

KKTC E-Fatura REST API Kullanım Kılavuzu

Sürüm: a595aabe

Bu kılavuz, Kuzey Kıbrıs Türk Cumhuriyeti (KKTC) E-Fatura sistemine ait yeni REST API'nin kullanım detaylarını açıklamak amacıyla hazırlanmıştır.

NOT: İstek gövdesinin en güncel ve eksiksiz JSON şemasına ulaşmak için dökümanın sonundaki ****Swagger UI**** adresini kullanınız.

Veri yapısında oluşabilecek olası değişiklikler için en doğru ve güncel kaynak Swagger dokümantasyonudur.

1. GİRİŞ

REST API, mükelleflerin fatura oluşturma, listeleme ve iptal işlemlerini HTTP protokolü üzerinden gerçekleştirmelerini sağlar.

Bu belge, geliştiricilerin bu servisleri doğru ve güvenli bir şekilde kullanmalarını sağlamak için hazırlanmıştır.

2. KULLANIM KOŞULLARI

- API servisleri yalnızca yetkili kullanıcılar tarafından kullanılabilir.
- Her istek, uygun yetkilendirme başlığı (Bearer Token) ile gönderilmelidir.
- Servis adresi, KKTC Maliye Bakanlığı tarafından sağlanır.
- JSON formatı kullanılmalıdır ve tüm yanıtlar da JSON döner.

3. ANA URL

<https://efatura.maliye.gov.ct.tr/api>

3.1 TEST URL

<https://test-efatura.maliye.gov.ct.tr/api>

3.2 GENEL AÇIKLAMALAR VE İŞ PRENSİPLERİ

Bu bölüm, KKTC E-Fatura sisteminin temel iş prensipleri ve API kullanımına yönelik önemli yaklaşımları açıklamaktadır.

Fatura Veri Yapısı ve UBL Uyumu

KKTC E-Fatura sistemi, uluslararası standartlara uyum ve gelecekteki esneklik için **UBL (Universal Business Language)** yapısına uygun olarak tasarlanmıştır.

Mevcut yerel gereksinimler UBL'nin tüm özelliklerini zorunlu kılmaya da, bu yaklaşım sayesinde API, olası yeni düzenlemelere ve farklı kullanım senaryolarına kolayca adapte olabilecek esnekliği korumaktadır.

Faturalarda Tedarikçi Bilgilerinin Tam Olarak Gönderilmesi (Değişmezlik Prensipleri)

Faturalar, hukuki olarak değiştirilemez belgelerdir. Bu nedenle:

- Bir fatura oluşturulurken **tedarikçi bilgileri** (unvan, adres, VKN/KKTCKN vb.) sisteme tam olarak, ID referansı olmaksızın gönderilmelidir.
- Bu prensip, tedarikçi bilgilerinde gelecekte bir güncelleme olması durumunda dahi, geçmiş faturaların hukuki geçerliliğini ve doğruluğunu korumasını sağlar.
- Aynı tedarikçiye ait olsa bile, bilgiler her fatura kaydında yeniden gönderilmelidir; tedarikçiyi yalnızca bir ID üzerinden referanslamak geçmiş fatura verisinin bozulmasına neden olabilir.

Kullanıcı Yetkilendirme ve Çoklu Mükellef Yönetimi

- **Kullanıcı Kaydı:** Fatura oluşturma yetkisine sahip kişi (fatura oluşturan taraf), sisteme kullanıcı olarak kayıt olmalı ve kullanıcı adı ile parolasını Vergi Dairesi'nden temin etmelidir. Kullanıcı adı ve parola, teknik olarak kişilere tahsis edilir, şirketlere değil.
- **Fatura Kesme Yetkisi:** Fatura kesme işlemi, oturum açan yetkili kullanıcı tarafından, yetkili olduğu mükellef (şirket) adına yapılır.
- **Çoklu Mükellef Yönetimi:** Bir kişi, aynı anda birden fazla şirketin yöneticisi veya yetkilisi olabilir ve bu şirketler adına fatura düzenleyebilir. Oturum açmış kullanıcının yetkili olduğu mükellefleri (şirketleri) görüntülemek için `/api/mukellefler` servisi kullanılmalıdır. Bu servis, kullanıcının hangi şirketler üzerinde yetkisi olduğunu döndürür.

Şube Tanımlaması ve Kullanımı

- Bir mükellefin birden fazla ticari şubesi bulunabilir.
- Sistemde, bu şubelerin birbirinden ayırt edilebilmesi amacıyla fatura oluşturma ve şube yönetimi işlemlerinde **şube bilgisi** alanı (`subeKod`) kullanılır.
- Şube tanımlama işlemi için ayrıca bir izin alınması gerekmez; yetkili kullanıcı tarafından doğrudan API üzerinden gerçekleştirilebilir.

Pul Ücreti ve Bakiye Yönetimi

- Oluşturulan her bir fatura için sistem tarafından otomatik olarak bir **pul ücreti** alınır.
- Bu ücret, faturayı düzenleyen mükellefin sistemdeki **pul bakiyesinden** otomatik olarak düşülür.

- Fatura oluřturma iřleminin bařarılı olabilmesi için mükellefin yeterli pul bakiyesi bulunmalıdır.
- Bakiye yetersizliđi durumunda fatura oluřturma iřlemi hata ile sonuçlanır.

3.3 Fatura Numarası Formatı

Fatura numaraları sistem genelinde belirli bir formatta ve benzersiz olmalıdır.

Fatura numarası oluřturulurken ařađıdaki yapı takip edilmelidir:

Format: VKN-YIL-SUBE-SIRA_NO

Örnek: 123456789-2025-KASA1-00000000001

Bileřenler:

- **VKN:** Őirketin Vergi Kimlik Numarası.
- **YIL:** Faturanın düzenlendiđi 4 haneli yıl bilgisi.
- **SUBE:** Őirket tarafından belirlenen, fatura serisini ayıran Őube kodu (Örn: OTEL, KASA, SUBE1, MERKEZ vb.).
- **SIRA_NO:** 11 haneli, her yıl bařında sıfırlanan artan sıra numarası.

İř Kuralları:

- Her birim (seri) kendi içinde bađımsız bir artan sıra numarası takip edebilir.
- Yıl deđiřtiđinde (her takvim yılı bařında), tüm birimlere ait sıra numaraları tekrar 1'den bařlar.

3.4 Fatura Doğrulama Kodu (UUID)

E-fatura sisteminde her fatura için benzersiz bir doğrulama kodu (UUID) oluřturulması gerekmektedir.

Bu kodun **UUIDv7** formatında olması zorunludur.

Eski nesil UUID'lerin aksine v7, iđerisinde zaman bilgisi barındırır.

UUIDv7 oluřturulurken sistem saati baz alınır.

Bu nedenle sunucu saatinin **güncel** ve **senkronize** olduđundan emin olunmalıdır.

3.5 Tarih Formatı

Sistemdeki tüm tarih ve zaman bilgileri **ISO 8601** formatında ve **zaman dilimi (timezone)** bilgisini iđerecek řekilde olmalıdır.

Formatlar:

- **Tarih ve Zaman:** YYYY-MM-DDTHH:mm:ss±HH:mm (Örn: 2025-01-20T14:30:00+03:00)

4. MÜKELLEFL İŐLEMLERİ

4.1 Fatura Oluřturma

POST /api/mukellefler/{VKN}/faturalar

Belirli bir mükellef adına bir veya birden fazla fatura oluşturmak için kullanılır.

- **Path Parametresi:** `{VKN}` - Fatura kesilecek mükellefin 9 haneli vergi kimlik numarası.

Özel Matrah Kapsamındaki Vergiler İçin Not

Özel matrah kapsamındaki vergiler için fatura gönderimi sırasında `vergiler` alanına aşağıdaki şekilde bilgi eklenmelidir:

```
{
  "vergiKodu": "0002",
  "vergiOrani": 0,
  "vergiTutari": 0,
  "vergiMuafiyetKodu": "101",
  "vergiMuafiyetAciklamasi": "Özel Matrah"
}
```

Bu durumda `vergiOrani` ve `vergiTutari` sıfır olmalı, `vergiMuafiyetKodu` "101" ve `vergiMuafiyetAciklamasi` "Özel Matrah" olarak belirtilmelidir.

Ülke kodu alanı ISO 3166-1 alpha-3 (kısaca **ISO3**) olmalıdır. örnek

```
Türkiye: TUR
Almanya: DEU
Amerika Birleşik Devletleri: USA
Kuzey Kıbrıs Türk Cumhuriyeti: XCT
```

4.2 Fatura Listeleme

GET /api/mukellefler/{VKN}/faturalar

Faturaları filtreleme, sıralama ve sayfalama ile listeler.

- **Path Parametresi:** `{VKN}` - Fatura listelenecek mükellefin 9 haneli vergi kimlik numarası.

5. ŞUBE İŞLEMLERİ

- POST /api/mukellefler/{VKN}/subeler → Yeni şube oluşturma

- PUT /api/mukellefler/{VKN}/subeler/{subeld} → Şube bilgisi güncelleme

- DELETE /api/mukellefler/{VKN}/subeler/{subeld} → Şube silme (soft delete)
- GET /api/mukellefler/{VKN}/subeler/{subeld} → Şube detay bilgisi
- GET /api/mukellefler/{VKN}/subeler → Mükellef'e ait tüm şubeleri listeleme

6. Fatura Türleri

- GET /api/fatura-turleri

7. Para Birimleri

- GET /api/para-birimleri

Sistemde tanımlı geçerli para birimlerini listeler.

8. Vergiler

- GET /api/vergiler

Sistemde tanımlı vergi türlerini (ad ve vergi kodu) listeler.

9. Birim Türleri

- GET /api/birim-turleri

Faturalarda kullanılacak miktar birimlerini (Adet, KG, Metre vb.) ve bunlara ait kodları listelemek için kullanılır.

10. Giriş işlemi

KKTC E-Fatura API'si için Keycloak tabanlı giriş işlemi aşağıdaki gibidir:

Tüm API istekleriniz için bir "Bearer Token" kullanmanız gerekmektedir. Bu token'ı almak için bir POST isteği göndermeniz gerekmektedir.

HTTP Metodu: POST Content-Type Başlığı: İstek başlıklarınızda Content-Type: application/x-www-form-urlencoded bilgisini göndermelisiniz.

İstek URL'leri:

Ana URL: <https://keycloak.maliye.gov.ct.tr/realms/vergi/protocol/openid-connect/token>

Test URL: <https://keycloak.maliye.gov.ct.tr/realms/vergi-stage/protocol/openid-connect/token>

Gövde kısmını `x-www-form-urlencoded` formatında, aşağıdaki anahtar-değer çiftlerini kullanarak doldurmalısınız:

Anahtar	Değer
<code>grant_type</code>	password
<code>client_id</code>	efatura-frontend
<code>username</code>	Sisteme kayıtlı kullanıcı adınız.
<code>password</code>	Kullanıcı adınıza ait parolanız.

Yanıt ve `access_token` Elde Edilmesi:

Bu isteğin başarılı bir şekilde gönderilmesi sonucunda, bir JSON objesi alacaksınız. Bu objenin içinde yer alan `access_token` alanı, API'nin diğer uç noktalarına yapılacak isteklerde kullanılacak olan **Bearer Token**'dir.

Sonraki İsteklerde Token Kullanımı:

API'nin diğer uç noktalarına yapacağınız her istekte, elde ettiğiniz bu `access_token` 'ı **Authorization** başlığı altında aşağıdaki formatta göndermeniz gerekmektedir:

```
Bearer ACCESS_TOKEN
```

Örneğin:

```
Authorization: Bearer <Giriş işleminden aldığınız access_token buraya gelecek>
```

11. YANIT KODLARI

Kod	Açıklama	Geliştirici Notu
400 Bad Request	Hatalı istek.	Gövde yapısını veya veri tiplerini kontrol edin.
401 Unauthorized	Yetkisiz erişim.	Gönderilen Bearer Token 'in geçerliliğini ve süresini kontrol edin.
422 Unprocessable Entity	Doğrulama Hatası.	İş kurallarına uymayan veri (örn. eksik zorunlu alan) gönderilmiştir. Detaylar yanıt gövdesinde olabilir.
404 Not Found	Kaynak bulunamadı.	İstekteki mukellefid gibi parametrelerin doğruluğunu kontrol edin.
500 Internal Server Error	Sunucu hatası.	Sistemle ilgili genel bir hata oluşmuştur.

□ Not: API'yi Hızlıca Test Etmek İçin

Geliştiriciler, bu dökümana ek olarak sağlanan Swagger adresinden yada Postman Collection dosyasını içe aktararak tüm uç noktaları doğrudan test edebilirler.

Bu yöntem, entegrasyon öncesi istek yapılarını ve yanıt formatlarını hızlıca anlamak ve denemek için önerilir.

Swagger UI adresi: <https://test-efatura.maliye.gov.ct.tr/api/public/swagger-ui>

Postman için Adımlar:

1. [Postman](#) uygulamasını indirin veya web üzerinden açın.
2. Import menüsüne tıklayın.
3. Talep üzerine, tarafınıza ilettiğimiz postman_collection.json dosyasını içe aktarın.
4. İlgili istekleri test etmeye başlayın.

Collection, örnek JSON gövdeleri ve yanıt örnekleri ile birlikte gelmektedir.
