

Movie Store WebApi Yazılması

Bu proje kapsamında bir **Movie Store** uygulamasının servislerinin yazılması beklenmektedir.

Proje Yapısı

1.Bir filmin sahip olması gereken özellikleri şu şekildedir:

- Film Adı
- Film Yılı
- Film Türü
- Yönetmen
- Oyuncular
- Fiyat

Yönetmenler ve oyuncular ayrı tutulmalıdır. Bir oyuncu aynı zamanda yönetmen de olabilir, ama bunlar iki ayrı yapıdır.

2.Oyuncuların sahip olması gereken özellikler temelde şu şekildedir :

- İsim
- Soyisim
- Oynadığı filmler

Bir filmde birden fazla oyuncu oynayabilir, bir oyuncunun da birden fazla filmi olabilir.

3.Yönetmenlerin sahip olması gereken özellikler temelde şu şekildedir:

- İsim
- Soyisim
- Yönettiği filmler

4.Uygulama içerisinde bir de Customer rolü olmalıdır. Özellikleri ise temelde şu şekildedir:

- İsim

- Soyisim
- Satın aldığı filmler
- Favori türler

Müşterinin birden fazla favori film türü olabilir. Satın aldığı bir türü tekrar satın alabilir. Buraya bir kısıtlama koymaya gerek yok.

5.Customer için bir login endpoint'i yaratılmalıdır. Yetkisiz kullanıcı uygulama içerisinde satın alma işlemi yapamamalı.

6.Müşterilerin satın aldıkları filmler siparişlerim gibi bir yapı içerisinde tutulmalıdır. Temel özellikleri ise şu şekilde olmalıdır:

- Satın alan müşteri
- Satın alınan film
- Fiyat
- Satın alma tarihi

Uygulama Gereksinimleri

1.Film Ekleme/Silme/Güncelleme/Listeleme

2.Müşteri Ekleme/Silme

3.Oyuncu Ekleme/Silme/Güncelleme/Listeleme

4.Yönetmen Ekleme/Silme/Güncelleme/Listeleme/

5.Film Satın Alma

- Müşteri istediği bir filmi uygulama üzerinden satın alabilir.

6.Müşteri bazlı satın alma verisinin listelenmesi. Satın alınan filmler daha sonra sistemden silinebilir. Bu sipariş datasını etkilememeli. Bu durumu sağlamak için film verileri hard olarak silinmemelidir. Bir Aktif-Pasif özelliği ile yönetilebilir.

7.Film türleri uygulama çalıştırıldığından varsayılan olarak yaratılabilir. Servisler ile yönetilmesine gerek yoktur.

Teknik Gereksinimler

1.Entity objeleri input/output olarak kullanılmamalı.

2.Gereken yerlerde model ve Dto'lar kullanılmalıdır. Automapper kütüphanesi yardımıyla objeler birbirine dönüştürülebilmelidir.

3.Controller içerisinde servis işleri yapılmamalıdır.

4.Her servisin bir validation sınıfı olmalıdır. Uygulama sıkı kurallar ile korunmalı, olası tüm validasyonlar yazılmalıdır.

5.Exception ve loglama altyapısı middleware kullanılarak yazılmalıdır.

- Uygulama yapılan her request ve response console'a yazılarak loglanmalıdır.
- Uygulama hata aldığı yada validasyon hatası fırlattığında anlamlı bir şekilde bu hatalar console'a log olarak yazdırılmalıdır.
- Loglama yöntemi uygulama içerisine yayılmamalıdır. Gerek duyulduğunda sadece bir noktada değişiklik yaparak loglama şekli değiştirilebilmeli.

6.Proje içerisinde bağımlılık (dependency) yaratılmamasına dikkat edilmelidir. Gerekli noktalarda DI Container ile Dependency Injection kullanılarak bağımlılıklar tek bir noktadan yönetilmelidir.

7.Projeye temel seviyede bir Authentication ve Authorization yapısı implemente edilmelidir. Satın alma endpoint'i sadece müşteri tarafından kullanılacak bir endpoint olmalıdır.

8.Projenin birim testleri eksiksiz şekilde yazılmalıdır. Tüm testler hatasız çalışmalıdır.

9.Silme servislerinde veri tutarlılığı dikkate alınmalıdır. Diğer tablolarda ilişkili datası bulunan kayıtlar silinemez.