



API Yazılım Klavuzu

GOVERNID API DOKÜMANTASYON

Teknik Doküman



Bu dokümandaki herhangi bir bilginin kullanılması halinde dokümanın “Kullanım Şartları” bölümünde belirtilen şartlar kabul edilmiş sayılır. Bu şartların kabul edilmemesi durumunda doküman kullanılmamalıdır.

İçindekiler

GİRİŞ.....	2
API AKTİVASYONU	3
İŞLEMLER	4
1. API’ya veri gönderilmesi ve işlem kodunun alınması	7
2. Karekod ile doldurulan formun işlem onayının gönderilmesi	7
3. Veri Giriş Noktasına gelen kişisel verilerin listelenmesi	8
4. Veri Giriş Noktasına gelen ve zaman aşımına giren (expired) kişisel verilere ait işlem kodlarının	10
a. Listelenmesi (varsa kurum içi/dışı akışları ile)	10
b. Sunucuya silinme/anonimleştirme onayının gönderilmesi	11
5. Kurum Veri Akış Hatları’na bağlı Rıza Yönetimi kullanılıyorsa, rızası geri çekilen akış kodlarının	12
a. Listelenmesi	12
b. Sunucuya onay bildiriminin yapılması	13
KULLANIM ŞARTLARI.....	14
A. Dokümana erişim ve dokümanın kullanımı:.....	14
B. Bu dokümandan yazılı ve/veya görsel nitelikte bilgi indirme ve kullanma:	14
1. Kişisel kullanım;	14
2. Kişisel olmayan kullanım;.....	15
C. Sorumluluk Reddi	15

GİRİŞ

GovernID, 6698 Sayılı Kişisel Verileri Koruma Kanunu'nun (**KVKK**) öngördüğü şekilde hem kurumlara, hem de son kullanıcılara kişisel veri yönetim araçları sunar. Bu araçlar şunları sağlar:

Kurumlar için:

- Kişisel Verilerin yasal düzenlemelere uygun şekilde alınmasını sağlamak.
- Kişisel Verilerin alınırken rıza yönetimini sağlamak.
- Kişisel Verilerin kurum içindeki veya kurum dışına akışını takip etmek.
- Kişisel Veri zaman aşımını takip etmek.
- Geri çekilen rızaları takip etmek.

Son Kullanıcılar için:

- Hangi kuruma hangi verisini verdiğini takip etmek.
- Zaman aşımına uğrayan verisinin silinip silinmediğini takip etmek.
- Vermiş olduğu rızaları geri çekebilmek.
- Geri çekmiş olduğu rızaların onaylanıp onaylanmadığını takip etmek.

GovernID, kurumların, üstte belirtilen hukuki sorumluluklarını yerine getirebilmesi için **HTTP1.1** protokolü ile çalışan bir **API** servisi hizmete sunmuştur. API servisi karşılıklı veri transferi için **JSON** veri tipini kullanır.

API servisi sayesinde kurumlar şu **işlemleri** gerçekleştirebilirler:

1. API'ya veri gönderilmesi ve işlem kodunun alınması
2. Karekod ile doldurulan formların işlem onayının gönderilmesi
3. Veri Giriş Noktasına gelen kişisel verilerin listelenmesi (yetkiye göre açık ve maskelenmiş)
4. Veri Giriş Noktasına gelen ve zaman aşımına giren (expired) kişisel verilere ait işlem kodlarının
 - a. Listelenmesi (varsa kurum içi/dışı akışları ile)
 - b. Sunucuya silinme/anonimleştirme onayının gönderilmesi
5. Kurum Veri Akış Hatları'na bağlı Rıza Yönetimi kullanılıyorsa, rızası geri çekilen akış kodlarının
 - a. Listelenmesi
 - b. Sunucuya onay bildirimini yapılması

API AKTİVASYONU

API hizmetinin kullanılabilmesi için, bu hizmetin kurum yetkilisi tarafından aktive edilmesi gerekmektedir. Aktivasyon işlemi ücretsizdir.

Kurum yetkilisi, uygulama içindeki “Kurum Ayarları” menüsünden API Erişimi sayfasını açıp, açılan sayfada sağ alttaki (+) butonunu tıklayıp API erişim anahtarını oluşturur. Oluşturma sırasında, kişisel veri erişiminin aktif/pasif durumunu ayarlar. Bu aktif edilirse veriler açık şekilde, pasif edilirse veriler **** şeklinde maskelenmiş olarak listelenecektir. Ayrıca hemen altında, oluşturulan anahtarın hangi formlarda işlem yapma yetkisi olacağını seçer. Bir API anahtarı birden fazla formda işlem yapabilir. Son bölümde ise API’ya erişmesi istenen sunucuların IP adreslerini tanımlar. Kurumunuzun içinde ise kurumunuzun internete çıkan sabit IP adresi olması önerilir ve bu IP adresi API erişimi için tanımlanmalıdır. Bu sayede API erişiminde kullanılacak erişim anahtarına ek olarak 2nci bir güvenlik katmanı oluşturulmaktadır.

IP adresi tanımında, X.X.X.X şeklinde IPv4 adres girilmesi gerekir. Aynı network içinde birden fazla çıkış IP adresi var ise, bunlar alt alta tanımlanabileceği gibi, ilgili subnet’i ifade eden CIDR notasyonu kullanılabilir. Örneğin kurumunuza ait 1.2.3.1 – 1.2.3.253 IP bloğunun tamamına API erişimi vermek isterseniz, 1.2.3.0/24 şeklinde tanımlayabilirsiniz.

Her ne kadar önerilmese de, **sabit IP adresiniz olmaması durumunda** IP kısıtlaması alanına **0.0.0.0/0** şeklinde kayıt girmek suretiyle herhangi bir IP adresinden erişimi açık hale getirebilirsiniz.

Belirlenen IP adresleri dışında API erişimi gerçekleşmeyecektir.

IP adreslerinin tanımlanması sonrasında sistem otomatik olarak API erişim anahtarı üretir ve kurumunuza tanımlar.

Kurum dilediği kadar **API anahtarı** oluşturabilir ve bunları farklı yetkilerle ve farklı IP adreslerinden erişecek şekilde ayarlayabilir.

Kurum verilen API anahtarını kullanarak, yukarıda belirtilen işlemleri programatik olarak gerçekleştirebilecektir.

Not: API anahtarları UUID v4 formatındadır.

İŞLEMLER

API'ya gönderilecek isteklerin türü (REQUEST_TYPE), **POST** olmalıdır.

Gönderilen isteklerde header paketi içinde **Content-Type: application/json** olarak belirtilmelidir.

Kimlik doğrulama için; yine header paketi içinde **Governid-Apikey** parametresi ile **API anahtarı** gönderilmelidir.

Tüm API isteklerinde {success: true, ...} şeklinde veya hatalı bir istek göndermişseniz {success: false, reason: "hata nedeni açıklaması"} şeklinde bir yanıt dönecektir.

Örnek API İsteği (PHP):

```
<?php
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "https://api.governid.com/api_endpoint_url");
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Governid-Apikey: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX'
));
...
...
...

```

Not: Üstteki örnekteki gösterilen curl_init ve curl_setopt fonksiyonları PHP_CURL uzantısının fonksiyonlardır. Bunların kullanılabilmesi için PHP_CURL paketi yüklü olmalıdır.

APT kullanan Linux dağıtımları için: *apt-get install php8.0-curl (php versiyonunuza göre 8.0 yerine uygun versiyon kodunu yazınız)*

FreeBSD 10+ için: *pkg add php80-curl (php versiyonuna göre 8.0 yerine uygun versiyon kodunu yazınız)*

FreeBSD <10 için: *cd /usr/ports/lang/phpXX-extensions/ sonra make config & make install clean (XX: versiyon)*

Örnek API İsteği (JavaScript/NodeJS):

```
var xhr = new XMLHttpRequest();
xhr.open('POST', 'https://api.governid.com/api_endpoint_url');
xhr.setRequestHeader('Content-Type', 'application/json');
xhr.setRequestHeader('Governid-Apikey', 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX');
...
...
...

```

Listeleme sonuçlarında bir takım temel arama/filtreleme/sayfalama/sıralama işlemleri de yapılabilmektedir. Bunlar header içinde değil, gövde (body) paketi olarak POST edilmelidir. Bunların detaylı kullanım şekilleri ilerleyen bölümlerde anlatılacaktır. Herhangi bir kriter belirtilmediği takdirde varsayılan sonuç kümesi, kayıt tarihe göre tersten sıralı, yani **son 100 kayıt** olarak listelenecektir.

Veri Paketi (JSON)

API'ya gönderilecek veri paketleri JSON encode edilmiş ve string olarak encode edilmiş olmalıdır. Yaygın kullanılan programlama dilleri (ASP.NET, PHP, Python, Javascript, Java, VB, C# vb.) , JSON veri tipinin açılması/paketlenmesi (encode/decode) fonksiyonlarını içermektedir.

Not: Göndereceğiniz JSON veri tiplerinin geçerliliğini <https://jsonlint.com/> adresinden kontrol edebilirsiniz.

Örnek API İsteği (PHP):

```
<?php
$data = array(
    "param1"=>"value1",
    "param2"=>"value2"
);
$data_encoded = json_encode($data, JSON_PRETTY_PRINT | JSON_UNESCAPED_UNICODE);
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "https://api.governid.com/api_endpoint_url");
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Governid-Apikkey: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX'
));
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_encoded);
curl_exec($ch);
...
```

Örnek API İsteği (Javascript/NodeJS):

```
var data = {
    param1: 'value1',
    param2: 'value2'
};
var data_encoded = JSON.stringify(data);
var xhr = new XMLHttpRequest();
xhr.open('POST', 'https://api.governid.com/api_endpoint_url');
xhr.setRequestHeader('Content-Type', 'application/json');
xhr.setRequestHeader('Governid-Apikkey', 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX');
xhr.send(data_encoded);
...
```

Farklı programlama dillerinde JSON encode/decode fonksiyonları:

.NET: <https://docs.microsoft.com/en-us/dotnet/api/system.web.helpers.json.encode?view=aspnet-webpages-3.2>

PYTHON: <https://pynative.com/python-json-encode-unicode-and-non-ascii-characters-as-is/>

PERL: <https://metacpan.org/pod/JSON>

JAVA: <https://www.javatpoint.com/java-json-example>

Karakter kodlaması

API'ya gönderilen ve API'dan alınan tüm verilerin karakter kodlaması (character encoding) UTF-8 formatındadır.

Zaman Dilimi

API'ya gönderilen ve API'dan alınan tüm verilerin zaman dilimi UTC olarak ayarlıdır. İlerleyen zamanlarda, API erişim ayarlarınızda ön tanımlı bir zaman dilimi seçilebilmesi imkanı tanınacaktır.

Zaman Gösterim Formatı

HTTP protokolü sunucu-istemci veri transferinde, STRING tipi veriler ile çalıştığı için, zaman değerleri de STRING formatında gönderilmelidir. API altyapımızdaki JSON iletişim standardı, timestamp değerlerini otomatik olarak ISO8601 formatında göstermektedir. Avantajı, tüm yaygın kullanılan yazılım dillerinin bu formatı tanınması, format dönüşümü yapılmaya dahi string halinin sıralanabilir olmasıdır. Bkz. https://en.wikipedia.org/wiki/ISO_8601

Örnek Gösterim: 2018-06-12T16:52:35Z

Kendi yazılımınız tarafında saat dilimini Europe/Istanbul, gösterim tipini de German (dd.mm.yyyy) olarak ayarlarsanız Türkiye UTC+3 olduğu için, compiler bu işlemi tarihini 12.06.2018 19:52 olarak gösterecektir. Yazılımınızda saat dilimini America/New York olarak ayarlarsanız, istemci tarafında otomatik olarak kışın 06/12/2018 11:52 yazın ise 06/12/2018 12:52 olarak gösterilecektir. (Newyork kış saati UTC-5 yaz saati UTC-4)

Telefon Numaraları

Kişisel verilerin açık listelenmesi durumunda telefon numaraları E.164 formatında ve boşluk içermeyecek şekilde +CCXXXXXX... olarak gösterilecektir. (CC ülke telefon kodu) Bkz. <https://www.itu.int/rec/T-REC-E.164-201011-I/en>

Kişisel Veri Alanları (API sonuçlarında user_data)

Bu alanlar, uygulama içinde sabit olarak tanımlanmış bir takım kişisel veri tiplerini belirtir. Sürekli olarak yeni user_data tipleri eklenmektedir.

_FULLNAME: Ad Soyad

_EMAIL: E-posta Adresi

_EMAIL_VERIFIED: true/false BOOLEAN değer

_TEL: Telefon Numarası (+CCXXXXXXXX şeklinde, CC: Country Code)

_TEL_VERIFIED: true/false BOOLEAN değer

_TCKN: TC Kimlik Numarası

_PHOTO: base64 encoded string (bunu olarak direk kullanabilirsiniz
...

_COMPANY_TITLE: Çalıştığı Firma Ünvanı (veri alınırken kurumsal profil seçilmiş ise)

_WORK_TITLE: İş Ünvanı (veri alınırken kurumsal profile seçilmiş ise)

Ziyaretçi Giriş Kapıları için, API şu alanları da kişisel veri alanında listeler

_VISITING_TO: Ziyaret edilen kişi

_VISITING_REASON: Ziyaret Nedeni

1. API'ya veri gönderilmesi ve işlem kodunun alınması

API Endpoint: https://api.governid.com/v2/submit/{form_uuid}

Örnek API İsteği (JavaScript/NodeJS):

```
var postData = {
  _FULLNAME: 'Ahmet YILMAZ',
  _EMAIL: 'xxxxxxxxx@xxxxxx.xxx',
  _TEL: '+90XXXXXXXXXX',
  ...
}
var postDataEncoded = JSON.stringify(postData);
var xhr = new XMLHttpRequest();
xhr.open('POST', 'https://api.governid.com/v2/submit/{form_uuid}');
xhr.setRequestHeader('Content-Type', 'application/json');
xhr.setRequestHeader('Governid-Apikey', 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX');
xhr.send(postDataEncoded);
...
...
```

Örnek API yanıtı:

```
'{"success":true,"transids":["ab12xy34","cd56tz78"]}'
```

Sunucu yanıtında normal şartlarda tek işlem kodu dönmesi gerekir ancak bazı durumlarda e-posta adresi başka kullanıcıya, telefon ise bir diğer kullanıcıya ait olması durumu olabilir. Bu durumda GovernID tarafında iki ayrı kişi için KVKK işlem kaydı gerçekleşir. Burada olması beklenen; işlemi yapmamış olan diğer son kullanıcının uygulamayı verimli kullanıyor olması ve işlemi kendisinin yapmadığı gerekçesiyle itiraz etmesidir. Bu şekilde bir talep olursa sizin de o kişiye ait işlem koduna bağlı kayıtları veritabanınızdan silmeniz gerekir.

2. Karekod ile doldurulan formun işlem onayının gönderilmesi

API Endpoint: <https://api.governid.com/v2/verify/{transid}>

Karekod ile doldurulan formlarda, sayfanızdaki forma sistem tarafından otomatik olarak bir adet gizli (hidden) governid_transid isminde input eklenir ve içine işlem kodu (transid) değeri atanır. Form gönderildiğinde (submit edildiğinde) backend kodunuzda bu değerli alıp işlemin tamamlandığı bilgisini GovernID API suncusuna iletmeniz gerekir.

Örnek API İsteği (JavaScript/NodeJS):

```
var xhr = new XMLHttpRequest();
xhr.open('POST', 'https://api.governid.com/v2/verify/{transid}');
xhr.setRequestHeader('Content-Type', 'application/json');
xhr.setRequestHeader('Governid-Apikey', 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX');
xhr.send(postDataEncoded);
...
...
```

Örnek API yanıtı:

```
'{"success":true}'
```


3. Veri Giriş Noktasına gelen kişisel verilerin listelenmesi

API Endpoint (yetki verilen tüm formlar): <https://api.governid.com/v2/entries>

API Endpoint (tek form için): <https://api.governid.com/v2/entries/{form uuid}>

Örnek API isteği (JavaScript/NodeJS):

```
var xhr = new XMLHttpRequest();
xhr.open('POST', 'https://api.governid.com/v2/entries');
xhr.setRequestHeader('Content-Type', 'application/json');
xhr.setRequestHeader('Governid-Apikey', 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX');
xhr.send();
```

Bu işlem neticesinde son 100 kayıt JSON stringified object olarak dönecektir.

Gelen veride:

totalPages: Toplam kaç sayfa veri olduğunu, sayfa numarası belirtilmemişse toplam veri adedi / 100 olarak

rows: Veriler array formatında, array elementleri (row) ise object formatındadır

row.form_uuid: Verinin alındığı form'un tanımlayıcı kodu (uuid v4)

row.indate: ISO8601 formatında UTC zaman diliminde işlem tarihini gösterir

row.transid: Veritabanında ilişkilendireceğiniz işlem kodunu içerir

row.userdata: Object formatında, veri giriş noktasında toplanan **verileri** listeler

Not: user_data olarak veri giriş noktasında farklı parametreler belirtilebildiği için bu alanlar ayrıca bir nesne içinde gösterilir.

Örneğin, A girişinde ad soyad ve telefon, B girişinde ad soyad, e-posta ve telefon alınmış olabilir.

Örnek API yanıtı:

```
{
  "success": true,
  "totalPages": 24,
  "rows": [
    {
      "form_uuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
      "transid": "ab12xy34",
      "indate": "2018-06-12T16:52:35Z",
      "user_data": {
        "_FULLNAME": "Ahmet YILMAZ",
        "_EMAIL": "xxxxxxxx@xxxxxx.xxx",
        "_EMAIL_VERIFIED": true,
        "_TEL": "+90XXXXXXXXXX",
        "_TEL_VERIFIED": false
      }
    },
    {
      "form_uuid": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
      "transid": "cd56tz78",
      "indate": "2018-06-13T17:52:35Z",
      "user_data": {
        "_FULLNAME": "Mehmet YILMAZ",
        "_EMAIL": "xxxxxxxx@xxxxxx.xxx",
        "_EMAIL_VERIFIED": false,
        "_TEL": "+90XXXXXXXXXX",
        "_TEL_VERIFIED": true,
        "TCKN": "12345678901"
      }
    }
  ]
}
```

Gelen veri, json decode, json parse gibi fonksiyonlarla nesne (object) yapısına dönüştürülebilir. Örneğin API üzerinden aşağıdaki gibi bir **data** dönmüş olsun, Chrome Developer Tools konsolunda bunu deneyebiliriz:

```
> var data = '{"success":true,"totalPages":24,"rows":[{"form_uuid":"XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX", "transid":"ab12xy34", "indate":"2018-06-12T16:52:35Z", "user_data":{"_FULLNAME":"Ahmet YILMAZ", "_EMAIL": "xxxxxxxx@xxxxxx.xxx", "_EMAIL_VERIFIED": true, "_TEL": "+90XXXXXXXXXX", "_TEL_VERIFIED": false}}, {"form_uuid":"XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX", "transid":"cd56tz78", "indate":"2018-06-13T17:52:35Z", "user_data":{"_FULLNAME":"Mehmet YILMAZ", "_EMAIL": "xxxxxxxx@xxxxxx.xxx", "_EMAIL_VERIFIED": false, "_TEL": "+90XXXXXXXXXX", "_TEL_VERIFIED": true, "TCKN": "12345678901"}}]';
< undefined
> JSON.parse(data);
< {success: true, totalPages: 24, rows: Array(2)}
  rows: Array(2)
    0: {
      form_uuid: "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"
      indate: "2018-06-12T16:52:35Z"
      transid: "ab12xy34"
      user_data: {_FULLNAME: "Ahmet YILMAZ", _EMAIL: "xxxxxxxx@xxxxxx.xxx", _EMAIL_VERIFIED: true, _TEL: "+90XXXXXXXXXX", _TEL_VERIFIED: false}
      __proto__: Object
    }
    1: {form_uuid: "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX", transid: "cd56tz78", indate: "2018-06-13T17:52:35Z", user_data: {}}
      length: 2
      __proto__: Array(0)
    success: true
    totalPages: 24
    __proto__: Object
  }
> JSON.parse(data).rows.forEach(row=>{
  console.log(
    new Date(row.indate).toLocaleString('tr'),
    row.transid,
    row.user_data._FULLNAME
  );
});
12.06.2018 19:52:35 ab12xy34 Ahmet YILMAZ VM54289:2
13.06.2018 20:52:35 cd56tz78 Mehmet YILMAZ VM54289:2
```

Filtreleme/Sayfalama işlemleri

Bu parametreler bir nesneye atanıp, API'ya JSON encoded string olarak POST edilmelidir. Tüm parametreler opsiyoneldir ve tek tek kullanılabilir.

Sayfalandırma

Parametre: **paging**

Değer: 5-500 arası herhangi bir INTEGER değer.

Varsayılan değeri 100 olarak ayarlıdır. Veri listeleme sırasında her bir sayfada kaç satır çekileceğini belirler. Örneğin sayfa 1'de 1-100 arası kayıtlar, sayfa 2'de 101-200 arası kayıtlar listelenir. Sayfalandırmayı 50 olarak belirlerseniz, sayfa 1'de 1-50 arası kayıtlar, sayfa 2'de 51-100 arası kayıtlar listenecektir.

Sayfa No

Parametre: **page**

Değer: Toplam sayfa adedine kadar 1 veya daha büyük INTEGER, varsayılan 1

Varsayılan değeri 1 olarak ayarlıdır. İlk istek atıldıktan sonra gelen sonuç kümesinde totalPages değerindeki toplam sayfa adedini kullanınız.

Sadece toplam sayfa için API Endpoint: <https://api.governid.com/v2/entries/total> veya https://api.governid.com/v2/entries/total/{form_uuid} adresine istek gönderiniz. {totalPages: 24} şeklinde yanıt dönecektir.

Sıralama

Parametre: **sortby**

Değer: indate|transid|(user_data içindeki olası değerler _FULLNAME, _EMAIL gibi), DEFAULT: indate

Sıralama Yönü

Parametre: **sorttype**

Değer: ASC|DESC, DEFAULT: DESC

Arama

Parametre: **query**

Değer: en az 2 karakter uzunluğunda metin, tüm veri alanlarında aranır

Tarihe Göre Filtreleme

Parametreler: **date_after, date_before**

Değer: YYYY-MM-DD olarak değer verilmelidir. >= ve <= olarak aranacaktır. Tek tek veya ikisi birlikte kullanılabilir.

Örnek API İsteği (JavaScript/NodeJS):

```
var postData = {
  paging: 50,
  page: 2,
  sortby: '_FULLNAME',
  sorttype: 'ASC',
  query: '%test%',
  date_after: '2019-06-01'
}
var postDataEncoded = JSON.stringify(postData);
var xhr = new XMLHttpRequest();
xhr.open('POST', 'https://api.governid.com/v2/entries');
xhr.setRequestHeader('Content-Type', 'application/json');
xhr.setRequestHeader('Governid-Apikey', 'XXXXXXXX-XXXX-XXXX-XXXXXXXXXXXX');
xhr.send(postDataEncoded);
...
```

4. Veri Giriş Noktasına gelen ve zaman aşımına giren (expired) kişisel verilere ait işlem kodlarının

a. Listelenmesi (varsa kurum içi/dışı akışları ile)

API Endpoint: <https://api.governid.com/v2/expired> veya

API Endpoint: https://api.governid.com/v2/expired/{form_uuid}

Örnek API isteği (Javascript/NodeJS):

```
var xhr = new XMLHttpRequest();
xhr.open('POST', 'https://api.governid.com/v2/expired');
xhr.setRequestHeader('Content-Type', 'application/json');
xhr.setRequestHeader('Governid-Apıkey', 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX');
xhr.send();
...
...
```

Bu işlem neticesinde **onaylamanızı bekleyen** son 100 kayıt JSON stringified object olarak dönecektir.

Filtreleme

1. Adımda belirtilen Arama (query) hariç tüm filtreleme/sayfalama parametreleri kullanılabilir.

Gelen veride:

totalPages: Toplam kaç sayfa veri olduğunu, sayfa numarası belirtilmemişse toplam veri adedi / 100 olarak

rows: Veriler array formatında, array elementleri (row) ise object formatındadır

row.indate: ISO8601 formatında UTC zaman diliminde işlem tarihini gösterir

row.transid: Veritabanında ilişkilendireceğiniz işlem kodunu içerir

row.userdata: Object formatında, veri giriş noktasında toplanan **veri tiplerini** listeler

pipes: (opsiyonel) Veri Akış Hatları'nızı tanımlamışsanız, verilerin iç/dış akış yolları, akış hatları akış sırasına göre array olarak listelenir

Örnek API yanıtı:

```
'{"success":true,"totalPages":11,"rows":[{"transid":"ab12xy34","indate":"2018-06-12T16:52:35Z","user_data":{"_FULLNAME","_EMAIL","_PHONE"},"pipes":["pipe1","pipe3","pipe5"]},{"transid":"cd56tz78","indate":"2018-06-13T17:52:35Z","user_data":{"_FULLNAME","_EMAIL","_PHONE","_TCKN"},"pipes":["pipe1","pipe2","pipe6"]}']'
```

Gelen veri setindeki işlem kodlarına bağlı hangi kişisel veri alanlarının çekildiği bilgisi, user_data içinde gösterilir.

Örnek olarak:

ab12xy34 kodlu işlem sırasında kullanıcıdan ad soyad, e-posta ve telefon alınmış olduğu görülüyor. **Hukuki bir zorunluluk olarak**, (basılı veya dijital) tüm veri kayıt alanlarınızdan bu kodla ilişkili ad soyad, e-posta, telefon numaralarını silmeniz veya anonimleştirmeniz gerekiyor.

Anonimleştirme; verilerin başka verilerle eşleştirilse dahi kimliği belirli veya belirlenebilir bir gerçek kişiyle ilişkilendirilemeyecek hale getirilmesidir. ... Veri sorumlusu, kişisel verilerin anonim hale getirilmesiyle ilgili gerekli her türlü teknik ve idari tedbirleri almakla yükümlüdür.

Diğer cd56tz78 adlı koda bağlı olarak fazladan TCKN verisi de alındığı görülmekte. Bu durumda bu kodla ilişkili olan kayıtlarda, ekstra olarak TCKN içeren verileri de silmeniz veya anonimleştirmeniz gerekmektedir.

b. Sunucuya silinme/anonimleştirme onayının gönderilmesi

API Endpoint: https://api.governid.com/v2/expired_feedback

Onay gönderimleri tek tek veya toplu işlem kodları için yapılabilir. Array halindeki işlem kodlarını JSON stringified olarak API sunucusuna post etmelisiniz.

Örnek API isteği (JavaScript/NodeJS):

```
var postData = [  
  'ab12xy34',  
  'cd56tz78'  
];  
var postDataEncoded = JSON.stringify(postData);  
var xhr = new XMLHttpRequest();  
xhr.open('POST', 'https://api.governid.com/v2/expired_feedback');  
xhr.setRequestHeader('Content-Type', 'application/json');  
xhr.setRequestHeader('Governid-Apikey', 'XXXXXXXX-XXXX-XXXX-XXXXXXXXXXXX');  
xhr.send(postDataEncoded);  
...  
...
```

Bu işlem neticesinde post edilen array içindeki işlem kodlarının silindiği veya anonimleştirildiği bildirilir.

İşlem başarılı olsun ya da olmasın sunucu tarafından success: true şeklinde bir sonuç döndürülür

Örnek API Yanıtı:

```
' {"success":true}'
```

5. Kurum Veri Akış Hatları'na bağlı Rıza Yönetimi kullanılıyorsa, rızası geri çekilen akış kodlarının

a. Listelenmesi

API Endpoint: <https://api.governid.com/v2/revoked>

Not: Rızası geri çekilen işlem kodlarının listelenebilmesi için Veri Akış Hatları (PIPE) ve Veri Geçiş/Kayıt Noktaları (NODE) tanımlanmış olmalıdır.

Örnek API isteği (JavaScript/NodeJS):

```
var xhr = new XMLHttpRequest();
xhr.open('POST', 'https://api.governid.com/v2/revoked');
xhr.setRequestHeader('Content-Type', 'application/json');
xhr.setRequestHeader('Governid-Apikey', 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX');
xhr.send();
...
...
```

Bu işlem neticesinde **onaylamanızı bekleyen** son 100 kayıt JSON stringified object olarak dönecektir.

Gelen veride:

totalPages: Toplam kaç sayfa veri olduğunu, sayfa numarası belirtilmemişse toplam veri adedi / 100 olarak

rows: Veriler array formatında, array elementleri (row) ise object formatındadır

row.indate: ISO8601 formatında UTC zaman diliminde işlem tarihini gösterir

row.transid: Veritabanında ilişkilendireceğiniz işlem kodunu içerir

row.userdata: Object formatında, veri giriş noktasında toplanan **veri tiplerini** listeler

pipes: (opsiyonel) Tanımlamışsanız, verilerin iç/dış akış yolları, akış sırasına göre sıralı array içinde listelenir

Örnek API yanıtı:

```
'{"success":true,"totalPages":11,"rows":[{"transid":"ab12xy34","indate":"2018-06-12T16:52:35Z","revoked_from":["pipe1"]}, {"transid":"cd56tz78","indate":"2018-06-13T17:52:35Z","revoked_from":["pipe3"]}]]'
```

Gelen veri setindeki işlem kodlarına bağlı hangi rızanın geri çekildiği bilgisi, revoked_from içinde gösterilir. Burada belirtilen pipe ismi, tanımladığını rıza işlemine tabi veri akış hattının tarafınızca verilmiş bulunan kodudur.

Örnek olarak:

ab12xy34 kodlu işlem sırasında kullanıcıdan pipe1 olarak tanımlanmış bir rızası alınmış olduğu ve bunun kullanıcı tarafından geri çekilmek istendiği görülüyor. **Hukuki bir zorunluluk olarak**, (basılı veya dijital) tüm veri kayıt alanlarınızdan bu pipe sırasında aktarılan kişisel veri aktarımının durdurulması gerekiyor.

b. Sunucuya onay bildiriminin yapılması

API Endpoint: https://api.governid.com/v2/revoked_feedback

Onay gönderimleri tek tek veya toplu işlem kodları için yapılabilir. Array halindeki işlem kodlarını JSON stringified olarak API sunucusuna post etmelisiniz.

Örnek API isteği (JavaScript/NodeJS):

```
var postData = [
  {
    transid: 'ab12xy34',
    revoked_from: 'pipe1'
  },
  {
    transid: 'cd56tz78',
    revoked_from: 'pipe3'
  }
];
var postDataEncoded = JSON.stringify(postData);
var xhr = new XMLHttpRequest();
xhr.open('POST', 'https://api.governid.com/v2/revoked_feedback');
xhr.setRequestHeader('Content-Type', 'application/json');
xhr.setRequestHeader('Governid-Apikey', 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX');
xhr.send(postDataEncoded);
...
...
```

Bu işlem neticesinde post edilen array içindeki işlem kodlarına bağlı rızaların geri çekildiğinin onaylandığı sunucuya bildirilir.

İşlem başarılı olsun ya da olmasın sunucu tarafından success: true şeklinde bir sonuç döndürülür

Örnek API Yanıtı:

```
' {"success":true}'
```

KULLANIM ŞARTLARI

Bu doküman GovernID™'den hizmet alan kurumlar için oluşturulmuş bir dokümandır. Doküman içerisinde yer alan bilgiler yayınlandığı tarih itibari ile kullanımda olmakla beraber sürekli yapılan geliştirme çalışmaları sonucu değişiklikler olabilir. Bu nedenle yapacağınız çalışmalardan önce GovernID™ ile temasa geçmenizi tavsiye ediyoruz.

A. Dokümana erişim ve dokümanın kullanımı:

1. A. Dokümana erişim ve dokümanın kullanımı, GovernID™ nin belirlediği aşağıdaki esas ve şartlara tabidir.
2. Bu dokümanda yer alan tüm bilgiler GovernID™den izin alınmaksızın değiştirilemez, çoğaltılamaz, yayımlanamaz, dağıtılamaz, umuma iletilemez, başka bir lisana çevrilemez.
3. Dokümanı kullanan kişinin, dokümanın ilk defa kullanım anından itibaren geçerlilik kazanan "Kullanım Koşulları" kabul edilmiş sayılır. GovernID™, "Kullanım Koşullarını", değişiklikleri çevrim içi yayınlamak koşulu ile her an değiştirme hakkını saklı tutar. Çevirim içi yayınlanan değişikliklerden zamanında haberdar olmak için "Kullanım Koşullarını" düzenli olarak takip etmek, doküman kullanıcısının sorumluluğundadır. Yapılan değişikliklerden sonra dokümanı kullanmaya devam eden kişi, "Kullanım Koşulları"ndaki değişiklikleri kabul etmiş sayılır.
4. GovernID™ 24 saat erişilebilir olması için çalışmaktadır. Ancak değişik sebeplerle Sitenin erişilebilir olmamasından sorumlu değildir.
5. GovernID™, doküman kullanıcılarına sunduğu bilgilerin doğru olması için her türlü gayreti göstermekle beraber, dokümanın güncelliği konusunda GovernID™ ile temasa geçilmemesi durumunda yanlış veya eksik bilgiden sorumlu değildir.

B. Bu dokümandan yazılı ve/veya görsel nitelikte bilgi indirme ve kullanma:

1. Kişisel kullanım;

- 1.1. Dokümanda yer alan bilgilerin kişisel kullanım ve/veya bilgi edinme amacı ile sınırlıdır. Ancak bu kapsamda gerçekleştirilen çoğaltma fiilleri hak sahibinin meşru menfaatlerine haklı bir sebep olmadan zarar veremez ya da eserden normal yararlanmaya aykırı olamaz, ticari amaçla kullanılamaz.
- 1.2. Dokümandaki yazılı veya görsel materyal hiçbir şekilde değiştirilemez, telif hakkı ibareleri silinerek kullanılamaz.
- 1.3. Dokümanda yer alan bilgilerin bütünü veya bir kısmı değiştirilerek veya başka bir suretle diğer bir dokümanda/internet sitesinde izinsiz olarak kullanılamaz.
- 1.4. Bu dokümandaki veriler ticari olmayan, bilgi alma amaçlı ve kişisel kullanım için indirebilir veya yazdırılabilir.
- 1.5. Bu doküman her hangi bir ticari amaç olmadan üçüncü şahıslara onların kişisel bilgilendirilmeleri için, içeriğin GovernID™ tarafından sağlandığını ve bu kayıt ve şartların onlara da uygulandığını ve bunlara uymak zorunda olduklarını belirtmek şartıyla gönderilebilir.

2. Kişisel olmayan kullanım;

Bu doküman yazılı ve/veya görsel nitelikte bilgi indirme ve baskı alma gibi çoğaltma yahut internet sitelerinde kullanma:

Kullanılacak bilgiye ilişkin izinler, GovernID™ den yazılı olarak talep edilmelidir.

C. Sorumluluk Reddi

GovernID™, elinizdeki dokümanın güncel olduğunu garanti etmemektedir. İşbu dokümanda yer alan beyanlar hukuken taahhüt niteliğinde sayılmamaktadır ve bağlayıcı değildir. Dokümanın güncelliğinin GovernID ile temasa geçilerek yapılacak çalışmalar konusunda bilgi verilip dokümanın güncelliğinin onaylatılmaması durumunda, ancak bunlarla sınırlı kalmamak kaydıyla, zımnen, açıkça ya da yasal olarak hiçbir garanti vermemektedir.