

# PAYTR KART SAKLAMA ENTEGRASYONU (CAPI)

## KULLANICININ KARTINI KAYDETME – ÖDEME SIRASINDA (CAPI PAYMENT)

Bu servisi kullanarak ödeme esnasında PAYTR’da kayıtlı bir kullanıcı ve kullanıcıya ait bir kart oluşturabilirsiniz. Bunun için izlenecek süreç aşağıdaki şekilde olmalıdır.

- 1- Direkt API Dokümanında belirtildiği şekilde ödeme sayfanızı oluşturun.
- 2- Kredi kartı bilgilerinin girildiği adımda kullanıcıya kartını kaydetmek istediğini seçebileceği bir onay kutucuğu ekleyin.
- 3- Kullanıcı kart bilgilerini kaydetmeyi seçerse gerekli bilgileri POST içeriğine ekleyin:
  - a. Kullanıcı adına sistemde ilk kez bir kart kaydediliyorsa POST içeriğinde yalnızca “store\_card” parametresi gönderilir.
  - b. Kullanıcının daha önceden sistemde tanımlanmış bir kartı varsa ve yeni bir kart kaydetmek istiyorsa, POST içeriğinde “utoken” ve “store\_card” parametreleri birlikte gönderilmelidir.
- 4- Ödemenin sonucunda gelen bildirimde (Bildirim URL’e), kart saklama için gönderilen aşağıdaki değerleri ilgili tablolarınıza kaydedin ve bir sonraki işlem için hazır durumda tutun.

### Ödeme işlemi sonucunda ödemeye ilgili bilgilere ek olarak dönen kart saklama bilgileri

Açıklama	Alan adı / tipi
User Token: Sitenizdeki kullanıcıya özel PAYTR tarafından oluşturulmuş token. Bu token’ı sisteminizde işlemi yapan kullanıcıyla eşleştirmelisiniz.	utoken (string)

## KULLANICININ KAYITLI KART LİSTESİNİ ALMA (CAPI LIST)

- 1) Bir kullanıcı ödeme işlemine başlarken kullanıcıya PAYTR’da kayıtlı olan kartları listelemek için <https://www.paytr.com/odeme/capi/list> adresine aşağıdaki parametreler ile istek yapın.

Zorunlu	Açıklama	Alan adı / tipi
✓	Mağaza No: PAYTR tarafından size verilen Mağaza numarası	merchant_id (integer)
✓	User Token: Kart kayıt sonrası ödeme bildiriminde tarafınıza PAYTR sisteminden bildirilen kullanıcıya özel token	utoken (string)
✓	Paytr Token: İsteğin sizden geldiğine ve içeriğin değişmediğine emin olmamız için oluşturacağınız değerdir (Hesaplama ile ilgili olarak örnek kodlara bakmalısınız)	paytr_token (string)

- 2) Yapılan isteğe aşağıdaki tabloda bulunan değerler JSON formatında dönecektir. Gönderdiğiniz bilgiler ile herhangi bir eşleşme bulunamadığında cevap boş JSON olarak döner.

Açıklama	Alan adı / tipi	Olası / Örnek Değerler
Status: Hata durumunda <b>error</b> olarak döner, işlem başarılı olduğunda döndürülmez	status (string)	error
Error Message: İstek başarısız olduğu durumlarda err_msg’de hata nedeni döndürülür	err_msg (string)	Örnek: Bağlantı hatası oluştu
Card Token: Kullanıcının kayıtlı kartını tanımlayan token	ctoken (string)	
Son 4: Kayıtlı kartın son 4 hanesi	last_4 (string)	
CVV gerekli: Bu kayıtlı kart ile ödeme yapmak için CVV gerekip gerekmediği	require_cvv (string)	0 veya 1 (Bir kart için 1 dönerse, kart ile ödeme yapılabilmesi için kullanıcıdan CVV bilgisini almanız gerekmektedir)

Ay: Kartının son kullanma tarihinin ay bilgisi	month (string)	Örnek: 05
Yılı: Kartının son kullanma tarihinin yıl bilgisi	year (string)	Örnek: 28
Banka: Kartının bankası Karta ait BIN kayıtlı değil veya yurt dışı kart ait ise c_bank değeri <b>boş</b> olarak dönecektir. Bu sayede yurtdışı kartlar ayırt edilerek işlem sağlanabilir.	c_bank (string)	Kayıtlı bir BIN olması halinde Örnek: c_bank => Finansbank şeklinde dönüş olacaktır. Kayıtlı olmayan BIN veya yurtdışı kart olması durumunda Örnek: c_bank => (boş şekilde dönecektir)
Adı Soyadı: Kullanıcının kart kayıt sırasında girdiği ad soyadı	c_name	
Kart Program Ortaklığı İsmi	c_brand (string)	Örnek: maximum, bonus, world vb.
Kart Tipi: Kredi kartı veya banka kartı / ön ödemeli kart	c_type (string)	credit veya debit
Şirket Kartı: Kartın şirket kartı olup olmadığı bilgisi	businessCard (string)	y / n
Kartın şeması. Kartın hangi şemaya ait olduğu bilinmiyorsa OTHER döner.	schema (string)	VISA, MASTERCARD, AMEX, TROY, vb.

- 3) Dönen kart bilgileri alınarak kullanıcıya seçebileceği kayıtlı kartlarını listeleyin.
- 4) Seçilen kayıtlı kartın ctoken bilgisi ve kullanıcıya ait utoken bilgisini kullanarak ödeme başlatın (Eğer seçilen kart için require\_cvv değeri 1 ise kullanıcıya CVV gireceği bir alan sunmalısınız ve ödeme isteğinde CVV'yi göndermelisiniz).

## KULLANICI KARTINI SİLME (CAPI DELETE)

- 1- Bir kullanıcının kayıtlı kartları arasından bir kart silmek için <https://www.paytr.com/odeme/capi/delete> adresine aşağıdaki parametreleri göndererek istek yapın.

Zorunlu	Açıklama	Alan adı / tipi
✓	Mağaza No: PAYTR tarafından size verilen Mağaza numarası	merchant_id (integer)
✓	Paytr Token: İsteğin sizden geldiğine ve içeriğin değişmediğine emin olmamız için oluşturacağınız değerdir (Hesaplama ile ilgili olarak örnek kodlara bakmalısınız)	paytr_token (string)
✓	User Token: Kart kayıt sonrası ödeme bildiriminde tarafınıza PAYTR sisteminden bildirilen kullanıcıya özel token	utoken (string)
✓	Card Token: Kullanıcının kayıtlı kartını tanımlayan token.	ctoken (string)

- 2- Yapılan isteğe aşağıdaki tabloda bulunan değerler JSON formatında dönecektir. Dönen cevaba göre kullanıcıyı bilgilendirebilirsiniz.

Açıklama	Alan adı / tipi	Olası / Örnek Değerler
Status: Yapılan kart silme isteğinin başarılı ya da başarısız olduğunu belirtir	status (string)	success veya error
Error Message: İstek başarısız olduğu durumlarda err_msg'de hata nedeni döndürülür	err_msg (string)	Örnek: Kart yok veya daha önce silinmiş

## KAYITLI KART İLE TEKRARLAYAN ÖDEME ALMA (RECURRING PAYMENT)

Bu servisi kullanarak tekrarlayan ödeme için PAYTR’da kullanıcınıza ait kayıtlı bir kart ile dilediğiniz zaman veya aralıklarla ödeme alabilirsiniz.

- 1- Recurring Payment adımı belirlenen değerlerle birlikte ödeme istek bloğunu oluşturun. Ödeme işlemi, kendi oluşturacağınız yapı üzerinden, kayıtlı kart bilgileri ile servise göndereceğiniz istek sonucunda oluşacaktır. Bu sebepten dolayı kullanıcıyla etkileşime girecek form oluşturulmasına gerek bulunmamaktadır.
- 2- İşlemler Non3D (Non Secure) olarak gerçekleşecektir. Kullanıcınız herhangi bir ek işlem yapmayacak veya işlem sırasında kendisinden herhangi bir bilgi talep edilmeyecektir (Kullanabilmek için mağazanızda Non3D yetkilerinin açık olması gerekmektedir).
- 3- CAPI LIST servisinde, adına ödeme gerçekleştirilmek istenen kullanıcınıza ait utoken verisini kullanarak, ctoken verisine ulaşmanız gerekmektedir. Akabinde utoken, ctoken ve aşağıda ki tabloda belirtilen değerlerle birlikte <https://www.paytr.com/odeme> adresine **POST** metodu ile ödeme isteğinde bulunabilirsiniz.

Zorunlu	Açıklama	Alan adı / tipi	Kısıtlar
✓	Mağaza No: PayTR tarafından size verilen Mağaza numarası	merchant_id (integer)	
✓	PayTR Token: İsteğin sizden geldiğine ve içeriğin değişmediğine emin olmanız için oluşturacağınız değerdir	paytr_token (string)	<b>Hesaplama ile ilgili olarak örnek kodlara bakmalısınız</b>
✓	Müşteri ip: İstek anında aldığınız müşteri ip numarası (Önemli: Lokal makinenizde yapacağınız denemelerde mutlaka dış IP adresini gönderdiğinizden emin olun)	user_ip (string)	En fazla 39 karakter (ipv4)
✓	Mağaza sipariş no: Satış işlemi için belirlediğiniz <b>benzersiz</b> sipariş numarası. (Not: Sipariş no ödeme sonuç bildirimi esnasında geri dönen değerler arasındadır)	merchant_oid (string)	En fazla 64 karakter, <b>Alfa numerik</b>
✓	Müşteri eposta adresi: Müşterinin sisteminizde kayıtlı olan veya form aracılığıyla aldığınız eposta adresi	email (string)	En fazla 100 karakter
✓	Ödeme tipi	payment_type(string)	('card', 'card_points')
✓	Ödeme tutarı: Siparişe ait toplam ödeme tutarı	payment_amount (double), ondalık olarak nokta (.) ve noktadan sonra iki hane	Örn: 100.99 veya 150 veya 1500.35
✓	Taksit sayısı	installment_count(int)	0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
	Kart tipi (Taksitli işlemlerde kullanmak üzere)	card_type(string)	advantage, axess, combo, bonus, cardfinans, maximum, paraf, world, saglamkart
	Para birimi	currency(string)	<b>TL(veya TRY), EUR, USD (Boş ise TL kabul edilir)</b>
	Ödeme sürecinde kullanılacak dil	client_lang(string)	Türkçe için <b>tr</b> veya İngilizce için <b>en</b>

			(Boş gönderilirse tr geçerli olur)
	Mağaza canlı modda iken test işlem yapmak için 1 olarak gönderilebilir	test_mode	0 veya 1
	Kayıtlı kart ile ödeme alabilmek için Non 3D 1 gönderilebilir	non_3d	1
	Non 3D işlemde, başarısız işlem durumunu test etmek için 1 gönderilir (test_mode ve non_3d değerleri 1 ise dikkate alınır!)	non3d_test_failed	0
✓	Müşterinin başarılı ödeme sonrası yönlendirileceği sayfa (Örn. Siparişlerim takip sayfası) (Uyarı: Müşteri bu sayfaya ulaştığında henüz sipariş onaylanmış olmaz)	merchant_ok_url(string)	En fazla 400 karakter Uyarı: Tam URL olmalıdır
✓	Müşterinin ödemesi sırasında beklenmeyen bir hatada yönlendirileceği sayfa	merchant_fail_url(string)	En fazla 400 karakter Uyarı: Tam URL olmalıdır
✓	Müşteri adı ve soyadı: Müşterinin sisteminizde kayıtlı olan veya form aracılığıyla aldığınız adı ve soyadı	user_name (string)	En fazla 60 karakter
✓	Müşteri adresi: Müşterinin sipariş sırasında ilettiği adresi	user_address (string)	En fazla 400 karakter
✓	Müşteri telefon numarası: Müşterinin sipariş sırasında ilettiği telefon numarası	user_phone (string)	En fazla 20 karakter
✓	Sepet içeriği: Müşterinin siparişindeki ürün/hizmet bilgilerini içermelidir	user_basket (string)	JSON tipinde (Örnek kodları inceleyin)
	Hata döndür: PayTR'a yanlış veya eksik bilgi iletilmesi durumunda sistemden hata mesajı döndürülmesi için 1 gönderilmelidir	debug_on (int)	0 veya 1 <b>(Entegrasyon ve test sürecinde hataları tespit etmek için mutlaka 1 gönderin)</b>
✓	User Token: Sitenizdeki kullanıcıya özel PAYTR tarafından oluşturulmuş token. Bu token'ı sisteminizde işlemi yapan kullanıcıyla eşleştirmelisiniz.	utoken(string)	
✓	Card Token: Kullanıcının kayıtlı kartını tanımlayan token	ctoken(string)	
	Recurring: Ödeme isteğinin gönderilmesi ardından işlem sonucuna göre başarılı veya başarısız sayfasına yönlendirme yapılmadan, JSON formatında olan yanıt direkt olarak istek sonucuna döner. Ek olarak; işleme ait detaylar tanımlı olan Bildirim URL adresine gönderilir. Recurring sonucunda dönen status alanının alabileceği değerler "failed", "wait_callback" ve "success" şeklindedir.  Not: Bu işlem için mağazanızda Non3D yetkisinin açık olması gerekmektedir.	recurring(int)	0 veya 1 (İlgili yetkinin mağazaya tanımlanabilmesi için tarafımıza talep iletilmesi gerekmektedir. Birimlerimizin onayından geçmesi halinden yetki mağazaya tanımlanacaktır.)

RECURRING YANITLARI		
status	msg (açıklama)	try_again (tekrar dene)
failed	<i>"Kart banka tarafından kapatılmış. Bu kart ile tekrar işlem göndermeyin."</i> veya farklı bir hata mesajı.	false
failed	Henüz devam eden bir işleminiz bulunmaktadır, sonuçlandıktan sonra tekrar deneyebilirsiniz.	true
wait_callback	Ödeme Kontrol Ediliyor, Bildirimi Bekleyin.	-
success	Ödeme Başarılı.	-