OnlineAsistan

Entegrasyon rehberi kendi yazılımınızla

Hangi yazılımı kullanıyor olursanız olun, asistanı kendi verinizle konuşturabilirsiniz. Uçları siz açarsınız, biz çağırırız; veri sizde kalır. Bu rehber sözleşmeyi, örnek kodu ve sahada öğrendiğimiz tuzakları anlatır.

Altı bağımsız model

Hepsini birden kurmanız gerekmez; her biri tek başına çalışır. Çoğu firma **widget + sağlayıcı API** ile başlar, temsilcisi olan firma devri de ekler.

01 · VERİ

Sağlayıcı API

Asistan sizin verinizle konuşur: müşteri bulur, kimlik doğrular, borç söyler, talep açar. Uçları siz açarsınız, sözleşme sabittir.

Kimin için: kendi yazılımı olan her firma

02 · SOHBET

Web sitesi widget'ı

Sitenize tek satır kod. Ziyaretçi asistanla konuşur, gerekirse temsilciye bağlanır. Alan adı kısıtı panelden verilir.

Kimin için: sitesi olan herkes

03 · DEVİR

Webhook ile aktarma

Görüşme canlı temsilciye düştüğünde kendi yazılımınıza gider; temsilci oradan yazar. İstemezseniz bizim panelimizi kullanırsınız.

Kimin için: kendi çağrı/destek ekranı olan firma

04 · TELEFON

Santral (SIP)

Asistan santralinize bir dahili gibi kaydolur. Gelen çağrı kurallarından o dahiliye yönlendirme yaparsınız.

Kimin için: telefonla aranan firma

05 · MESAJ

WhatsApp · Instagram · Facebook

Kendi numaranız ve kendi sayfanız. Gelen mesaj ve yorumlar asistana düşer, cevaplar aynı kanaldan gider.

Kimin için: sosyal medyadan mesaj alan firma

06 · KÜNYE

Künye API

Firma künyesini (çalışma saati, ürünler, karşılama) kendi sisteminizden güncellersiniz; panele girmenize gerek kalmaz.

Kimin için: çok şubeli / sık değişen işletme

Bu rehberin konusu birinci model

Diğerlerinin adım adım kurulumu `onlineasistan.net/entegrasyon` sayfasında ve `ornekler/` klasöründeki hazır kodlarda anlatılıyor.

Uçları siz açarsınız, biz çağırırız

Sizden bize veri göndermenizi istemiyoruz. Müşteri bir şey sorduğunda biz sorarız, siz o an cevaplarsınız. Veri sizde kalır; bizde kopyası durmaz.

```
Müşteri  ⟶  OnlineAsistan  ⟶  Sizin yazılımınız
"borcum ne kadar?"

1. POST /auth/token   → Bearer jeton (1 saat)
2. GET  /customers/find?find=5321112233
3. POST /agent/verify  → ad-soyad + TC son 4
4. GET  /customers/find?find=1001234  → borç + faturalar

Asistan: "İki ödenmemiş faturanız var, toplam 1.500,00 TL."
```

Adres ve kimlik

Yol öneki	`/api/onlineasistan/v1`
Kimlik	Basic `client_id:client_secret` → Bearer jeton
Protokol	Yalnız HTTPS, geçerli sertifika
Zaman aşımı	20 sn — hedefiniz **1 sn altı** olmalı (telefonda insan bekliyor)
İstek kaynağı	`45.83.35.125`

Zarf — her cevap aynı biçimde

```
// başarı
{ "data": { … }, "meta": { "version": "v1" }, "errors": [] }

// hata — HTTP kodu ne olursa olsun errors doluysa hata sayılır
{ "data": null, "meta": {},
  "errors": [{ "code": "CUSTOMER_NOT_FOUND", "detail": "Kayıt bulunamadı" }] }
```

> **Zarfsız cevap okunmaz**
>
> Doğrudan `{"records": [...]}` dönerseniz asistan cevabı çözemez ve müşteriye "sistemde göremiyorum" der. `code` alanı makine tarafından okunur: metnini değiştirin, **kodunu değiştirmeyin**.

Üç uç olmadan asistan çalışmaz

Bunlar açılmadan müşteri tanınmaz, kimlik doğrulanamaz ve davranış kurallarınız asistana geçmez. Açmadığınız diğer uçları **404** ile cevaplayın — asistan o yeteneği kapalı sayar ve müşteriye "yapabiliyorum" demez.

UÇ	NE İÇİN	
`POST /auth/token`	Bearer jeton	ZORUNLU
`GET /customers/find`	Müşteri arama (telefon, no, TC, ad)	ZORUNLU
`POST /agent/verify`	Kimlik doğrulama (KVKK kapısı)	ZORUNLU
`GET /agent/config`	Öğretiler, devir hedefi, marka	ZORUNLU

Müşteri kaydı — beklenen alanlar

```
{ "data": { "records": [{
    "abone_no": "1001234",
    "isim": "AYHAN", "soyisim": "SARI",      // maskelemeyin
    "durum": "Aktif", "tarife": "FIBER 100",
    "tarife_fiyat": "750,00", "toplam_borc": 1500,
    "tesis_adres_satir": "ÖZGÜR MAH. KAKTÜS CAD. NO 5 D 13",
    "telefon_1": "5321112233",
    "faturalar": [{ "fatura_no": "F-2026-001", "kalan": 750,
                    "durum": "Ödenmedi", "son_odeme_tarihi": "2026-08-31" }]
  }] }, "errors": [] }
```

Kimlik doğrulama cevabı

```
{ "data": { "result": {
    "dogrulandi": true, "abone_no": "1001234", "kalan_deneme": 3,
    "abonelikler": [ // birden çok kayıt varsa hepsi
      { "abone_no": "1001234", "adres": "ÖZGÜR MAH. …" } ]
  } }, "errors": [] }
```

> **Birden çok kaydı olan müşteri**
>
> `abonelikler` dolduğunda asistan numaralı liste sunar ("1) … 2) …") ve müşteriye yalnız numarayı söyletir. Liste vermezseniz asistan hangi hattın sorulduğunu bilemez.

Açtığınız her uç bir yeteneği açar

Sırayla açabilirsiniz; hiçbiri diğerinin ön koşulu değildir.

UÇ	AÇTIĞINDA ASİSTAN NE YAPAR
`POST /customers/connection-check`	**Hizmet durumu söyler.** `status` sabit sözlükten: `online`, `offline`, `inactive`, `expired`, `radacct_only`, `inconsistent`, `mikrotik_unreachable`
`POST /tickets`	**Talep / arıza kaydı açar.** Kayıt numarası dönmezse "açıldı" demez. Açık kayıt varsa `existing: true` dönün, yenisi açılmasın
`POST /agent/message`	**Ödeme linki / şifre gönderir.** Mesajı siz gönderirsiniz; numara **sizde kayıtlı** numaradır
`POST /agent/altyapi`	**Yeni müşteriye kapsama sorgular.** `adres_kesin` false ise asistan adrese özel söz vermez
`GET /agent/genel-ariza`	**Toplu kesintiyi anlatır.** `abone_no` verildiğinde *o müşteri* etkileniyor mu, onu dönün
`POST /agent/calls`	**Görüşme özetini size yazar** — müşteri kartınızda görünsün diye

Asistanın davranışını siz yönetirsiniz

`/agent/config` içindeki `ogretiler` alanı en kıymetli yerdir: kendi panelinizden yazdığınız kurallar web, WhatsApp ve telefonda — üç kanalda da — dakikalar içinde geçerli olur.

```
{ "ogretiler": [
    { "baslik": "İptal talepleri",
      "icerik": "Kendin işleme alma, temsilciye aktar." }
  ],
  "aktarma": { "temsilci_var": true, "temsilci_dahililer": ["100", "101"],
               "temsilci_mesai": "09:00-18:00" },
  "kimlik_dogrulamasiz_yasak_aksiyonlar": ["borc_sorgula", "ariza_kaydi_ac"] }
```

Çalışan örnek kod var

`ornekler/4-saglayici-api/ornek-saglayici.php` — tek dosya, bütün uçları örnek veriyle cevaplıyor. `php -S 127.0.0.1:8080` ile çalıştırıp Postman'i ona doğrultun, sonra içini kendi veritabanınıza bağlayın.

Bunların hepsi canlıda yaşandı

Aşağıdaki maddelerin her biri gerçek bir görüşmede müşteriye yanlış bilgi verilmesine yol açtı. Sözleşmenin bu kadar ayrıntılı olmasının sebebi budur.

Para alanını bazen çıplak bazen biçimli vermek

Aynı kayıtta `toplam_borc: 33800` ve `tarife_fiyat: "19.500,00"` vardı. Model çıplak sayıyı kuruş sandı: **33.800 TL borç müşteriye "338,00 TL" diye söylendi.** Müşteri o tutara göre ödeme linkine girdi. Bir alanı hangi biçimde veriyorsanız hepsini aynı biçimde verin.

Mahalle adını birebir eşleştirmek

Kayıtta `ÖZGÜR` yazıyordu, müşteri "Özgür Mahallesi" dedi. Sorgu bulunamadı ve **hizmet verilen adrese "bölgemizde değil" dendi**; müşteri sinirlenip kapattı. Tür ekini atın (*mahallesi* / *mah.* / *mh*), Türkçe küçültme uygulayın (I→ı, İ→i).

Ad-soyadı maskelemek

Kimlik doğrulanmadan önce ad `"Riza*****"` geliyordu; asistan çağrıyı **"Sayın Riza yıldız yıldız yıldız"** diye karşıladı. Müşteri numarasını ve adresi maskeleyin — **ad-soyadı maskelemek yalnız karşılamayı bozar**, KVKK'ya katkısı olmaz.

Fatura satırlarını `fatura_no` olmadan vermek

Tek fatura kalemlere bölünüp birden çok satır olarak geliyordu ve her satırda faturanın toplamı yazıyordu: müşteriye **"4 ödenmemiş faturanız var"** dendi, gerçekte 2 fatura vardı. Numara verirseniz biz tekilleştiririz.

Kayıt açılmadığı hâlde başarı dönmek

Kayıt numarası dönmediğinde asistan "kayıt açıldı" **demez** — bu kasıtlıdır. Açık kayıt varken yenisini açmayın: `existing: true` ve mevcut numarayı dönün.

Postman ile başlayın

Koleksiyonda her istek için sözleşme sınaması var: zarf, alan adları, maskesizlik, mükerrer kayıt koruması. Hepsi yeşilse uyumlusunuz.

1

Koleksiyonu ve ortamı içe aktarın

- `OnlineAsistan-Saglayici-API.postman_collection.json`
- `OnlineAsistan-Saglayici-API.postman_environment.json`
- Ortamda `base_url`, `client_id`, `client_secret` ve gerçek bir test müşterisinin bilgilerini doldurun

2

Klasörleri sırayla çalıştırın

1 · Kimlik → jeton alınır ve ortama yazılır. **2 · Zorunlu uçlar** → müşteri arama, kimlik doğrulama (yanlış ve doğru bilgiyle), yapılandırma. **3 · İsteğe bağlı** → açtığınız uçlar. Runner ile toplu da çalışır.

3

Panelden bağlayın

OnlineAsistan panelinde **Ayarlar → API Bağlantısı** → *Standart API (kendi yazılımınız)*. Adres, `client_id` ve `client_secret` girilir. Yol öneki sabittir, ayrıca yazmazsınız.

4

Bir görüşme yapın

Widget'tan “borcum ne kadar” diye sorun. Asistan kimlik ister, doğrular ve gerçek tutarı söyler. Aynı akış telefonda ve WhatsApp'ta da geçerlidir — tek bağlantı üç kanalı birden açar.

Güvenlik

Taşıma	Yalnız HTTPS; self-signed sertifika reddedilir
Sır	`client_secret` bizde şifreli saklanır, panelde bir daha görünmez
IP kısıtı	İsterseniz yalnız `45.83.35.125` adresine izin verin
İptal	Jetonu geçersiz kılmanız yeterli; bağlantı anında kesilir
Kapsam	Kişisel veriyi yalnız asistan sorduğu anda dönün; toplu aktarım istemiyoruz

Yardım

bilgi@onlineasistan.net · `onlineasistan.net/entegrasyon` · Sözleşmenin tam metni: `ornekler/SAGLAYICI-API.md`