

Dotnet Core Temelleri

Entity Framework Nedir ?

Hafif, genişletilebilir ve open source olan veri erişim teknolojisidir. Microsoft tarafından geliştirilmektedir. EF Core kullanırken sıklıkla karşımıza çıkacak iki ifade vardır. Entity Framework Core (EF Core), Microsoft tarafından geliştirilen, .NET platformu için bir nesne ilişkisel eşleme (Object-Relational Mapping, ORM) framework'üdür. ORM, veritabanıyla nesne tabanlı programlama arasındaki uyumu sağlamak amacıyla kullanılır. Entity Framework Core, özellikle .NET Core ve .NET 5 gibi çeşitli .NET platformlarında çalışabilen hafif ve açık kaynaklı bir ORM çözümüdür.

Entity Framework Core, .NET uygulamalarının veritabanlarıyla etkileşimde bulunmasını kolaylaştırmak için kullanılır. Bu framework sayesinde, veritabanındaki tabloları, sütunları ve ilişkileri nesne modelleriyle eşleştirebilirsiniz. Bu sayede, veritabanı işlemlerini SQL sorguları yazmak yerine nesne yönelimli bir yaklaşımla gerçekleştirebilirsiniz.

- ORM (Object-Relational Mapping)
- LINQ (Language Integrated Query)

ORM: Veri tabanına erişim tekniğidir. OOP'nin tüm nimetlerinden faydalanarak veri tabanlarına karşı queryler oluşturmamızı sağlar. (Create,update,delete vb.) Bu ORM tekniği altında birçok veri erişim teknolojisi vardır. Mini ORM olarak ifade edilen Dapper veri erişim teknolojisidir.

Linq: C# 3.0 ile hayatımıza giren sorgular oluşturmamıza imkan veren bir teknolojidir. EF core ve linq birbirini tamamlayan teknolojilerdir. Örneğin C# dilindeki collectionlara karşı Queryler oluşturabiliyorsunuz(List,Linked List,Dictionary). Bunlara karşı sorgular oluşturabiliyorsunuz ve bunları tip güvenli olarak gerçekleştirebilirsiniz. Linq ve EF Core farklı teknolojilerdir. EF core veri erişim teknolojisidir. Sorguların SQL cümlecğine çevrilmesi, veri tabanına gönderilmesi, veri tabanından gönderilen datanın Linq tarafında belirtilen tiplere maplenmesi gibi tüm süreçler EF Core tarafından yönetilmektedir. Linq sadece tip güvenli sorgular oluşturmamıza imkan vermektedir.

EF Core'un temel özellikleri arasında şunlar bulunur:

- **Code First Yaklaşımı:** Veritabanı şemalarını kodunuzdaki sınıflara dayalı olarak oluşturmamızı sağlar. Kod tarafında sınıfları tanımlayarak, EF Core otomatik olarak veritabanını oluşturabilir.
- **Migration Desteği:** Veritabanındaki şemada yapılan değişiklikleri takip edebilmeniz ve bu değişiklikleri uygulayabilmeniz için migration (geçiş) mekanizması sunar.

- **LINQ Desteği:** LINQ (Language Integrated Query) kullanarak veritabanına sorgular yazabilirsiniz. Bu, SQL sorguları yazmak yerine daha okunabilir ve yönetilebilir sorgular oluşturmanıza olanak tanır.
- **Çeşitli Veritabanı Motorları Desteği:** EF Core, çeşitli veritabanı motorlarına uyumlu olarak çalışabilir. Microsoft SQL Server, PostgreSQL, MySQL, SQLite gibi popüler veritabanlarına destek sağlar.
- **Performans İyileştirmeleri:** EF Core, performansı artırmak ve gereksiz sorguları önlemek için çeşitli optimizasyonlar içerir.

Entity Framework Core, geliştiricilere veritabanı işlemlerini basitleştirmek ve daha az kod yazarak veritabanıyla etkileşimde bulunmalarını sağlamak için güçlü bir araç sunar.

Entity Framework Core Avantajları nelerdir?

Entity Framework, .NET tabanlı uygulamalarda veritabanı işlemlerini kolaylaştırmak ve veritabanı ile nesne modelleri arasında bir köprü görevi görmek üzere tasarlanmış bir ORM (Object-Relational Mapping) framework'üdür. İşte Entity Framework'ün avantajları:

Nesne Yönelimli Tasarım (OOD): Entity Framework, veritabanı tablolarını nesne modelleriyle eşleştirerek, nesne yönelimli tasarımın güzelliklerini sunar. Bu, geliştiricilere veritabanı işlemlerini nesne tabanlı bir şekilde gerçekleştirebilmelerini sağlar.

Code First ve Database First Yaklaşımları: Entity Framework, iki farklı yaklaşım sunar. Code First yaklaşımıyla sınıfları oluşturarak veritabanını oluşturabilirsiniz. Database First yaklaşımıyla ise var olan bir veritabanını modele dönüştürebilirsiniz.

LINQ Entegrasyonu: LINQ (Language Integrated Query) kullanarak Entity Framework üzerinden veritabanı sorgularını yazabilirsiniz. Bu, SQL sorgularını yazmak zorunda kalmadan, daha okunabilir ve yönetilebilir sorgular oluşturmanıza olanak tanır.

Migration Desteği: Veritabanındaki değişiklikleri takip etmek ve uygulamak için migration'ları kullanabilirsiniz. Bu, modelinizi güncellediğinizde veritabanınızı otomatik olarak güncellemenizi sağlar.

Çeşitli Veritabanı Motorları Desteği: Entity Framework, farklı veritabanı motorlarına uyumlu olarak çalışabilir. Microsoft SQL Server, PostgreSQL, MySQL, SQLite gibi popüler veritabanlarına destek sağlar.

Performans İyileştirmeleri: Entity Framework, performansı artırmak ve gereksiz sorguları önlemek için çeşitli optimizasyonlar içerir. Örneğin, sorguların önbelleğe alınması gibi.

Model İlişkileri ve Bağlantıları Kolay Yönetim: İlişkileri ve bağlantıları modelleme ve yönetme konusunda Entity Framework oldukça esnek ve güçlüdür. Foreign key, primary key gibi ilişkileri kolayca tanımlayabilirsiniz.

Unit Test Desteği: Entity Framework, unit test yazarken veritabanı bağımlılıklarını yönetmeyi kolaylaştırır. In-memory veritabanı gibi özelliklerle testleri daha etkili bir şekilde gerçekleştirebilirsiniz.

Entegrasyon Kolaylığı: .NET platformuyla entegre olduğu için, Entity Framework kullanmak, diğer .NET teknolojileriyle uyumluluğu artırır.

Güçlü Veri Modelleme: Entity Framework, güçlü bir veri modelleme yeteneği sunar. Veritabanındaki karmaşık yapıları, sınıflar ve ilişkiler üzerinden nesne modellerine dönüştürme konusunda etkilidir.

Entity Framework, bu avantajlarıyla birlikte .NET geliştiricilerine veritabanı işlemlerini daha kolay, hızlı ve etkili bir şekilde gerçekleştirme imkanı sunar. Ancak, her avantajın yanında bazı durumlar için dikkate alınması gereken dezavantajlar da olabilir, bu nedenle kullanmadan önce projenizin ihtiyaçlarına uygun olduğundan emin olmalısınız.

Entity Framework Dezavantajları nelerdir?

Performans Sorunları:

Entity Framework, özellikle büyük ve karmaşık veritabanları ile çalışırken performans sorunlarına neden olabilir. Özel optimizasyonlar ve dikkatli sorgu yazımı olmadan, sorguların veritabanında yetersiz performansa neden olma olasılığı vardır.

Veritabanı Tasarımı ve Optimizasyon Zorlukları:

Entity Framework kullanırken, veritabanı tasarımında ve optimizasyonunda bazı kısıtlamalarla karşılaşabilirsiniz. Özellikle, gelişmiş veritabanı optimizasyonlarını gerçekleştirmek istediğinizde, doğrudan SQL sorguları yazmak daha etkili olabilir.

Karmaşıklık ve Öğrenme Eğrisi:

Entity Framework, büyük projelerde kullanıldığında karmaşık hale gelebilir ve öğrenme eğrisi yüksek olabilir. Uygulama gereksinimlerinize uygun şekilde yapılandırma ve kullanma konusunda dikkatli olmak önemlidir.

Yetersiz Özel Mappings:

Entity Framework, bazı durumlarda özel veritabanı sütunlarını sınıflara bağlamakta veya özel bir şekilde sorgu oluşturmada yetersiz kalabilir. Bu durumda, stored procedure veya doğrudan SQL kullanma ihtiyacı doğabilir.

Çalışma Zamanı Performansı:

Entity Framework, veritabanı sorgularını çalışma zamanında oluşturduğu için, bu durum performans maliyetine neden olabilir. Özellikle, aynı sorguların tekrar tekrar kullanıldığı durumlarda bu maliyet artabilir.

Gelişmiş Veri Kontrolü Eksikliği:

Bazı durumlarda, Entity Framework'un otomatik olarak oluşturduğu SQL sorguları, gelişmiş veri kontrolü veya özel işlemler gerçekleştirme ihtiyacını karşılamayabilir. Bu durumda, doğrudan SQL veya stored procedure kullanma ihtiyacı doğabilir.

Veritabanı Bağımlılığı:

Entity Framework, genellikle bir veritabanı motoruna bağımlı olarak çalışır. Bu durum, veritabanı motorunu değiştirmek istediğinizde uyumluluk sorunlarına neden olabilir.

Zayıf Offline Desteği:

Entity Framework, offline çalışma senaryolarını desteklemede bazı zorluklar yaşayabilir. Mobil uygulamalar veya sınırlı ağ bağlantıları olan senaryolarda offline desteği eksik olabilir. Bu dezavantajlar, Entity Framework'ün kullanımı sırasında karşılaşılabilecek potansiyel sorunlardır. Her projenin ihtiyaçları farklı olduğu için, Entity Framework kullanımı öncesinde avantajları ve dezavantajları dikkatlice değerlendirmek önemlidir.

EF Core hangi veri tabanlarında çalışır?

Entity Framework Core, birden çok veritabanı yönetim sistemini destekler. Ancak, desteklenen veritabanları zamanla değişebilir ve geliştikçe yeni sürümlerle yeni veritabanları eklenir. Entity Framework Core 5.0 sürümüne kadar, aşağıdaki veritabanı sistemleri desteklenmiştir:

Microsoft SQL Server: Entity Framework Core, Microsoft SQL Server ile entegrasyon sağlar. Microsoft.EntityFrameworkCore.SqlServer paketini kullanarak SQL Server ile etkileşime geçebilirsiniz.

MySQL: MySQL veritabanı için Entity Framework Core sürücüsü mevcuttur. MySql.Data.EntityFrameworkCore paketi, MySQL veritabanı ile çalışmak için kullanılabilir.

PostgreSQL: Entity Framework Core, PostgreSQL ile uyumludur. Npgsql.EntityFrameworkCore.PostgreSQL paketi, PostgreSQL veritabanı ile etkileşim için kullanılır.

SQLite: Entity Framework Core, hafif ve yerel bir veritabanı olan SQLite ile kullanılabilir. Microsoft.EntityFrameworkCore.Sqlite paketi, SQLite veritabanı ile çalışmak için kullanılır.

In-Memory Database: Entity Framework Core ayrıca, hafızada çalışan geçici bir veritabanı olarak kullanılabilecek bir in-memory veritabanını destekler. Bu, testler ve hızlı prototipler oluşturmak için kullanışlıdır.

Bu, Entity Framework Core'un desteklediği bazı veritabanlarıdır, ancak gelişen zaman içinde yeni sürümlerle yeni veritabanları eklenmiş olabilir. Projelerinizin ihtiyaçlarına ve kullanılan teknolojilere bağlı olarak, uygun veritabanı sürücüsünü seçmek önemlidir.