



**NetOpenX REST TestApp**

**Kullanım Dökumanı**

Güncelleme Tarihi: 27.11.2015

*Hazırlayan: Bora KÖKEN*

1. **Yardım Sayfası**

Kurulum sonrası, ya da uygulamaya erişmeden önce kurulum dokümanında bahsedilen servislerin çalışır durumda olduğunun testinin yapılması gerekmektedir.

Bu test sonrası servisler çalışır durumda görünüyor ise adresi yardım sayfasına;
<http://localhost:7070/api/v2/help> üzerinden ulaşılabilir. Desteklenen tüm fonksiyon, aldıkları parametre ve nesne tiplerine detaylıca erişilmektedir.



Bu adres üzerinden ihtiyacımız olan kullanılacak API adresleri seçilir. Bu bilgileri kullanacağımız aracımız ise **“NetOpenX.Rest.TestApp.exe”** nin Operation Sekmesinin ‘Path’ alanıdır.

**\*NetOpenX.Rest.TestApp.exe, Rest kullanımında verilerin doğru alınıp gönderilebildiğini gözlememizi sağlayan test aracıdır.**

1. **NetOpenX.Rest.TestApp.exe**

Bu exe’ye **“***C:\Program Files (x86)\Netsis\Nox\WSManager\Test”*yolu altından erişebilirsiniz.



TestApp açıldıktan sonra gerekli alanlar doldurulur;
**DB Name:** Veri tabanı ismi yazılır

**DB Password:** Veritabanı şifresi yazılır

**Netsis User:** Netsis kullanıcı ismi yazılır

**Netsis Pass:** Netsis şifresi yazılır.



Gerekli alanlar doldurulduktan sonra, **“Get Token”**  butonuna basılarak Token oluşturulur.

**!** Access Token alanına aşağıdaki şekilde bir veri gelirse, Token alma işleminde başarılı olmamış demektir:



*{"StatusCode":400,"IsSuccessStatusCode":false,"error":"invalid\_grant","error\_description":"\r\nSQL : Select Sube\_Kodu,MeRKEZMI,ISLETME\_KODU FROM TBLSUBELER WITH (NOLOCK) Where Sube\_Kodu = 0\r\n\r\nLogin failed for user 'sa'.(NetsisMS-42000,18456)\r\n | NetOpenX50.Kernel"}*

Bu durumda servis durmuş olabilir, **NetOpenXRestManager** *(NetOpenX.Rest.Service.WSManager.exe)*  ile servisin çalışır durumda olduğunu kontrol ediniz.

Token alma işlemi başarılı ise Settings tabında aşağıdaki şekilde bir bilgi oluşur: 

Ardından otomatik olarak Operation tabı aktif olur:



**!** Get Result Butonuna basıldığında aşağıdaki hata geliyorsa, önceki taba gelerek, tekrar Get Token işlemi yapılması gerekmektedir. Gerekiyorsa, uygulama kapatılıp tekrar açılarak Get Token yapılmalıdır.



Gerekiyorsa **NetOpenX Rest WSManager.exe** üzerinden “Access Token Geçerlilik Süresi”ni arttırıp tekrar deneyebilirsiniz:



Token başarı ile alınıp **Operation Tabı** aktif olduktan sonra, buranın header bölümüne path, method, input bilgileri girilir:

**- Path:** <http://localhost:7070/api/v2/help> adresinde (“Yardım Sayfası” konu başlığı altında anlatılan) detaylarını bulanan, kullanılacak *API* bilgisi girilir.
**- Method:** İlgili hangi metodun, yani hangi “paket tipi”nin kullanılacağı bilgisidir.

(Detaylarını aşağıda Methodlar ve Açıklamaları başlığı altında bulabilirsiniz.

* **Input:** Method olarak PUT ya da POST kullanımı sırasında gerekli bilginin yazıldığı alandır.

Ardından “Get Result” butonuna basılır.



Result kısmına gelen bilgi, bizim elde etmek istediğimiz sonuç bilgisini döndürür. Buradaki JSON stringi bir online JSON viewer kullanarak, formatlı ve düzgün bir hallede de görüntüleyebilirsiniz:

Örnek bir site: <http://jsonviewver.stack.hu>



‘Text’ kısmına yazılan JSON, ‘Viewer’ sekmesinde görüntülenebilir.



Ek olarak, ‘Query’ tab ı altından JSON formatında serbest bilgi döndürülebilir.



1. **Methodlar (paket tipleri) ile kullanılabilecek Path (API) örnekleri ve açıklamaları**

**a) GET**

Toplu veri listesi döndürmek ya da tek bir kayıt döndürmek için kullanılır.

Örneğin;

api/v2/ARPs işleminde tüm cari listesini getirirken, api/v2/ARPs/C001 işleminde sadece *C001 carisinin* bilgilerini getirmektedir.

Toplu veri listesi işlemlerinde limit, sayfalama, sıralama, filtreleme gibi bir çok kısıt yapılabilmektedir:

**Limit**

<http://localhost:7070/api/v2/ARPs>?limit=10 her sayfanın kayıtların sayısını belirler. Varsayılan sınır değeri 20'dir.

**Sayfalama**

<http://localhost:7070/api/v2/ARPs>?offset=20 sayfalamanın başladığı değerdir. Varsayılan değer 0 dır.

**Sıralama**

<http://localhost:7070/api/v2/ARPs>?sort=CARI\_KOD ASC şeklinde sıralama yapmaktadır. Birden fazla sıralama verilmesi durumunda ‘,’ ile ayrılmalıdır.

**Alanlar**

<http://localhost:7070/api/v2/ARPs>?fields=CARI\_KOD,CARI\_ISIM sadece belirtilen üst bilgiler döndürülmektedir. Varsayılan değer olarak tüm kolonları döndürmektedir.

**İlk Kayıt**

<http://localhost:7070/api/v2/ARPs>?first=true ilk sonuç kümesini getirir.

**Son Kayıt**

<http://localhost:7070/api/v2/ARPs>?last=true son sonuç kümesini getirir.

**Kısıt**

<http://localhost:7070/api/v2/ARPs>?q=(SQL Select sorgusu) Verilen SQL Select sorgusu sonucu alır. Örnek > ?q=CARI\_KOD='J0001'

**b) DELETE**

Kayıt silme işlemleri olduğunda, paket tipi olarak “Delete” gönderilmesi gerekmektedir.

**c) POST**

Kayıt eklemek istendiğinde mevcut sınıf değerini paket tipi “POST” olarak şekilde gönderilmesi gerekmektedir. Bilgiler ‘Input’ alana doldurulur.

**d) PUT**

Kayıt güncelleme işlemlerinde, paket tipi “Put” olarak sadece değişecek sınıf değerlerini göndermek yeterli olacaktır. Bilgiler ‘Input’ alana doldurulur.

**! Rest Servis Erişimi:**

Rest üzerindeki “public” işlemler dışındaki diğer fonksiyonlar için “access token” gerekmektedir.