# Test Senaryoları AI Karşılaştırması

Bu dokümandaki senaryolar AI araçlarına verilip alınan çıktılar paylaşılmıştır.

## Senaryo 1:

Elimde bir rest servis var. Burada çok fazla hatalı istek alıyorum. Sana json formatındaki request yapısını ve servisime ait teknik dokümanı iletiyorum ve bana detaylı test senaryoları üretmeni istiyorum. Bu senaryolar için kısa açıklamalar ile birlikte doldurulmuş request bilgisine ihtiyacım var. Olabildiğince fazla senaryo üret teknik dokümana göre her durumu değerlendir.

Teknik doküman içeriğim:

ad: Müşteri adının tutulduğu alandır. Boş veya null olmamalıdır. Rakam bulunduramaz. " ve ? karakterlerini alamaz.

ikinciAd: Müşteri ikinci adının tutulduğu alandır. Boş veya null olarak gönderilebilir. Eğer değer gönderildiyse değerde rakam olmamalıdır. " ve ? karakterleri alamaz.

soyad: Müşteri soyadının tutulduğu alandır. Boş veya null olmamalıdır. Rakam bulunduramaz. " ve ? karakterlerini alamaz.

varlikTuru: Gerçek kişi ise "G", Tüzel kişi ise "T" değerleri alabilir. Bunun dışındaki hiçbir karakteri alamaz.

tckn: varlikTuru alanı "G" ise boş veya null olmamalıdır. 11 karakter uzunluğunda olmalıdır. Sadece rakamlardan oluşabilir. varlikTuru alanı "T" ise boş veya null olmalıdır.

vkn: varlikTuru alanı "T" ise boş veya null olmamalıdır. 10 karakter uzunluğunda olmalıdır. Sadece rakamlardan oluşabilir. varlikTuru alanı "G" ise boş veya null olmalıdır.

tutar: Ödeme tutarının tutulduğu alandır. Sadece pozitif tam sayı değer alabilir. Başka değer alamaz.

urunId: Müşterinin ödeme yaptığı ürünün id değeridir. Sadece pozitif tam sayı değer alabilir. Başka değer alamaz.

Request yapım:

{

"ad": "string",

"ikinciAd": "string",

"soyad": "string",

"varlikTuru": "string",

"tckn": "integer",

"vkn": "integer",

"tutar": "integer",

"urunId": "integer"

}

---------------------------------------------------------------------------------------------------------------

Ücretsiz ChatGPT modeli kullanılmıştır. 9 Nisan 2025 – GPT 4o mini

Bu input sonrasında ChatGPT den biraz daha senaryosu üretmesi istenmiş ve çıktılar birleştirilmiştir.

### **ChatGPT**

**✅ Pozitif Test Senaryoları**

**1. Gerçek Kişi – Geçerli tüm alanlar**

**Açıklama:** Kurallara uygun olarak tüm alanlar doğru şekilde doldurulmuş.

{

"ad": "Ahmet",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "G",

"tckn": 12345678901,

"vkn": null,

"tutar": 1000,

"urunId": 12

}

**2. Gerçek Kişi – ikinciAd boş**

**Açıklama:** ikinciAd alanı boş olabilir.

{

"ad": "Ayşe",

"ikinciAd": "",

"soyad": "Kara",

"varlikTuru": "G",

"tckn": 23456789012,

"vkn": null,

"tutar": 500,

"urunId": 3

}

**3. Tüzel Kişi – Geçerli tüm alanlar**

**Açıklama:** varlikTuru "T", tckn boş, vkn geçerli.

{

"ad": "ABC",

"ikinciAd": null,

"soyad": "Holding",

"varlikTuru": "T",

"tckn": null,

"vkn": 1234567890,

"tutar": 2500,

"urunId": 45

}

**❌ Negatif Test Senaryoları**

**1. ad alanı boş**

**Açıklama:** ad alanı boş bırakılamaz.

{

"ad": "",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "G",

"tckn": 12345678901,

"vkn": null,

"tutar": 100,

"urunId": 5

}

**2. ad içinde rakam var**

{

"ad": "Ali5",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "G",

"tckn": 12345678901,

"vkn": null,

"tutar": 100,

"urunId": 5

}

**3. soyad null**

{

"ad": "Ali",

"ikinciAd": "Kemal",

"soyad": null,

"varlikTuru": "G",

"tckn": 12345678901,

"vkn": null,

"tutar": 100,

"urunId": 5

}

**4. ikinciAd içinde rakam var**

{

"ad": "Ali",

"ikinciAd": "K3mal",

"soyad": "Yılmaz",

"varlikTuru": "G",

"tckn": 12345678901,

"vkn": null,

"tutar": 100,

"urunId": 5

}

**5. varlikTuru geçersiz karakter**

{

"ad": "Ali",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "X",

"tckn": 12345678901,

"vkn": null,

"tutar": 100,

"urunId": 5

}

**6. Gerçek kişi ama tckn eksik**

{

"ad": "Ali",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "G",

"tckn": null,

"vkn": null,

"tutar": 100,

"urunId": 5

}

**7. Tüzel kişi ama vkn eksik**

{

"ad": "ABC",

"ikinciAd": null,

"soyad": "Holding",

"varlikTuru": "T",

"tckn": null,

"vkn": null,

"tutar": 500,

"urunId": 12

}

**8. tckn 11 karakterden kısa**

{

"ad": "Ali",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "G",

"tckn": 12345,

"vkn": null,

"tutar": 100,

"urunId": 5

}

**9. tutar negatif**

{

"ad": "Ali",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "G",

"tckn": 12345678901,

"vkn": null,

"tutar": -100,

"urunId": 5

}

**10. urunId 0**

{

"ad": "Ali",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "G",

"tckn": 12345678901,

"vkn": null,

"tutar": 100,

"urunId": 0

}

**11. ad içinde yasaklı karakter (")**

{

"ad": "Ali\"",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "G",

"tckn": 12345678901,

"vkn": null,

"tutar": 100,

"urunId": 5

}

**12. tckn alfanumerik**

{

"ad": "Ali",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "G",

"tckn": "12A45678901",

"vkn": null,

"tutar": 100,

"urunId": 5

}

**13. vkn alfanumerik**

{

"ad": "XYZ",

"ikinciAd": null,

"soyad": "Holding",

"varlikTuru": "T",

"tckn": null,

"vkn": "12345AB890",

"tutar": 300,

"urunId": 10

}

**14. ikinciAd içinde yasaklı karakter (?)**

{

"ad": "Mehmet",

"ikinciAd": "Al?i",

"soyad": "Kaya",

"varlikTuru": "G",

"tckn": 98765432109,

"vkn": null,

"tutar": 120,

"urunId": 8

}

**15. soyad içinde rakam**

{

"ad": "Fatma",

"ikinciAd": "Zeynep",

"soyad": "Yilmaz1",

"varlikTuru": "G",

"tckn": 12345678901,

"vkn": null,

"tutar": 100,

"urunId": 9

}

**16. tüm alanlar null**

{

"ad": null,

"ikinciAd": null,

"soyad": null,

"varlikTuru": null,

"tckn": null,

"vkn": null,

"tutar": null,

"urunId": null

}

**17. varlikTuru boş string**

{

"ad": "Ali",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "",

"tckn": 12345678901,

"vkn": null,

"tutar": 200,

"urunId": 6

}

**18. tutar ondalıklı**

{

"ad": "Ali",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "G",

"tckn": 12345678901,

"vkn": null,

"tutar": 100.50,

"urunId": 5

}

**19. urunId negatif**

{

"ad": "Ali",

"ikinciAd": "Kemal",

"soyad": "Yılmaz",

"varlikTuru": "G",

"tckn": 12345678901,

"vkn": null,

"tutar": 300,

"urunId": -1

}

**20. hem tckn hem vkn dolu**

{

"ad": "Test",

"ikinciAd": "Orta",

"soyad": "Kullanıcı",

"varlikTuru": "G",

"tckn": 12345678901,

"vkn": 1234567890,

"tutar": 150,

"urunId": 7

}

### KAI

## Senaryo 2:

Elimde rest bir servis var. Request ve response yapısı aşağıdaki gibi.

Request

{

"kimlikNo": "string"

}

Response

{

"bireyselKrediler": [

{

"kurumKodu": "string",

"kurumAdi": "string",

"tutar": "integer"

}

]

}

Bu servis request headerdaki "kurumKodu" alanına göre kurumAdi alanını \*\*\* şeklinde maskeliyor. Headerdaki kurumKodu ile response üzerindeki kurumKodu aynı ise kurum adı maskelenmeden gösteriliyor. Ayrıca request headerdaki kurumKodu "9999" ise yine maskelenmeden bu alan açık bir şekilde gösteriliyor. Headerdaki kurumKodu alanı 4 haneli rakamdan oluşup string olarak tutulmaktadır. Bana bu bilgiler doğrultusunda detaylı test senaryoları üretir misin? Olası tüm durumları kapsasın. Request, response ve kısa açıklamalarla cevabını bekliyorum.

### **ChatGPT**

**✅ Pozitif Test Senaryoları**

**🧪 Test 1 - Header kurumKodu, response kurumKodu ile aynı → kurumAdi açık gösterilmeli**

**Açıklama**: Header'daki kurumKodu ile response içindeki bireyselKrediler'deki kurumKodu eşleştiğinde kurumAdi maskelenmemelidir.

**Request**

**Header:**

kurumKodu: "1234"

**Body:**

{

"kimlikNo": "12345678901"

}

**Response**

{

"bireyselKrediler": [

{

"kurumKodu": "1234",

"kurumAdi": "X Bankası",

"tutar": 10000

}

]

}

**🧪 Test 2 - Header kurumKodu farklı, response kurumKodu farklı → kurumAdi maskelenmeli**

**Açıklama**: Header’daki kurum kodu ile eşleşmeyen diğer kurumların adları maskelenmelidir.

**Request**

**Header:**

kurumKodu: "1111"

**Body:**

{

"kimlikNo": "12345678901"

}

**Response**

{

"bireyselKrediler": [

{

"kurumKodu": "1234",

"kurumAdi": "\*\*\*",

"tutar": 10000

}

]

}

**🧪 Test 3 - Header kurumKodu: "9999" (özel durum) → tüm kurumAdı açık gösterilmeli**

**Açıklama**: "9999" özel kurum kodudur, tüm kurumAdi alanları açık görünmelidir.

**Request**

**Header:**

kurumKodu: "9999"

**Body:**

{

"kimlikNo": "12345678901"

}

**Response**

{

"bireyselKrediler": [

{

"kurumKodu": "1234",

"kurumAdi": "X Bankası",

"tutar": 10000

},

{

"kurumKodu": "5678",

"kurumAdi": "Y Finans",

"tutar": 5000

}

]

}

**🧪 Test 4 - Birden fazla kredi kaydı, bazıları eşleşiyor → sadece eşleşenler açık gösterilmeli**

**Açıklama**: Sadece header’daki kurum kodu ile eşleşenlerin kurumAdı alanı açık kalmalı, diğerleri maskelenmeli.

**Request**

**Header:**

kurumKodu: "1234"

**Body:**

{

"kimlikNo": "12345678901"

}

**Response**

{

"bireyselKrediler": [

{

"kurumKodu": "1234",

"kurumAdi": "X Bankası",

"tutar": 10000

},

{

"kurumKodu": "5678",

"kurumAdi": "\*\*\*",

"tutar": 2000

}

]

}

**🧪 Test 5 - bireyselKrediler listesi boş → boş response**

**Açıklama**: Kredi kaydı yoksa liste boş dönmeli, hata verilmemeli.

**Request**

**Header:**

kurumKodu: "1234"

**Body:**

{

"kimlikNo": "12345678901"

}

**Response**

{

"bireyselKrediler": []

}

**❌ Negatif Test Senaryoları**

**🧪 Test 6 - Header kurumKodu eksik → 400 Bad Request**

**Açıklama**: Header’da kurumKodu gönderilmemişse servis validasyon hatası döner.

**Request**

**Header:**

(kurumKodu gönderilmez)

**Body:**

{

"kimlikNo": "12345678901"

}

**Response**

{

"hataKodu": "400",

"hataMesaji": "kurumKodu header bilgisi zorunludur."

}

**🧪 Test 7 - Header kurumKodu 4 haneli değil → 400 Bad Request**

**Açıklama**: kurumKodu 4 haneli rakamsal string olmalıdır, aksi durumda validasyon hatası döner.

**Request**

**Header:**

kurumKodu: "12"

**Body:**

{

"kimlikNo": "12345678901"

}

**Response**

{

"hataKodu": "400",

"hataMesaji": "kurumKodu 4 haneli string formatında olmalıdır."

}

**🧪 Test 8 - kimlikNo eksik veya boş → 400 Bad Request**

**Açıklama**: Request body’de kimlikNo zorunlu bir alandır.

**Request**

**Header:**

kurumKodu: "1234"

**Body:**

{

"kimlikNo": ""

}

**Response**

{

"hataKodu": "400",

"hataMesaji": "kimlikNo alanı zorunludur."

}

**🧪 Test 9 - kimlikNo formatı hatalı (alfabetik karakter içeriyor) → 400 Bad Request**

**Açıklama**: kimlikNo 11 haneli sayısal bir değer olmalıdır.

**Request**

**Header:**

kurumKodu: "1234"

**Body:**

{

"kimlikNo": "ABCDEF12345"

}

**Response**

{

"hataKodu": "400",

"hataMesaji": "kimlikNo 11 haneli sayısal bir değer olmalıdır."

}