

Trendyol Checkout Case

Bu case'de e-ticaret kullanan herkesin aşına olduğu sepet uygulamasının bir benzerini sizden geliştirmenizi bekliyoruz.

Uygulamanıza bir database bağlamak yada framework kullanmak zorunda değilsiniz. Ancak bununla alakalı bir kısıtlama da bulunmuyor. Aşağıdaki örneklerde Input ve output şemaları verilmiştir bunlara uygun REST tasarım yapmanız gerekmektedir.

Burada önemli olan kriterler, **clean code, SOLID, design patterns, object oriented design, unit testing** hatta tecrübenize göre **DDD** kullanarak yeteneklerinizi göstermeniz. Case'i yazarken testable ve extendible, mümkünse **TDD** pratiklerini kullanarak yazmaya özen göstermeniz faydalı olacaktır.

Case'in bütününe okumadan kodlamaya **başlamamanız** faydalı olacaktır. Kodlamaya başlamadan önce bütün case'i okuyup anladığınıza emin olun.

Aşağıda business'ı açıklayan anahtar konseptler ve davranışları verilmiştir.

Cart

Bütün nesneleri içinde bulunduran bir nesnedir. Bütün nesneler Cart kümesine uygulanır. Bir sepette maksimum **10** adet benzersiz item (VasItem dahil değil) bulunabilir. Toplam ürün adedi **30**'dan fazla olamaz. Cart'ın toplam tutarı **500.000** TL'den fazla olamaz.

Item

Cart üzerinde bulunan ürünlerdir. Cart'a item eklenip, çıkartılabilir. Cart item'ları sıfırlanabilir. Itemlar birden fazla türde olabilir **VasItem, DefaultItem, DigitalItem** gibi. Item birden fazla adette eklenebilir. Bir itemin maksimum eklenme adedi **10**'dur. Her item'in fiyatı farklı şekilde belirlenir ve input olarak uygulamaya verilir. Sepetteki itemların satıcı ve kategori ID'leri vardır.

DigitalItem

DigitalItem olan bir cart'a sadece digital item eklenebilir. Digital item; steam card, bağış kartı vb. gibi itemlerdir. DigitalItem'ın maksimum eklenme adedi **5**'tir. d. CategoryID'si **7889** olan itemlar DigitalItem olarak tanımlanır. Bu CategoryID ile başka türden bir Item tanımlanamaz.

DefaultItem

Default item geleneksel e-ticarete kullanan klasik ürünlerdir. Örneğin, t-shirt, cep telefonu, deterjan gibi. DefaultItem'a eklenen VasItem'ın fiyatı DefaultItem'dan yüksek olamaz.

VasItem

Value Added Service item sigorta, montaj gibi hizmetleri ifade eder. **Bu ürünler fiziksel bir ürünü değil belli bir ürüne bağlı hizmeti ifade eder ve tek başlarına bir anlamı yoktur.** Bu nedenle sadece Mobilya (CategoryID: **1001**) ve Elektronik(CategoryID: **3004**) kategorisindeki default itemlara **alt item** olarak eklenebilir. Bir DefaultItem'a maksimum **3** adet VASItem eklenebilir. VasItem'ın CategoryID'si **3242**'dir. VasItem'ların satıcı ID'si **5003**'tur. Satıcı ID'si **5003** olmayan VasItemlar tanımlanamaz. Satıcı ID'si **5003** olan baska türde bir Item türü yoktur.

Promotion

Cart üzerinde belirli itemlara veya tüm cart'a indirim uygulayan entity.

SameSellerPromotion

SameSellerPromotion ID'si **9909** dur. Cart'taki item'ların satıcısı aynı ise(VasItem dahil değil) Cart'a SameSellerPromotion tanımlanır cart'ın toplam tutarına **%10** indirim uygulanır.

CategoryPromotion

CategoryPromotion ID'si **5676** dur. Cart'ın üzerinde bulunan ve CategoryID'si **3003** olan itemlara **%5** indirimli promotion tanımlanır, Cart'ın toplam tutarına uygulanır.

TotalPricePromotion

TotalPricePromotion ID'si **1232**'dir. Eğer cart'in fiyatı **5.000 TL**'den az ise (**5.000** dahil değil) **250 TL**, **5.000 TL - 10.000 TL**'e kadarsa (**10.000** dahil değil) **500 TL**, **10.000 TL -50.000 TL**'e kadarsa (**50.000** dahil değil) **1000 TL**, **50.000 TL** ve yukarıda ise **2000 TL** indirim uygulanır.

Cart'a birden fazla promotion uygulanmaz. Eğer birden fazla promotion'a hit eden bir cart varsa, müşteri için en avantajlı olan promotion uygulanır(En avantajlı promosyon, türünden bağımsız müşteriye toplamda en fazla indirim uygulayan promosyondur).

Json

Aşağıda uygulamanızın Rest input ve outputları gösterilmiştir.

Action 1

Add Item

Input { "itemId":int, "categoryId":int, "sellerId":int, "price":double, "quantity":int }

Output { "result":boolean, "message": string }

Action 2

Add vasitem to item

Input { "itemId": int, "vasItemid":int, "categoryId": int, "sellerId":int, "price":double, "quantity":int }

Output { "result":boolean, "message": string }

Action 3

Remove Item

Input { "itemId":int }

Output { "result":boolean, "message": string }

Action 4

Reset Cart

Output { "result":boolean, "message": string }

Action 5

Display Cart

Output

```
{"result":boolean, "message":{"items":[ty.item], "totalPrice":double, "appliedPromotionId":int, "totalDiscount":double}}
```

```
ty.item -> {itemId: int, categoryId: int, sellerId:int, price:double, quantity:int, "vasItems":[ty.vasItem]}
```

```
ty.vasItem -> {"vasItemId":int, "categoryId": int, "sellerId":int, "price":double, "quantity":int}
```