empty store-generated keys: **Added** Root* & rest are marked: **Unchanged**	Empty store-generated keys: **Added** Root* & rest are marked: **Modified**	Root only marked **Deleted**

*In EF Core 1.1, root is unaffected by presence of key values. This will change in the future.

Update Behavior Is Handy for Mixed State



Changes and Additions to Change Tracker

```
Entry(objGraph).State=EntityState
```

Tracking Graphs with Entry.State

Root object set to specified state

Rest of graph is ignored

Important change from EF6!

Entry.State:Track a Single Entity from a Graph

```
DbContext.ChangeTracker
          .TrackGraph(graph, function)
```

Tracking Graphs with EF Core's TrackGraph

Iterate through graph

Perform function on each entity

TrackGraph with State Property as Function

```
DbContext.ChangeTracker
    .TrackGraph(graph, e=>e.State=EntityState.Attach)
```

Special rules will apply to already tracked entities.
See Arthur Vickers' post & my MSDN Mag article for specifics.

Review

	EF -> EF6	EF Core 1.1
Eager via Include()	✓	✓
Eager via projection	✓	Not quite
Explicit Load	✓	✓
Lazy Loading	✓	Future release

Tracking Graphs with Entry.State

Root object set to specified state

Rest of graph is ignored

Important change from EF6!

Add AddRange	Attach AttachRange	Update UpdateRange	Remove RemoveRange
All objects marked **Added** Regardless of key values	Empty store-generated keys: **Added** Root* & rest are marked: **Unchanged**	Empty store-generated keys: **Added** Root* & rest are marked: **Modified**	Root only marked **Deleted**

C:\WINDOWS\system32\cmd.exe

```
AddGraphViaTrackGraphAllNew
Samurai : Added
Quote : Added

AddGraphViaTrackGraphWithKeyValues
Samurai : Added
Quote : Added

AttachGraphViaTrackGraphAllNew
Samurai : Unchanged
Quote : Unchanged

AttachGraphViaTrackGraphWithKeyValues
Samurai : Unchanged
Quote : Unchanged

UpdateGraphViaTrackGraphAllNew
Samurai : Modified
Quote : Modified

UpdateGraphViaTrackGraphWithKeyValues
Samurai : Modified
Quote : Modified

DeleteGraphViaTrackGraphAllNew
Samurai : Deleted
Quote : Deleted

DeleteGraphViaTrackGraphWithKeyValues
Samurai : Deleted
Quote : Deleted
```

Resources

Entity Framework Core on GitHub github.com/aspnet/entityframework

EF Core Documentation docs.microsoft.com/ef

Arthur Vickers Change Tracking Post:
blog.oneunicorn.com/2016/11/17/add-attach-update-and-remove-methods-in-ef-core-1-1/#more-457

Data Points - EF Core Change-Tracking Behavior: Unchanged, Modified and Added:
msdn.microsoft.com/magazine/mt767693

Querying and Saving Related Data

Julie Lerman
MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK

@julielerman thedatafarm.com