Previous Next

Entity Framework Core: DbContext

The DbContext class is an integral part of Entity Framework. An instance of `DbContext` represents a session with the database which can be used to query and save instances of your entities to a database. `DbContext` is a combination of the Unit Of Work and Repository patterns.

`DbContext` in EF Core allows us to perform following tasks:

1. Manage database connection
2. Configure model & relationship
3. Querying database
4. Saving data to the database
5. Configure change tracking
6. Caching
7. Transaction management

To use `DbContext` in our application, we need to create the class that derives from `DbContext`, also known as context class. This context class typically includes DbSet<TEntity> properties for each entity in the model. Consider the following example of context class in EF Core.

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

    }

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
    }

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
    }
    //entities
    public DbSet<Student> Students { get; set; }
    public DbSet<Course> Courses { get; set; }
}
```

In the example above, the `SchoolContext` class is derived from the `DbContext` class and contains the `DbSet<TEntity>` properties of `Student` and `Course` type. It also overrides the `OnConfiguring` and `OnModelCreating` methods. We must create an instance of `SchoolContext` to connect to the database and save or retrieve `Student` or `Course` data.

The `OnConfiguring()` method allows us to select and configure the data source to be used with a context using `DbContextOptionsBuilder`. Learn how to configure a DbContext class at here.

The `OnModelCreating()` method allows us to configure the model using `ModelBuilder` Fluent API.

DbContext Methods

Method	Usage
Add	Adds a new entity to `DbContext` with Added state and starts tracking it. This new entity data will be inserted into the database when SaveChanges() is called.

AddAsync	Asynchronous method for adding a new entity to `DbContext` with Added state and starts tracking it. This new entity data will be inserted into the database when SaveChangesAsync() is called.
AddRange	Adds a collection of new entities to `DbContext` with Added state and starts tracking it. This new entity data will be inserted into the database when SaveChanges() is called.
AddRangeAsync	Asynchronous method for adding a collection of new entities which will be saved on SaveChangesAsync().
Attach	Attaches a new or existing entity to `DbContext` with Unchanged state and starts tracking it.
AttachRange	Attaches a collection of new or existing entities to `DbContext` with Unchanged state and starts tracking it.
Entry	Gets an `EntityEntry` for the given entity. The entry provides access to change tracking information and operations for the entity.
Find	Finds an entity with the given primary key values.
FindAsync	Asynchronous method for finding an entity with the given primary key values.
Remove	Sets Deleted state to the specified entity which will delete the data when SaveChanges() is called.
RemoveRange	Sets Deleted state to a collection of entities which will delete the data in a single DB round trip when SaveChanges() is called.
SaveChanges	Execute INSERT, UPDATE or DELETE command to the database for the entities with Added, Modified or Deleted state.
SaveChangesAsync	Asynchronous method of SaveChanges()
Set	Creates a `DbSet<TEntity>` that can be used to query and save instances of TEntity.
Update	Attaches disconnected entity with Modified state and start tracking it. The data will be saved when SaveChagnes() is called.
UpdateRange	Attaches a collection of disconnected entities with Modified state and start tracking it. The data will be saved when SaveChagnes() is called.
OnConfiguring	Override this method to configure the database (and other options) to be used for this context. This method is called for each instance of the context that is created.

OnModelCreating	Override this method to further configure the model that was discovered by convention from the entity types exposed in `DbSet<TEntity>` properties on your derived context.

DbContext Properties

Method	**Usage**
ChangeTracker	Provides access to information and operations for entity instances this context is tracking.
Database	Provides access to database related information and operations for this context.
Model	Returns the metadata about the shape of entities, the relationships between them, and how they map to the database.

Learn how to create the first simple EF Core console application in the next chapter.

Previous Next

feedback@entityframeworktutorial.net

TUTORIALS

- EF Basics
- EF Core
- EF 6 DB-First
- EF 6 Code-First

E-MAIL LIST

Subscribe to EntityFrameworkTutorial email list and get EF 6 and EF Core Cheat Sheets, latest updates, tips & tricks about Entity Framework to your inbox.

Email address GO

We respect your privacy.