

# Trendyol Case Study Solution

Dosya içerisinde ki belirli formatta commandları okuyan ve verilen formatta tek tek işleyerek sepet işlemlerini gerçekleştirip sonuçları bir dosyaya kaydeden Console uygulaması geliştirildi.

## Teknoloji:

In-Memory database kullanıldı.

Veri - Model bağlantısı için Entity Framework Core 7.0 versiyon kullanıldı.

Business test caseleri için NUnit test ile Happy path ve edge caseler düşünülerek testler yazıldı.

▼ □ TrendyolCheckoutCaseTests	99%	1/325
▼ {} TrendyolCheckoutCaseTests	99%	1/325
> {} PromotionServiceTests	100%	0/69
> {} CartServiceTests	100%	0/220
> {} Command	97%	1/36

Loglama yapıldı.

DDD mimari uygun şekilde klasör yapısı uygulanmaya çalışıldı.

Sabit değişkenler, servis response mesajları tek bir yerde toplanarak yönetimi ve okunabilirliği kolaylaştırıldı.

Solid ilkelerine uymaya çalışılarak, methodlara ve classlara tek sorumluluk verilmeye , bağımlılıkların azaltılması ve constructorlar üzerinden yönetilmesine dikkat edildi.

Modellerde Domain model ve Application model ayrımı yapıldı.

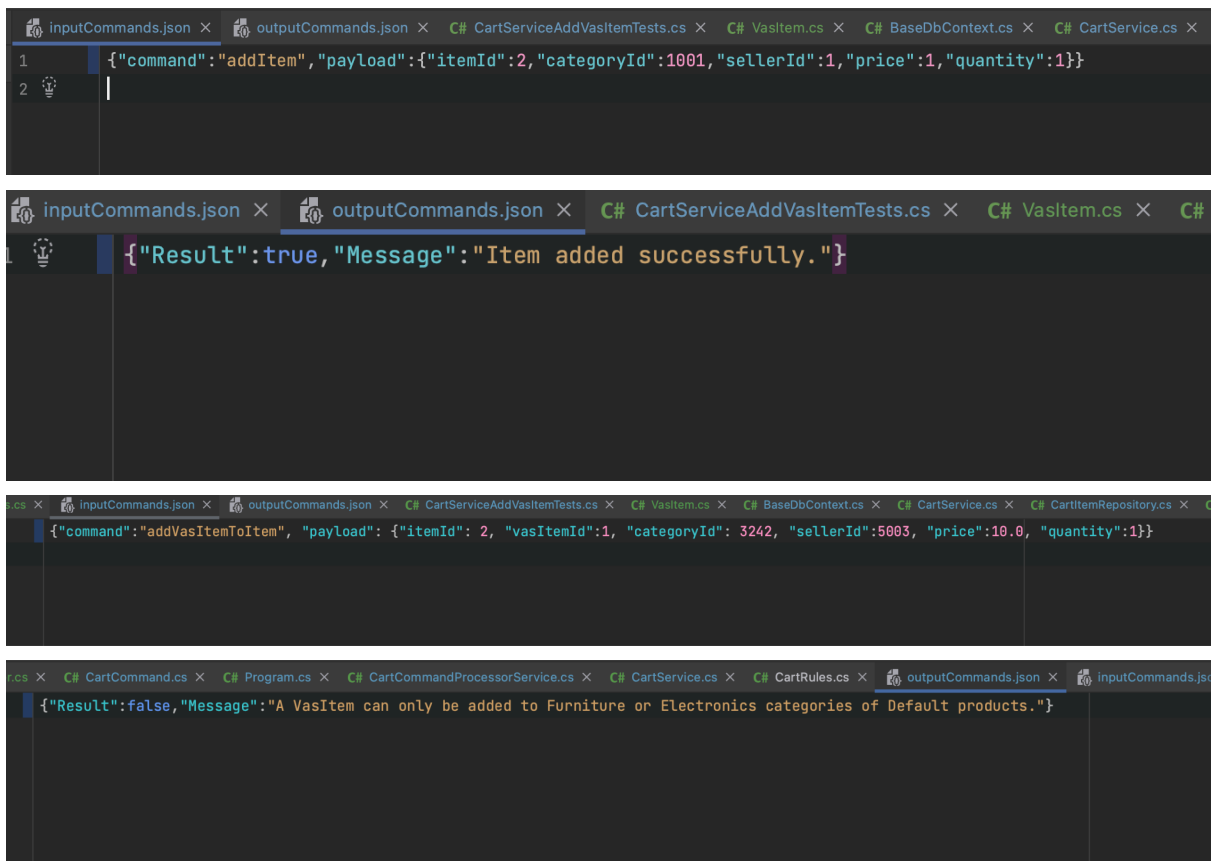
## Proje Test Adımları :

1. Öncelikle dosya yolu kontrol edilmelidir. Json dosyalar Repo içerisinde oluşturuldu. Dosya uzantılar FilePathCons.cs classın da tanımlanmıştır.
2. Input dosyasına yapılmasını istediğimiz işlemi yazarız. Verilen Input Command komutu girildikten sonra ilk komut işlenir. İşlem tamamlandıktan sonra bir sonra ki command için önce input dosyasında command dosyası güncellenir güncelleme yapıldıktan sonra Console ekranında Devam etmek istediğine dair "Y" girişi yapılır ve bir sonra ki command işlenir.
3. Her command sonrası output.json dosyası komut sonrası dönen response'u basar.

## Commands

```
{"command": "addItem", "payload": {"itemId": int, "categoryId": int, "sellerId": int, "price": double, "quantity": int}}  
{"command": "addVasItemToItem", "payload": {"itemId": int, "vasItemId": int, "categoryId": int, "sellerId": int, "price": double, "quantity": int}}  
{"command": "removeItem", "payload": {"itemId": int}}  
{"command": "resetCart"}  
{"command": "displayCart"}
```

## Örnek Ekran Görüntüleri



```
C# CartRules.cs × InputCommands.json × outputCommands.json × C# BaseDbContext.cs × C# CartItem.cs × C# CartService
{
  "Result": true, "Message": {
    "Items": [
      {
        "ItemId": 2,
        "CategoryId": 1001, "SellerId": 1, "Price": 1.0, "Quantity": 1},
        "TotalPrice": 21.0,
        "TotalQuantity": 1,
        "AppliedPromotionId": null,
        "VasItems": [
          {
            "VasItemId": 1, "CategoryId": 3242, "SellerId": 5003, "Price": 10.0, "Quantity": 1},
            {
              "VasItemId": 2, "CategoryId": 3242, "SellerId": 5003, "Price": 10.0, "Quantity": 1}
          ]
        }
      ]
    }
  }
}
```

```
inputCommands.json × outputCommands.json × C# CartServiceAddVasItemTests.cs
1 {
2   "Result": true,
3   "Message": {
4     "Items": [
5       {
6         "ItemId": 2,
7         "VasItems": [
8           {
9             "VasItemId": 1,
10            "CategoryId": 3242,
11            "SellerId": 5003,
12            "Price": 10.0,
13            "Quantity": 1
14          }
15        ],
16        "CategoryId": 1001,
17        "SellerId": 1,
18        "Price": 1.0,
19        "Quantity": 1
20      }
21    ],
22    "TotalPrice": 11.0,
23    "TotalQuantity": 1,
24    "AppliedPromotionId": null
25  }
26 }
```

```
"Items": [  
  {  
    "ItemId": 2,  
    "VasItems": [],  
    "CategoryId": 1000,  
    "SellerId": 1,  
    "Price": 10.0,  
    "Quantity": 4  
  },  
  {  
    "ItemId": 3,  
    "VasItems": [  
      {  
        "VasItemId": 1,  
        "CategoryId": 3242,  
        "SellerId": 5003,  
        "Price": 10.0,  
        "Quantity": 1  
      }  
    ],  
    "CategoryId": 1001,  
    "SellerId": 1,  
    "Price": 1.0,  
    "Quantity": 1  
  }  
],  
"TotalPrice": 111.7,  
"TotalQuantity": 5,  
"AppliedPromotionId": 9909
```