**Shopping Project**

Teslim Tarihi: 24.01.2023

Adı: Mustafa Alp YANIKOĞLU

Email: malpyanikoglu@gmail.com

* **Giriş**

Projenin amacı; Alışveriş yapılabilen mobil veya web uygulaması için backend kısmını geliştirmek

Auth Controller içerisindeki metotlar;

**Register :**

* Email, ad, soyad,şifre,telefon,adres alanları girilerek çalışır.
* Müşteri kayıt olurken email benzersiz olmalıdır
* Müşteri kayıt olduğunda müşteriye otomatik bir sepet ve cüzdan oluşturulur
* Müşteri kayıt olduğunda, müşteriye otomatik olarak member rolü atanır
* Müşterinin şifresi veri tabanında hash üretilerek tutulur

**Login :**

* Müşteri sisteme giriş yaparken email ve şifresi gereklidir ve bu bilgiler veri tabanındaki bilgiler ile eşleşmelidir
* Müşteri giriş yaptığında jwt ile token üretilir token içerisinde kullanıcının rolleri, id, ad soyad bilgileri tutulur

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Change Password :**

* Müşteri şifresini değiştirebilmesi için sisteme giriş yapmalı
* Şifre güncelleme sırasında müşteriden; mail, yeni şifresi ve yeni şifrenin tekrar girilmesi istenir
* Girilen yeni şifre ve şifre tekrarı eşleşmelidir

User Controller içerisindeki metotlar;

**From Auth :**

* Sisteme giriş yapılmadan şifre değiştirebilme işlemidir

**Delete User:**

* Girilen UserId ile kullanıcı silinir.
* Girilen UserId veri tabanında yok ise sistem bunu bildirir

**Update User:**

* Kullanıcının şifre hariç bilgilerinin günellenmesi için kullanılır
* Girilen UserId veri tabanında yok ise sistem bunu bildirir

**GetById :**

* Girilen UserId ile kullanıcı bilgisi listelenir.
* Girilen UserId veri tabanında yok ise sistem bunu bildirir

**GetList:**

* Kullanıcıların bilgileri listelenir
* Bu metot bütün entity’ler için aynı görevi görür

**GetList/ByDynamic:**

* Herhangi bir filtre ekleyerek kullanıcıları listelemenizi sağlar
* Örneğin;

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

* Burada kullanıcı Id’lerini küçükten büyüğe sıralar ve email bilgisi 11.satır veya 17. satırda ki değer ile işleşen kullanıcı bilgilerini listeler
* Bu metot bütün entity’ler için aynı görevi görür

UserCart Controller içerisindeki metotlar;

**Add :**

* Eğer herhangi bir işlem sonucu kullanıcının sepeti silindi ise admin kullanıcıya sepet oluşturabilir
* Sepeti olan bir kullanıcıya ikinci bir sepet oluşturulması engellenmiştir

**Delete :**

* Girilen UserCartId ile kullanıcı sepetini silmek için kullanılır
* Girilen UserCartId veri tabanında yok ise sistem bunu bildirir

**Update :**

* Kullanıcı sepeti ile eşleşen kullanıcı değiştirilebilir
* Girilen UserCartId veri tabanında yok ise sistem bunu bildirir
* Girilen UserId veri tabanında yok ise sistem bunu bildirir

**GetList-GetById-GetListByDynamic :**

* Kullanıcı sepetinin kime ait olduğunu gösterir
* UserId yerine kullanıcının adı ve soyadı kullanılır

Purse Controller içerisindeki metotlar;

**Add :**

* Kullanıcıya cüzdan oluşturulur
* Kullanıcıya ikinci bir cüzdan oluşturulması engellenmiştir

**Delete :**

* Girilen PurseId ile kullanıcı cüzdan bilgisi silinir
* Girilen PurseId veri tabanında yok ise sistem bunu bildirir

**Update :**

* Kullanıcı sepeti ile eşleşen kullanıcı değiştirilebilir
* Girilen PurseId veri tabanında yok ise sistem bunu bildirir

**GetList-GetById-GetListByDynamic :**

* Kullanıcı cüzdan bilgilerini gösterir
* UserId yerine kullanıcının adı ve soyadı kullanılır

Operation Claim Controller içerisindeki metotlar;

**Add :**

* Rol üretme işlemidir fakat yetki işlemleri kod ile sisteme gömüldüğü için herhangi bir üretilen rolün işlevi olmaz gerekli olan roller veri tabanında migration işlemi ile üretilmiştir.
* Bir hata sonucu rol veri tabanından silinmiş ise rolün kod içinde ‘admin’ olarak yazıldığı için bu isime ait bir rol eklenmesi yeterli olacaktır sistem sorunsuz çalışmaya devam eder.

**Delete :**

* Girilen OperationClaimId ile rol bilgisi silinir
* Girilen OperationClaimId veri tabanında yok ise sistem bunu bildirir

**Update :**

* Girilen bilgiler ile rol bilgisi güncellenir.
* Girilen OperationClaimId veri tabanında yok ise sistem bunu bildirir

**GetList-GetById-GetListByDynamic :**

* Rol bilgilerini gösterir

UserOperationClaim Controller içerisindeki metotlar;

**Add :**

* Kullanıcıya rol atama işlemi için kullanılır.
* UserId ve OperationClaimId girilerek kullanıcı ve rol bilgisi eşleşir.
* Girilen bilgilerden herhangi biri veri tabanında yok ise sistem bunu bildirir.

**Delete :**

* Girilen UserOperationClaimId ile kullanıcı rol bilgisi silinir.
* Girilen UserOperationClaimId veri tabanında yok ise sistem bunu bildirir

**Update :**

* Girilen bilgiler ile kullanıcı rol bilgisi güncellenir.
* Girilen UserOperationClaimId veri tabanında yok ise sistem bunu bildirir
* UserId ve OperationClaimId girilerek kullanıcı ve rol bilgisi eşleşir.

**GetList-GetById-GetListByDynamic :**

* Kullanıcı rol bilgilerini gösterir
* UserId yerine kullanıcının adı ve soyadı kullanılır
* OperationClaimId yerine rol adı ve rolün tanımı gösterilir

Category Controller içerisindeki metotlar;

**Add :**

* Sisteme kategori eklenir.
* Kategori adı benzersiz olmalıdır eğer girilen kategori adı veri tabanında var ise sistem bunu bildirir.

**Delete :**

* Girilen CategoryId ile kategori bilgisi silinir.
* Girilen CategoryId veri tabanında yok ise sistem bunu bildirir.

**Update :**

* Girilen bilgiler ile kategori güncellenir.
* Girilen CategoryId veri tabanında yok ise sistem bunu bildirir.

**GetList-GetById-GetListByDynamic :**

* Kategori bilgilerini gösterir

**GetByCategoryName :**

* Girilen kategori adına göre kategori bilgilerini gösterir.
* Girilen kategori adı veri tabanında yok ise sistem bunu bildirir.

Product Controller içerisindeki metotlar;

**Add :**

* Sisteme ürün eklenir.
* Ürün adı benzersiz olmalıdır eğer girilen ürün adı veri tabanında var ise sistem bunu bildirir.
* Ürün miktarı ve birim fiyatı sıfır dan küçük girilemez.

**Delete :**

* Girilen ProductId ile ürün bilgisi silinir.
* Girilen ProductId veri tabanında yok ise sistem bunu bildirir.

**Update :**

* Girilen bilgiler ile ürün bilgisi güncellenir.
* Girilen ProductId veri tabanında yok ise sistem bunu bildirir.
* Ürün miktarı ve birim fiyatı sıfır dan küçük girilemez.

**GetList-GetById-GetListByDynamic :**

* Ürün bilgilerini gösterir.
* CategoryId yerine kategori adı gösterilir.

**GetByProductName :**

* Girilen ürün adına göre ürün bilgilerisini gösterir.
* Girilen ürün adı veri tabanında yok ise sistem bunu bildirir.
* CategoryId yerine kategori adı gösterilir.

**GetListByCategoryName :**

* Girilen kategori adına göre ürün bilgilerini gösterir.
* Girilen kategori adı veri tabanında yok ise sistem bunu bildirir
* CategoryId yerine kategori adı gösterilir.

Order Controller içerisindeki metotlar;

**Add :**

* UserCartId => hangi sepete ekleneceğini,
* ProductId => hangi ürünün ekleneceğini,
* Quantity => üründen kaç adet ekleneğini belirtir.
* UserCartId ve ProductId veri tabanında yok ise sistem bunu bildirir.
* Quantity, ürün adeti sıfırdan küçük girilemez.
* Sipariş oluşturulmadı ise sipariş oluşturulur ve sepete ürün eklenir, sepette ürün var ise ve sepet onaylanmadıysa ürün direk sepete eklenir
* Sepette aynı ürün mevcut ise miktarı ve ürün tutarı güncellenir.
* Sipariş oluşturulduğu sırada siparişin oluşturulduğu tarih veri tabanında tutulur
* Sipariş oluşturulduğu sırada siparişe, 6 karakterli string olan hexadecimal bir ifade atanır(OrderNumber)
* OrderNumber yani sipariş numarası benzersiz olmalıdır eğer üretilen sipariş numarası veri tabanında var ise sistem yeni bir orderNumber üretir.

**Delete :**

* Girilen OrderId ile sipariş bilgisi silinir.
* Sipariş silindikten sonra OrderDetail yani sipariş içindeki ürünler de OrderDetail tablosundan silinir.

**Update :**

* Sipariş güncelleme işlemi.
* Girilen OrderId veri tabanında yok ise sistem bunu bildirir.
* Sipariş başka bir sepete taşınacak ise girilen UserCartId veri tabanında yok ise işlem durdurulur.

**Confirm Order:**

* Oluşturulan siparişi onaylamak için kullanılır.
* Sipariş onaylamak için OrderId bilgisi gereklidir ve veri tabanında yok ise sistem bunu bildirir.
* Onaylanmış bir sipariş ikinci kez onaylanması engellenmiştir.
* Sipariş tutarı hesaplanır ve kullanıcının bakiyesi kontrol edilir yetersiz ise işlem durdurulur.
* Sepette ürün yok ise sipariş onaylama işlemi gerçekleşmez

**GetList-GetById-GetListByDynamic :**

* Siparişin kime ait olduğunu (adı, soyadı, adresi)
* Sipariş bilgilerini (sipariş numarası, sipariş oluşturulma ve onaylanma tarihleri, siparişin onaylanıp onaylanmadığı gösterilir)

OrderDetail Controller içerisindeki metotlar;

**Add :**

* OrderId => hangi siparişe ekleneceğini,
* ProductId => hangi ürüne ekleneceğini
* Quantity => kaç adet ekleneceğini belirtir
* Girilen OrderId ve ProductId veri tabanında yok ise işlem durdurulur.
* Sipariş onaylandıysa siparişe ürün eklenmez.
* Sepette ürün varsa ürünün miktarı ve ürün tutarı güncellenir.

**Delete :**

* Girilen OrderDetailId ile siparişde ki ürün silinir.
* Girilen OrderDetailId veri tabanında yok ise sistem bunu bildirir.

**Update :**

* Adminin OrderDetail güncellemesi için kullanılır.
* Quantity, yani ürün miktarı sıfır girilemez yani admin sepetteki bir ürünü bu şekilde silemez.
* Girilen ProductId ve OrderId sıfırdan büyük olmalı ve veri tabanında olmalı yok ise sistem bunu bildirir.

**Update Product Quantity:**

* Id => orderDetailId yani siparişteki hangi ürün,
* Quantity => güncellenecek ürün miktarı.
* Quantity sıfır olabilir eğer sıfırsa ürün siparişten silinir.
* Siparişte ki ürün miktarı ve ürün tutarı güncellenir.
* Sipariş onaylandıysa ürün miktarı güncelleme gibi bir işlem gerçekleşmez.

**GetList-GetById-GetListByDynamic :**

* Alınan ürün bilgileri (Kategori adı, ürün adı, birim fiyatı, kaç adet alındığı, ürünün tutarı),
* Sipariş bilgileri, (sipariş oluşturulma tarihi, sipariş numarası, siparişin onaylandığı tarih)
* Kullanıcı bilgileri, (adı, soyadı, adresi) bilgileri gösterilir.

**GetList-GetById-GetListByDynamic :**

* Alınan ürün bilgileri (Kategori adı, ürün adı, birim fiyatı, kaç adet alındığı, ürünün tutarı),
* Sipariş bilgileri, (sipariş oluşturulma tarihi, sipariş numarası, siparişin onaylandığı tarih)
* Kullanıcı bilgileri, (adı, soyadı, adresi) bilgileri gösterilir.

**GetListPastOrder:**

* Kullanıcının geçmiş siparişlerini listeler
* Bu işlem için UserId kullanılır eğer kullanıcı veri tabanında yok ise sistem bunu bildirir.
* Kullanıcının geçmişte bir siparişi yoksa işlem durdurulur.
* Alınan ürün bilgileri (Kategori adı, ürün adı, birim fiyatı, kaç adet alındığı, ürünün tutarı),
* Sipariş bilgileri, (sipariş oluşturulma tarihi, sipariş numarası, siparişin onaylandığı tarih, siparişlerin toplam tutarı)
* Kullanıcı bilgileri, (adı, soyadı, adresi) bilgileri gösterilir.

**GetListMyUserCartByUserId:**

* Kullanıcının sepetinde olan siparişleri listeler.
* Bu işlem için UserId kullanılır eğer kullanıcı veri tabanında yok ise sistem bunu bildirir.
* Kullanıcının sepetinde ürün yok ise işlem durdurulur.
* Alınan ürün bilgileri (Kategori adı, ürün adı, birim fiyatı, kaç adet alındığı, ürünün tutarı),
* Sipariş bilgileri, (sipariş oluşturulma tarihi, sipariş numarası, siparişin onaylandığı tarih, siparişlerin toplam tutarı)
* Kullanıcı bilgileri, (adı, soyadı, adresi) bilgileri gösterilir.

**GetListByOrderNumberOrderDetail:**

* Girilen OrderNumber’a göre order bilgilerini listeler
* OrderNumber veri tabanında yok ise sistem bunu bildirir.
* Bu işlem için UserId kullanılır eğer kullanıcı veri tabanında yok ise sistem bunu bildirir.
* Kullanıcının geçmişte bir siparişi yoksa işlem durdurulur.
* Alınan ürün bilgileri (Kategori adı, ürün adı, birim fiyatı, kaç adet alındığı, ürünün tutarı),
* Sipariş bilgileri, (sipariş oluşturulma tarihi, sipariş numarası, siparişin onaylandığı tarih, siparişlerin toplam tutarı)
* Kullanıcı bilgileri, (adı, soyadı, adresi) bilgileri gösterilir.

**2.Geliştirme ve Çalıştırma Ortamları**

Geliştirmek ,test etmek ve çalıştırmak için Visual Studio Code editörünü kullandım.

**3.Tech Stack**

* .Net 6
* Entity Framework
* Web Api
* Swagger
* Data Transfer Object
* Automapper
* Unit of Work
* Jwt
* Fluent Validation
* Result Structure
* Generic Repository
* Dependency Injection
* Exception Handling
* MediatR
* AutoMapper

**4.Github**

* Proje ilerletilirken github kullanımına özen gösterilmiştir ve herhangi bir çakışma olmaması için 2 branch kullanılmıştır
* Değiştirilen kodlar öncelikle ‘may-dev-feat-[yapılan değişiklik tanımı]’ branch’e atılır , daha sonra githubta may-master-feat-[yapılan değişiklik tanımı] adında bir branch oluşturulur ve bu branche Pull Request oluşturulur
* Daha sonra ise may-master-feat-[yapılan değişilik tanımı] branchinden master’a atılır böylece master’da kod çakışmasını engellemiş oluruz
* Projenin test edilmesi için postman kullanılmıştır ve bunun linki de github da bulunur
* Github şuan da private proje süresi dolduktan sonra public yapabilirim ve erişebilirsiniz yada isterseniz şirketten birisini projeye ekleyebilirim.
* Github Linki; https://github.com/mustafaalpyanikoglu/CSharp-ShoppingProject