Entity Framework Datasheet

SIT331 Full Stack Development: Secure Backend Services

Andrew Belcher

What is Entity Framework?

Entity Framework (EF) an open source Object Relational Mapping (ORM) Framework that Microsoft makes available as part of .NET version 3.5 and later. It can be utilised to abstract the ties to a relational database, allowing for decoupling between our application and the logic of data access. Prior to EF, developers would often use ADO.NET or Enterprise Data Access to save or retrieve application data from the underlying database. This involved opening connections, fetching data, converting data, etc. which was cumbersome and error prone. EF automates all these database related activities. The two latest versions of EF:

	Release Date	.NET Framework
Entity Framework 6	2013	.NET 4.0 & .NET 4.5, VS 2012
Entity Framework Core 2.0	2017	.NET Core 2.0, VS 2017

Create, Read, Update & Delete (CRUD) Operations

Context class

The context class is a class which derives from System.Data.Entity.DbContext.DbContext. An instance of the context class can combine multiple changes under a single database transaction.

Context Class

The `ExampleContext` class is derived from `DbContext` which makes it a context class. It includes three entities `Cats`, `Dogs` & `Fish`.

```
public class ExampleContext : DbContext
{
    public ExampleContext()
    {

    }

    public DbSet<Cat> Cats { get; set; }
    public DbSet<Dog> Dogs { get; set; }
    public DbSet<Meal> Meals { get; set; }
}
```

Example Code

CRUD Operations

The example below shows how to use `Add`, `Remove` and `SaveChanges`.

```
using var petContext = new ExampleContext();

// Create - Adding a dog
petContext.Add(new Dog { Name = "Charlie"});
petContext.SaveChanges();

// Read - Querying for a cat
var cat = petContext.Single(c => c.Name = "Mittens");

// Update - Adding a meal to a cat
cat.Meals.Add(new Meal { Food = "Tuna", Weight = "50grams" });
petContext.SaveChanges();

// Delete - removing cat from the database
petContext.Remove(cat);
petContext.SaveChanges();
```

Querying Data

Querying

The examples below show how to query data.

```
// Load all cats and related meals
var cats = petContext.Cats.Include(c => c.Meals).ToList();

// Load one dog and its related meals
var dogs = petContext.Dogs.Single(d => d.Name == "Charlie").Include(d => d.Meals);
```

References

https://www.entityframeworktutorial.net/
https://github.com/dotnet/efcore
https://learn.microsoft.com/en-us/ef/
https://www.tutorialspoint.com/entity_framework/index.htm