

# Interacting with Your EF Core Model

---



**Julie Lerman**

MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK

@julielerman thedatafarm.com



# Overview



Initial focus is on simple objects

Logging with EF Core

Inserting data

Bulk operations when saving

Querying simple data

Updating objects

Disconnected updates

Querying with raw SQL and stored procedures

Executing raw SQL commands



# Inserting Simple Objects

---



EF6

System.Data.Entity.Internal.InternalContext.Log

EF Core

Microsoft.Extensions.Logging.ILoggerProvider



# Batch Commands When Saving

---



# Querying Simple Objects

---



# Two Ways to Express LINQ Queries

## LINQ Methods

```
context.Samurais.ToList();
```

```
context.Samurais  
.Where(s=>s.Name=="Julie")  
.ToList()
```

## LINQ Query Syntax

```
(FROM s IN context.Samurais)  
.ToList()
```

```
(FROM s  
  IN context.Samurais  
  WHERE s.Name=="Julie")  
.ToList()
```



# LINQ to Entities Execution Methods

ToList()  
First()  
FirstOrDefault()  
Single()  
SingleOrDefault()  
Count()  
LongCount()  
Min()  
Max()

ToListAsync()  
FirstAsync()  
FirstOrDefaultAsync()  
SingleAsync()  
SingleOrDefaultAsync()  
CountAsync()  
LongCountAsync()  
MinAsync()  
MaxAsync()

As of EF Core 1.1, these are evaluated on the client side

Last()  
LastOrDefault()  
Average()

LastAsync()  
LastOrDefaultAsync()  
AverageAsync()

Not a LINQ method, but a DbSet method

Find(keyValue)

FindAsync(keyValue)





# Database Connection Remains Open During Enumeration

```
foreach (var s in context.Samurais){  
    Console.WriteLine(s.Name);  
}
```

```
foreach (var s in context.Samurais){  
    RunSomeValidator(s.Name);  
    CallSomeService(s.Id);  
    GetSomeMoreDataBasedOn(s.Id);  
}
```

```
var samurais=context.Samurais.ToList()  
foreach (var s in samurais){  
    RunSomeValidator(s.Name);  
    CallSomeService(s.Id);  
    GetSomeMoreDataBasedOn(s.Id);  
}
```

◀ Minimal effort on enumeration, ok

◀ Lots of work for each result.  
Connection stays open until last  
result is fetched.

◀ Smarter to get results first



# So, You Don't Want Every Row?

---



# What Happens When There Aren't Any?

First()

Single()

Last()

**System.InvalidOperationException:**  
Sequence contains no elements

FirstOrDefault()

SingleOrDefault()

LastOrDefault()

**Returns null**



# Updating Simple Objects

---



# Disconnected Updates

---



# Update Options

`DbSet.Update(entity)`

`DbSet.UpdateRange(entityA, entity)`

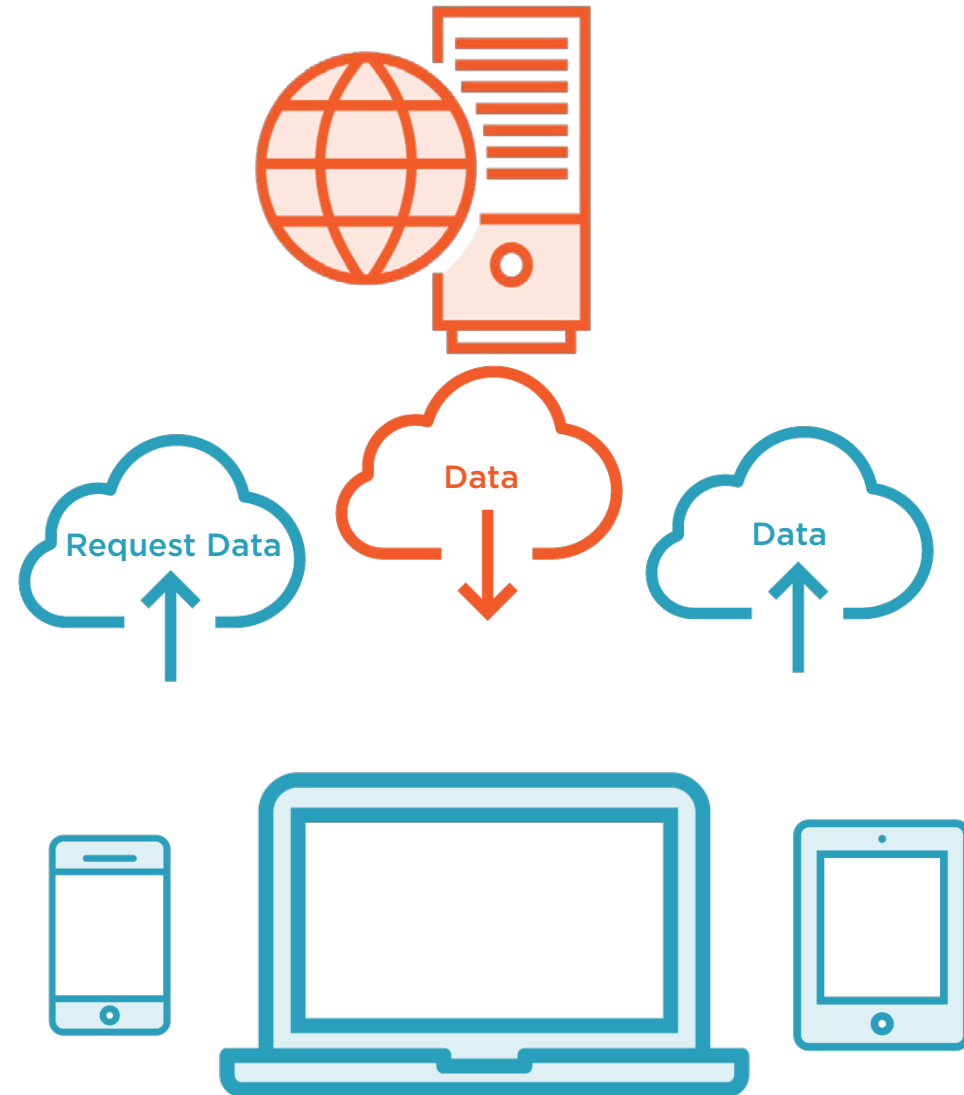
`DbContext.Update(entity)`

`DbContext.UpdateRange(entityA, entityB)`

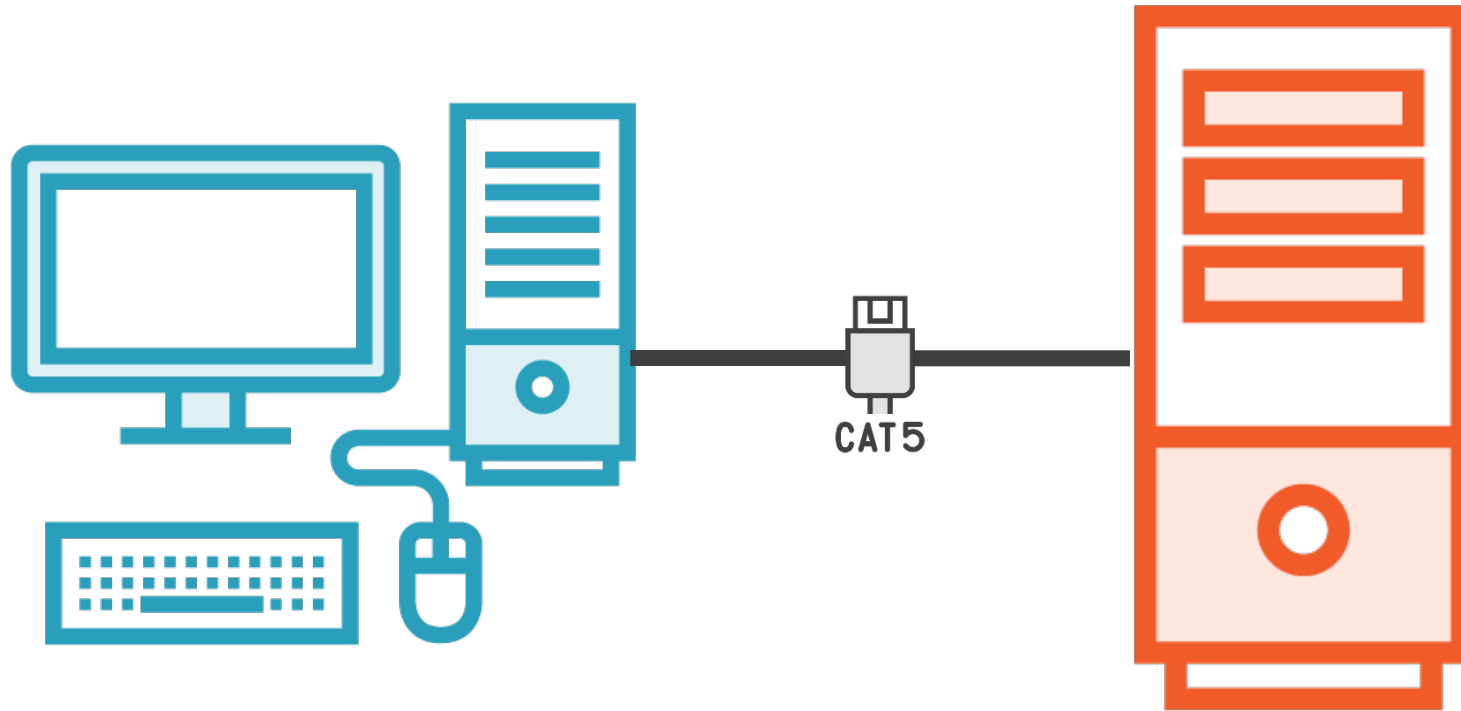
`DbContext.Entry(entity).State=EntityState.Modified`



# Disconnected Clients



# Network Connected Clients





# Client Storing Data Locally



```
public class SamuraiController : Controller
{
    private SamuraiContext _context = new SamuraiContext();

    public Samurai Get(int id) {
        return _context.Samurais.Find(id);
    }

    public void Put([FromBody] Samurai samurai) {
        _context.Samurais.Update(samurai);
        _context.SaveChanges();
    }
}
```

DbContext Will Be New for Each Web Request

**No knowledge of object state tracked by other instances**

**Add/Update/Attach/Remove or Entry.State to start tracking**



# Design for short-lived DbContexts in web apps



Do not create  
static DbContexts  
to enable ongoing  
change tracking  
in a web application



# Deleting Objects with EF Core

---



DbContext can only delete  
objects it is aware of,  
i.e., already tracking



# What About Delete by ID?

**DbSet.Find(int)**



**DbSet.Delete(int)**



# Alternate Ideas to Delete by ID

**Build a proxy object  
in memory  
using the key value**



**Use a stored procedure**





# Querying with Raw SQL and Stored Procedures

---



What About...

Queries that are hard to express in LINQ?

Queries that result in icky SQL?

Queries that result in sloooooow SQL?

All of my stored procedures?



# EF Core's Raw SQL Commands

## Query

**DbSet.FromSql( )**

*\*Replaces DbSet.SqlQuery, DbContext.Database.SqlQuery*

SQL query or execute stored procedure

Allows parameters

Must return full entity types (1.1)

Result set column names = mapped names

Query must be flat ...no related data

*Not yet: stored procedure mappings*

## Command

**DbContext.Database**

**.ExecuteSqlCommand( )**

*\*Same as earlier EF versions*

Returns int (rows affected)

Can use output parameters

ExecuteSqlCommandAsync available

Does not start a transaction ala SaveChanges



# Custom Functions in LINQ to EF Queries

---



# Smarter Query Composition

**More Server Side Execution**

**FromSql()**

**+**

**LINQ Methods**

**Smarter Query Interpretation**

**LINQ Methods**

**+**

**Your Custom Methods**



# Executing Raw SQL for Non-Query Commands

---



# EF Core's Raw SQL Commands

## Query

### **DbSet.FromSql( )**

*\*Replaces DbSet.SqlQuery, DbContext.Database.SqlQuery*

SQL query or execute stored procedure

Allows parameters

Must return full entity types (1.1)

Result set column names = mapped names

Query must be flat ...no related data

*Not yet: stored procedure mappings*

## Command

**DbContext.Database**

### **.ExecuteSqlCommand( )**

*\*Same as earlier EF versions*

Returns int (rows affected)

Can use output parameters

ExecuteSqlCommandAsync available

Does not start a transaction ala SaveChanges



# Review

Logging

Insert

Batched  
commands

Queries

Query  
execution

Update  
tracked  
objects

Update  
untracked  
objects

Delete  
needs full  
object

FromSql()

Raw  
queries

EXEC  
stored  
procs

ExecuteSqlCommand()

Input &  
output  
params

Compose  
queries





# Resources

Entity Framework Core on GitHub [github.com/aspnet/entityframework](https://github.com/aspnet/entityframework)

EF Core Documentation [docs.microsoft.com/ef](https://docs.microsoft.com/ef)

Richie Rump blog: <https://www.brentozar.com/archive/author/richie/>

Trackable Entities: <http://trackableentities.github.io/>

Breeze: <http://www.getbreezenow.com/>

Calling SQL Server stored procedures from Entity Framework

<http://simon-hughes.blogspot.com/2015/01/calling-sql-server-stored-procedures.html>



# Interacting with Your EF Core Model

---



**Julie Lerman**

MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK

@julielerman thedatafarm.com

