

KALİTE DOKÜMANIRev. Tarihi: 21.05.2017
Rev. No: 02Doküman No: YG.DD.009
Sayfa No: 1

Konu: GMS.NET API Dokümanı

Revizyon No**Açıklama****Revizyon Tarihi**01
02İlk Yayın
5.17' de 2. Sınıf firmalarla ilgili kesinleştirme tarihini gelir/gider bazında alabilmek için yeni üç metot eklendi.21.05.2017
29.06.2017**HAZIRLAYAN****ONAYLAYAN****Y. İ.****R. İ.**

Konu: GMS.NET API Dokümanı

İçindekiler

1- AMAÇ	4
2- KAPSAM	4
3- TANIMLAR	4
4- GENEL KAVRAMLAR	4
4.1 API Genel Yapısı	4
4.2 GMS.NET API Aracılığıyla Aktarılabilecek Bilgiler	4
4.3 GMS.NET API Kullanımına Uygun Senaryolar	4
4.4 GMS.NET API' nin Bulunduğu Dizin	4
4.5 GMS.NET API' nin Bir Projeye Referans Olarak Eklenmesi	5
4.6 API Kullanımı Örnek Projesi	5
4.7 API' nin Başlatılması	5
4.8 API Çalışma Testi	6
4.9 API Aracılığıyla GMS.NET' e Giriş Yapma	6
4.10 İzin Verilen Firma Listesi	6
4.11 İzin Verilen Dönem Listesi	6
4.12 Dönem Seçimi	7
4.13 Durum Öğrenme	7
4.14 Defter Tip Kodları	7
4.15 Belge Tiplerinin Listesi	8
5- 1. SINIF FİRMALARLA İLGİLİ İŞLEMLER	8
5.1 Hesap Planı Listesi	8
5.2 Yeni Hesap Ekleme	8
5.3 Hesap Bilgi Güncelleme	8
5.4 Hesap Silme	9
5.5 Hesap Fiş Girişinde Kullanılabilir mi?	9
5.6 Hesap Detayı	9
5.7 Stok Kodu Listesi	9
5.8 Yeni Stok Ekleme	9
5.9 Stok Bilgi Güncelleme	10
5.10 Stok Silme	10
5.11 Cari Listesi	10
5.12 Yeni Cari Ekleme	10
5.13 Cari Bilgi Güncelleme	11
5.14 Cari Silme	11
5.15 DeveloperID Kavramı	11
5.16 Standart Fiş, Özel Fiş ve Özel Alan Kavramları	11
5.17 Fiş Kesinleştirme Tarihi Kavramı	11
5.18 Standart Fiş Ekleme	12

Konu: **GMS.NET API Dokümanı**

5.19	Özel Fiş Ekleme.....	12
5.20	Özel Fiş Kontrol.....	12
5.21	Özel Alan Bilgili Fiş Satırlarının Silinmesi	12
5.22	İki Tarih Arası Fiş Silinmesi.....	13
6-	2. SINIF FİRMALARLA İLGİLİ İŞLEMLER	14
6.1	OKH Listesi.....	14
6.2	Yeni Hesap Ekleme	14
6.3	Hesap Bilgi Güncelleme.....	14
6.4	Hesap Silme	14
6.5	Standart Gelir/ Gider Kaydı Ekleme	14
6.6	Özel Gelir/ Gider Kaydı Ekleme	14
6.7	İki Tarih Arası Gelir/ Gider Kaydı Silinmesi.....	15
7-	EKLER VE LİNKLER	15
7.1	Örnek c# GMS.NET API Kullanım projesi	15
7.2	Örnek Yedek Dosyası	15

Konu: GMS.NET API Dokümanı

1- AMAÇ

Bu dokümanın amacı, GMS.NET API aracının işlevini ve kullanımını açıklamaktır. Hangi bilgilerin, hangi metodlarla aktarılacağı dokümanda detaylı şekilde açıklanmaktadır.

2- KAPSAM

Bu doküman yazılım geliştirme departmanını ve GMS.NET' e API aracılığıyla bilgi aktarımı yapacak diğer yazılım geliştiren firmaları kapsar.

3- TANIMLAR

3.1 **Zorunlu Alan:** Veri aktarımında mutlaka doldurulması gereken alanlardır. GMS.NET yazılımından giriş yapıldığı takdirde de benzer şekilde bu alanlar doldurulmadan ilgili kayıtlar yapılamaz.

3.2 **Defter- Firma Tipleri:** GMS.NET yazılımında bilançoya tabi firmalar(limited şirket, anonim şirket vb) 1. sınıf firma, işletme defteri tutan firmalar ve serbest meslek mensuplarının firmaları, basit usüle tabi firmalar ise 2. sınıf firmalar olarak kabul edilmektedir.

4- GENEL KAVRAMLAR

4.1 **API Genel Yapısı:** GMS.NET API, c# .net ile geliştirilmiş, GMS.NET yazılımına muhasebe fişi ve fiş içerisinde kullanılan stok, cari, hesap planı vb. bilgileri aktarılmak üzere geliştirilmiş bir araçtır. c#, Vb.net ve benzeri .net tabanlı yazılım geliştirme dilleri aracılığıyla doğal olarak referans eklenerek kullanılacağı gibi aynı zamanda ActiveX teknolojisini desteklemesi sayesinde VB6, asp vb. yazılım geliştirme ortamlarından da ActiveX teknolojisi aracılığıyla kullanılabilir.

4.2 GMS.NET API Aracılığıyla Aktarılacak Bilgiler

GMS.NET API aracılığıyla yeni firma oluşturmak, yeni dönem oluşturmak, hesap planı, stok, cari bilgilerini aktarmak mümkündür. Bu bilgiler aktarıldıktan sonra GMS.NET' e muhasebe fişi oluşturmak ve silmek mümkündür.

4.3 GMS.NET API Kullanımına Uygun Senaryolar

Yazılım geliştirme işiyle uğraşan 3. Parti firmalar çeşitli durumlarda muhasebe ile ilgili kısımlarını kendi yazılımlarında geliştirme yerine bu konuda uzmanlaşmış bir firmaların muhasebe ile ilgili kısımları yönetmesini tercih edebilirler. Örneğin Hastane Bilgi Yönetim Sistemi yazan bir firma, muhasebe ile ilgili kısımları programlarına eklemek yerine GMS.NET' e muhasebe fişi üretebilir ve bu sayede muhasebe ile ilgili kısımların GMS.NET' de takip edilmesini tercih edebilir. Bu durumda HBYS yazılımından fatura kesildiği zaman otomatik olarak bununla ilgili hareketleri muhasebe fişine dönüştürüp API aracılığıyla GMS.NET' e kaydettiğinde muhasebe bölümü sağlıklı bir şekilde bu bilgileri GMS.NET' den takip edebilecektir. Muhasebe bölümü kullanıcısı, detaylı bir şekilde mizan, muavin, yevmiye, kebir vb. dökümleri alabilecek, kısa sürede gereken beyannameleri hazırlayabilecektir. Bu sayede HBYS yazılımı geliştiren firma, muhasebe ile ilgili yenilik ve mevzuatı takip etmek zorunda kalmadan kendi uzmanlaşmış olduğu HBYS konusuna ağırlık verebilecektir.

4.4 GMS.NET API' nin Bulunduğu Dizin

GMS.NET API dosyasının adı MikroKomAPI.dll' dir. GMS.NET çalışma dizininde bulunur. (Varsayılan olarak
"C:\Program Files (x86)\Mikrokom\GMSNET\MikroKomAPI.dll" altındadır.)

Konu: GMS.NET API Dokümanı

4.5 GMS.NET API' nin Bir Projeye Referans Olarak Eklenmesi

GMS.NET API' ye erişim ActiveX teknolojisi sayesinde yapılacak ise ActiveX, GMS.NET' in kurulduğu dizini otomatik olarak konumlandıracaktır. .net tabanlı yazılım geliştirme projelerine referans eklemek istendiği takdirde ise iki türlü yaklaşım söz konusudur. Bunlardan ilki projeye MikroKomAPI.dll dosyasını referans olarak eklemektir. Bu durumda ilgili versiyonda sorunsuz olarak API kullanılabilir ancak bu yaklaşımın bir sakıncası vardır: API' nin versiyonu aylık güncellemelerle yükseltildiği zaman projenin yeniden derlenmesi gerekecektir. Bu sorunu aşmak için, API' yi projeye direkt referans olarak eklemek yerine runtime' da ondan bir instance üretmek daha yerinde bir yaklaşım olacaktır. Bu durumda aylık güncellemeler yapılsa dahi runtime' da API kullanılırken güncel versiyondan yeni instance türetileceğinden herhangi bir sorun oluşmayacaktır. Bu sebeple bu ikinci yöntem tarafımızca önerilmektedir.

4.6 API Kullanımı Örnek Projesi

c# .net ile geliştirilmiş, API' nin kullanımını gösteren örnek projedir. 7.1' deki linkten indirilebilir. Proje içerisinde kullanılan örnek hesap planı ve benzeri datayı içeren dönemlerin yedeği de 7.2' de verilen linkten indirilebilir. Yedek dosyası içerisindeki dönemlerin dönülmesi örnek projede herhangi bir değişiklik yapılmadan kodların takip edilebilmesi için kolaylık sağlayacaktır ancak API' nin anlaşılması için şart olan bir işlem değildir. Bu yedeği dönmek yerine kendi sisteminizdeki firma ve dönemlerle de çalışmanız mümkündür. Bu durumda ilgili kodlarda dönemin ID' si, sistemdeki ID ile eş olarak şekilde ayarlanmalıdır.

4.7 API' nin Başlatılması

c# örnek projesinde MikroKomApi adında bir wrapper class kullanılmıştır. Bu class late binding' i sağlamakla görevlidir. API' nin başlatılması için MikroKomApi.cs dosyasında MikroKomApi constructor metodu bulunmaktadır. Dll' in bulunduğu dizinin belirtilmesi için

```
string dllDizin = @"c:\Program Files (x86)\Mikrokom\GMSNET\MikroKomAPI.dll";  
Assembly asm = Assembly.LoadFile(dllDizin);
```

dllDizin adlı değişkende belirtilen çalışma dizini ayarlanmalıdır. Sonrasında

```
string mesaj = "";  
bool ret = api.InitDll(ref mesaj, ayarlarXmlPath);
```

metodu çağrılarak GMS.NET API dll' i başlatılabilir. InitDll metodundan geriye true değeri döndüğü zaman API başarıyla başlatılmış anlamına gelir. Eğer geriye false değeri dönerse API başarıyla başlatılamamış anlamına gelir ve nedeni mesaj değişkeninde görülebilir. Bu durumda API ile başka işlem yapılmamalıdır. Yapıldığı takdirde sonuç başarısız olacaktır. ayarlarXmlPath, MikroKom ayar dosyasının bulunduğu yerdir.

MikroKom GMS.NET veri dosyaları bir sürücünün MikroKom\GMSNET dizini altında bulunur. Bu sürücü programın kurulu olduğu dizin ile aynı sürücü olmak sorunda değildir. Örneğin program, C:\Program Files (x86)\Mikrokom\GMSNET dizini altında kurulu iken veri dosyaları D:\MikroKom\GMSNET dizini altında duruyor olabilir. Veri dosyalarının hangi sürücüde bulunduğu ile ilgili ayar programın çalışma dizini içinde bulunan veriyeri.txt adlı dosyada saklanmaktadır. Bu yüzden bu dosya içeriği okunup ayar dosyasının tam dizini aşağıdaki gibi üretilmelidir. Bu durumda ayarlarXmlPath, D:\MikroKom\GMSNET\ayarlar.xml olacaktır.

```
string veriyeriDosya = @"C:\Program Files (x86)\Mikrokom\GMSNET\veriyeri.txt";  
string surucu = File.ReadAllText(veriyeriDosya);  
surucu = surucu.Trim().Substring(0, 1);  
string veriyeri = surucu + @":\Mikrokom\GMSNET\ayarlar.xml";
```

Konu: GMS.NET API Dokümanı

4.8 API Çalışma Testi

```
string dbVersiyon = "";  
string mesaj = "";  
bool ret = api.DbVersiyonOgren(ref mesaj, ref dbVersiyon);
```

DbVersiyonOgren metodu çağrılabilir.

GMS.NET API sağlıklı bir şekilde çalışıyor ise geriye true değeri dönecek ve dbVersiyon değişkeninde veritabanı versiyonu dönecektir. Bu durumda API ve veritabanı bağlantısının sorunsuz olarak çalıştığından emin olunabilir.

4.9 API Aracılığıyla GMS.NET' e Giriş Yapma

GMS.NET yazılımı, güvenlik amacıyla girişte bir kullanıcı adı, şifre mekanizması ile programa giriş izni sağlamaktadır. API' nin de benzer şekilde bu girişi sağlaması gerekmektedir. API' nin yapmış olduğu her işlem GMS.NET tarafında gerçek bir kullanıcı aracılığıyla yapılıyor olmalıdır. Sonrasında yapılan işlemlerde ilgili kullanıcının belirli işlemleri yapmaya izni olup olmadığı kontrol edilecektir. Girişi sağlamak için Login metodu kullanılır. Login metodu öncesinde API başlatılmış olmalıdır.

```
string mesaj = "";  
string kullanıcı = "emrah";  
string sifre = "emrah";  
bool ret = api.Login(ref mesaj, kullanıcı, sifre);
```

metoddan geriye true döndü ise giriş başarıyla sağlanmıştır. false döndü ise mesaj değişkeninde hatanın nedeni görülebilir.

Şifresi değişmesi muhtemel belirli bir kullanıcı adı ile sisteme girmek yerine işlemi gms süper kullanıcısı aracılığıyla yapmak için gms süper kullanıcısının id' si belirtilerek sisteme şifresiz giriş sağlanabilir. Aşağıda belirtilen ID, gms süper kullanıcısının tüm sistemlerde değişmeyen ID' si olup her zaman sabittir.

```
string userID = "58BAE36C-804A-4DE5-94FC-AC4D90FC23A2";  
bool ret = api.LoginByKullaniciID(ref mesaj, userID);
```

4.10 İzin Verilen Firma Listesi

API aracılığıyla, kullanıcı girişi sağladıktan sonra, işlemlerin hangi dönemde yapılacağı seçilmiş olmalıdır. Entegrasyon yapılmak istenen 3. Parti yazılımda, GMS.NET' in birden fazla firma ve dönemine işlem yapılmak isteniyor olabilir. Bu nedenle, entegrasyonun hangi döneme yapılacağını belirleyebilmek adına 3. Parti Yazılıma, aktarım yapılmak istenen GMS.NET firma ve dönemini seçiniz şeklinde bir diyalog konabilir. Burada, ilgili firmaları listesini listeleyebilmek için:

```
DataSet ds = new DataSet();  
bool ret = api.IzinVerilenFirmaListesiGetir(ref mesaj, ref ds);
```

metodu kullanılmalıdır. Bu durumda sisteme giriş yapılan emrah kullanıcısının erişim izni olan firmaların listesi ds datasetindeki table 0' a doldurulacaktır.

4.11 İzin Verilen Dönem Listesi

API aracılığıyla, kullanıcı girişi sağladıktan sonra, işlemlerin hangi dönemde yapılacağı seçilmiş olmalıdır. Entegrasyon yapılmak istenen 3. Parti yazılımda, GMS.NET' in birden fazla firma ve dönemine işlem yapılmak isteniyor olabilir. Bu nedenle, entegrasyonun hangi döneme yapılacağını belirleyebilmek adına 3. Parti Yazılıma, aktarım yapılmak istenen GMS.NET dönemini seçiniz şeklinde bir diyalog konabilir. Burada, ilgili dönemlerini listesini listeleyebilmek için:

Konu: GMS.NET API Dokümanı

```
DataSet ds = new DataSet();  
bool ret = api.IzinVerilenDonemListesiGetir(ref mesaj, ref ds);
```

metodu kullanılmalıdır. Bu durumda sisteme giriş yapılan emrah kullanıcısının erişim izni olan dönemlerin listesi ds datasetindeki table 0' a doldurulacaktır. Listede dönen dönemlerin hangi firmalara ait olduğu FIRMA_ID sütununda tutulmaktadır. Bu sayede son kullanıcıya istenildiği taktirde 4.9' da seçilmiş olan firmaya ait dönemleri süzerek listelemek mümkün olacaktır.

4.12 Dönem Seçimi

API aracılığıyla yapılan, dönem bazındaki işlemlerde, işlemlerin hangi dönemde yapılacağını belirtmek adına dönem seçimi yapılmalıdır. Bu işlem, GMS.NET' e girişte, kullanıcı adı ve şifre girildikten sonra yapılan dönem seçimine karşılık gelmektedir.

```
string ornekDonemIDBilanco2012 = "D2643086-7048-4844-96A8-E23EF2531E22";  
string donemID = ornekDonemIDBilanco2012;  
bool ret = api.DonemSec(ref mesaj, donemID);
```

donemID değişkeninde belirtilen dönemin seçilmesi için DonemSec metodu çağrılır. Dönem seçimi başarılı ise true değeri dönecektir. Dönem ve firma ile ilgili bilgiler GMSNETORTAK..T_BIL_FIRMA ve GMSNETORTAK..T_BIL_DONEM tablolarında tutulmaktadır.

4.13 Durum Öğrenme

Program akışında, herhangi bir anda, API' nın bağlanmış olduğu dönem, ilgili dönemin veritabanı adı, çalışılan veritabanı sunucusunun yerel sürücüsü, yedek dizini vb. bilgiler öğrenilmek istendiğinde kullanılır.

```
ApiGlobal d = new ApiGlobal();  
bool ret = api.GetDurum(ref mesaj, ref d);  
string toplamMesaj = "";  
toplamMesaj += "ConnectionString : " + d.ConnectionString + Environment.NewLine;  
//SecilenDonem_DEFTER_TIP_KOD 1 : bilanço 2 : işletme 3: serbest meslek 4: deftere tabi değil.  
toplamMesaj += "SecilenDonem_DEFTER_TIP_KOD : " + d.SecilenDonem_DEFTER_TIP_KOD.ToString() + Environment.NewLine;  
toplamMesaj += "SecilenDonem_DONEM_AD : " + d.SecilenDonem_DONEM_AD + Environment.NewLine;  
toplamMesaj += "SecilenDonem_DONEM_DATABASE_ADI : " + d.SecilenDonem_DONEM_DATABASE_ADI + Environment.NewLine;  
toplamMesaj += "SecilenDonem_DONEM_ID : " + d.SecilenDonem_DONEM_ID + Environment.NewLine;  
toplamMesaj += "SecilenDonem_DONEM_YIL : " + d.SecilenDonem_DONEM_YIL.ToString() + Environment.NewLine;  
toplamMesaj += "SecilenDonem_FIRMA_AD : " + d.SecilenDonem_FIRMA_AD + Environment.NewLine;  
toplamMesaj += "SecilenDonem_FIRMA_DATABASE_ADI : " + d.SecilenDonem_FIRMA_DATABASE_ADI + Environment.NewLine;  
toplamMesaj += "SecilenDonem_FIRMA_ID : " + d.SecilenDonem_FIRMA_ID + Environment.NewLine;  
toplamMesaj += "SecilenDonem_FIRMA_TABLO_KODU : " + d.SecilenDonem_FIRMA_TABLO_KODU + Environment.NewLine;  
toplamMesaj += "SecilenKullanici_ID : " + d.SecilenKullanici_ID + Environment.NewLine;  
toplamMesaj += "SecilenKullanici_KULLANICI_AD : " + d.SecilenKullanici_KULLANICI_AD + Environment.NewLine;  
toplamMesaj += "SecilenKullanici_TAM_AD : " + d.SecilenKullanici_TAM_AD + Environment.NewLine;  
toplamMesaj += "SunucuDataDrive : " + d.SunucuDataDrive + Environment.NewLine;  
toplamMesaj += "XmlFileName : " + d.XmlFileName + Environment.NewLine;  
toplamMesaj += "YedekDizin : " + d.YedekDizin + Environment.NewLine;  
MessageBox.Show(toplamMesaj);
```

4.14 Defter Tip Kodları

GMS.NET yazılımında durum öğrenildiği zaman görülebilen defter tip kodları, GMSNETORTAK.T_KOD_DEFTER_TIP tablosunda görülebilir. Değerleri:

- 1- BİLANÇO
- 2- İŞLETME
- 3- SERBEST MESLEK
- 4- DEFTERE TABİ DEĞİL
- 5- BASİT USUL

Konu: GMS.NET API Dokümanı

4.15 Belge Tiplerinin Listesi

GMS.NET yazılımında, fiş üretilmesi sırasında, fiş satırlarında, belge türleri belirtilmek istenebilir. Burada belirtilecek olan belge tipleri ID ile belirtilmek durumundadır. Bu nedenle kullanıcıya, üretilecek fiş satırlarında hangi belge türünün kullanılacağı sorulmak istenebilir. Fiş satırlarında belge türü kullanımı zorunlu değildir. BelgeTurleriniGetir metodu ile ilgili dönemdeki tanımlı olan belge türlerinin listesi getirilebilir.

```
DataSet ds = new DataSet();  
bool ret = api.BelgeTurleriniGetir(ref mesaj, ref ds);
```

Dönen dataset' deki ID sütunu belge türü ID' sini, KOD alanı belge türü kodunu, ACIKLAMA sütunu ise belge türü açıklamasını göstermektedir. Belge tiplerinin listesi 1. sınıf ve 2. sınıf firmalar için aynı şekilde çalışmaktadır.

5- 1. SINIF FİRMALARLA İLGİLİ İŞLEMLER**5.1 Hesap Planı Listesi**

GMS.NET yazılımında, fiş üretilmesi sırasında, fiş satırlarında, hesap kodu belirtilmelidir. Fiş satırlarında hesap kodu kullanımı zorunludur. HesapPlanıGetir metodu ile ilgili dönemdeki tanımlı olan hesap listesi getirilebilir.

```
DataSet dsHesapPlan = new DataSet();  
bool ret = api.HesapPlanıGetir(ref mesaj, ref dsHesapPlan);
```

5.2 Yeni Hesap Ekleme

Hesap planına yeni bir hesap eklemek için kullanılır. Eklenecek olan yeni hesabın sistem kurallarına uygun olması ile ilgili testler otomatik olarak yapılmaktadır. Örneğin: Sistemde 100 hesap kodu var iken, 100.03 yok iken 100.03.1001 hesap kodu açılmaya çalışıldığında metod, yeni eklenmek istenen hesabın üst hesap kodu yok şeklinde hata döndürecektir. Hesap kodu en fazla 24 karakter uzunluğunda string, hesap adı 40 karakter uzunluğunda string, birim 5 karakter uzunluğunda string, stok kodu 64 uzunluğunda string olabilir. Tck/Vkn ise 10 veya 11 karakter uzunluğunda vergi kimlik numarası veya TC kimlik numarası string olabilir.

```
string hesapKodu = "100.03"; //24 karakter  
string hesapAdi = "Merkez Kasa USD"; //40 karakter  
string stokKod = "";  
string birim = "";  
string tckVkn = ""; //10 veya 11 karakter  
bool ret = api.HesapEkle(ref mesaj, hesapKodu, hesapAdi, birim, stokKod, tckVkn);
```

5.3 Hesap Bilgi Güncelleme

İstenen hesabın çeşitli bilgilerinin güncellenmesi için kullanılır. Verilen koddaki hesap bulunur, verilen bilgilerle güncellenir.

```
string hesapKodu = "100.02"; //24 karakter  
string hesapAdi = "Merkez Kasa USD"; //40 karakter  
string stokKod = "";  
string birim = "";  
string tckVkn = ""; //10 veya 11 karakter  
bool ret = api.HesapGuncelle(ref mesaj, hesapKodu, hesapAdi, birim, stokKod, tckVkn);
```


Konu: GMS.NET API Dokümanı

5.4 Hesap Silme

İstlenen hesabın hesap planından silinmesi için kullanılır. Silinmek istenen hesabın alt hesapları bulunmamalıdır ve hesap fiş satırları girişinde kullanılmamış olmalıdır. Aksi takdirde hesap silinemeyecektir.

```
string hesapKodu = "100.03"; //24 karakter  
bool ret = api.HesapSil(ref mesaj, hesapKodu);
```

5.5 Hesap Fiş Girişinde Kullanılabilir mi?

GMS.NET fiş girişinde, hesabın fiş satırlarında kullanılabilmesi için, hesabın, hesap planında tanımlı olması ve hesabın detay hesap olması gereklidir. Belirli bir hesabın bu kriterlere uyup uymadığını kontrol etmek için bu metod kullanılır.

```
string incelenecekHesap = "100.01";  
bool kullanılabilir = false;  
bool ret = api.HesapKullanilabilir(ref mesaj, ref kullanılabilir, incelenecekHesap);
```

Eğer hesap, fiş satırlarında kullanılabilir bir hesap ise kullanılabilir değişkeni true olarak geri dönecektir.

5.6 Hesap Detayı

Belirli bir hesabın detay bilgilerini öğrenmek için kullanılır.

```
string incelenecekHesap = "100.01";  
DataSet dsHesapDetay = new DataSet();  
bool ret = api.GetHesapDetay(ref mesaj, ref dsHesapDetay, incelenecekHesap);
```

Eğer hesap mevcut ise dsHesapDetay'ın Table 0'ında ilgili hesabın tüm alanları görülebilecektir.

5.7 Stok Kodu Listesi

GMS.NET yazılımında, fiş üretilmesi sırasında, fiş satırlarında, stok kodu belirtilebilir. Stok kodu belirtilmesi zorunlu değildir. StokKodGetir metodu ile ilgili dönemdeki tanımlı olan stok kodu listesi getirilebilir.

```
DataSet dsStok = new DataSet();  
bool ret = api.StokKodGetir(ref mesaj, ref dsStok);
```

5.8 Yeni Stok Ekleme

Yeni bir stok eklemek için kullanılır. Alan kısıtlamaları ve lookup alanların alabileceği değerler ile ilgili olarak örnek proje incelenebilir.

```
bool ret = api.StokEkle (ref mesaj, stokID, stokKodu, stokAciklamasi, birim, birimEFaturaEsdegeri,  
satisFiyati, satisKDVOrani, satisTevkifatTurKodu, satisTevkifatOrani, satisStopajTurKodu,  
satisStopajOrani, alisKDVOrani, alisTevkifatTurKodu, alisTevkifatOrani, alisStopajTurKodu,  
alisStopajOrani, hesapKoduAlis, hesapKoduAlisIade, hesapKoduAlisIndirim, hesapKoduYurticiSatis,  
hesapKoduYurtIciSatisTevkifatli, hesapKoduYurtdisiSatis, hesapKoduSatisIade,  
hesapKoduSatisIndirim, hesapKoduSatilanMalinMaliyeti);
```

Konu: GMS.NET API Dokümanı

5.9 Stok Bilgi Güncelleme

İstenen stoğun çeşitli bilgilerinin güncellenmesi için kullanılır. Verilen koddaki stok bulunur, verilen bilgilerle güncellenir. Alan kısıtlamaları ve lookup alanların alabileceği değerler ile ilgili olarak örnek proje incelenebilir.

```
bool ret = api.StokGuncelle(ref mesaj, stokKodu, stokAciklamasi, birim, birimEFaturaEsdegeri, satisFiyati, satisKDVOrani, satisTevkifatTurKodu, satisTevkifatOrani, satisStopajTurKodu, satisStopajOrani, alisKDVOrani, alisTevkifatTurKodu, alisTevkifatOrani, alisStopajTurKodu, alisStopajOrani, hesapKoduAlis, hesapKoduAlisIade, hesapKoduAlisIndirim, hesapKoduYurticiSatis, hesapKoduYurtIciSatisTevkifatli, hesapKoduYurtdisiSatis, hesapKoduSatisIade, hesapKoduSatisIndirim, hesapKoduSatilanMalinMaliyeti);
```

5.10 Stok Silme

İstenen stoğun silinmesi için kullanılır.

```
string stokKodu = "SK03"; //64 karakter string  
ret = api.StokSil(ref mesaj, stokKodu);
```

5.11 Cari Listesi

GMS.NET yazılımında, fiş üretilmesi sırasında, fiş satırlarında, ilgili cariye ait TCK/VKN belirtilebilir. Belirtilmesi zorunlu değildir. CariListeGetir metodu ile ilgili tanımlı olan cari kodu listesi getirilebilir.

GMS.NET Yazılımında iki tür cari vardır. Bunlardan birincisi muhasebe carisidir. Muhasebe carisinde ilişkiler TCK/VKN üzerinden kurulur. Bu yüzden ilgili TCK/VKN' nin sistemde tekil olması gereklidir. Diğer tür cari ise fatura carisidir. Fatura carisinde ilişkiler cari ID üzerinden kurulur. Bu yüzden fatura carilerinde birden fazla VKN' ye ait cari bulunabilir. GMS.NET API temel olarak muhasebe kısmı ile ilgili olduğundan muhasebe carileri ile ilgilenmektedir.

Muhasebe carileri ilgili döneme kaydedilir ancak tüm firmalarda kullanılabilecek olan carilerin tekrar tekrar kaydedilmesi bir zaman kaybıdır. (Örn: Türk Telekom Carisi) Bu durumun önüne geçebilmek için muhasebe carilerini ortak veritabanına da kaydetmek mümkündür. Bu yüzden gerek kayıt yapılırken gerekse mevcut kayıtlar listelenirken hangi veritabanındaki kayıtlar ile ilgilenildiği belirtilmek zorundadır. Burada:

- 0: Sadece Dönemdeki cariler ,
- 1: Sadece Ortaktaki cariler,
- 2: Hem dönem hem ortaktaki cariler anlamına gelmektedir.

```
DataSet dsCari = new DataSet();  
int tip = 0;  
bool ret = api.CarilisteGetir(ref mesaj, ref dsCari, tip);
```

5.12 Yeni Cari Ekleme

Yeni bir cari eklemek için kullanılır. Alan kısıtlamaları ve lookup alanların alabileceği değerler ile ilgili olarak örnek proje incelenebilir.

```
bool ret = api.CariEkle(ref mesaj, cariID, tip, tckVkn, soyad, ad, adres1, adres2, adres3, adres4, ilce, il, postaKodu, adresNo, bagkurKesintisi, ulkeKodu, vergiDaireKod, vergiDaireAd, telKod, telNo, faxKod, faxNo, ePosta, web, yetkili, yetkiliGorev, hitapOnek, ekAciklamalar);
```

Konu: GMS.NET API Dokümanı

5.13 Cari Bilgi Güncelleme

İstenen carinin çeşitli bilgilerinin güncellenmesi için kullanılır. Verilen TCK/VKN' li cari bulunur, verilen bilgilerle güncellenir. Alan kısıtlamaları ve lookup alanların alabileceği değerler ile ilgili olarak örnek proje incelenebilir.

```
bool ret = api.CariGuncelle(ref mesaj, tip, tckVkn, soyad, ad, adres1, adres2, adres3, adres4, ilce, il, postaKodu, adresNo, bagkurKesintisi, ulkeKodu, vergiDaireKod, vergiDaireAd, telKod, telNo, faxKod, faxNo, ePosta, web, yetkili, yetkiliGorev, hitapOnnek, ekAciklamalar);
```

5.14 Cari Silme

İstenen carinin silinmesi için kullanılır. Muhasebe carileri için bir veritabanı ilişkisi bulunmadığından, ilişkiler TCK/VKN üzerinden cari mevcut ise kurulduğundan, silmek istediğiniz cari, fş satırlarında ya da herhangi bir yerde kullanılmış olsa dahil silinecektir.

```
string tckVkn = "6210028566";  
int tip = 0;  
bool ret = api.CariSil(ref mesaj, tckVkn, tip);
```

5.15 DeveloperID Kavramı

3. Parti programdan GMS.NET' e fiş kaydı senaryosunda sorun çıktığı zaman belirli tarih aralığında hatalı aktarılan fişlerin silinip yeniden kaydedilmesi ihtiyacı doğabilir ancak bu aralıkta GMS.NET kullanıcısı 3. Parti programdan olmayıp kendi iç işlerini yürütmek için de çeşitli fişler kaydemiş olabilir. Bu durumda iki tarih aralığındaki tüm fişleri silmek bu tür fişlerin kaybolmasına neden olacaktır. Bu sorunu aşabilmek için DeveloperID çözümüne gidilmiştir. DeveloperID, 3. Parti firmayı temsil eden 64 karakter uzunluğunda bir stringdir. (Örn: FLAWMORE_YAZILIM) Bu değer fişler kaydedilirken veritabanına yazılacak, sonrasında silme vb. işlemler yapılırken de bu alan dikkate alınacak, standart fişler bu sayede zarar görmeyecektir. Bu yüzden 3. Parti firmalar yaptıkları fiş işlemlerinde DeveloperID' yi kullanmalıdır.

5.16 Standart Fiş, Özel Fiş ve Özel Alan Kavramları

GMS.NET yazılımından, kullanıcının kaydedebileceği fiş, standart fiş olarak kabul edilmektedir. Bir özelliği yoktur. Bu fişten farklı olarak GMS.NET' den kaydedilen standart fişlere DeveloperID yazabilmek mümkündür. GMS.NET API dokümanlarında özel fiş olarak bahsedilen fiş ise, fişin üzerinde DeveloperID ve isteğe bağlı özel alan içeren fiştir. Örneğin: 3. Parti FLAWMORE_YAZILIM, bir İnsan Kaynakları yazılımı geliştiriyor olsun. Üretmiş oldukları bordrolarla ilgili olarak her ay sadece bir adet tahakkuk fişi kesiyor olsun. 2017 Mayıs ayı ile ilgili bir tahakkuk fişi kesildiğinde bu aya ait fişin sistemde olup olmadığı kontrol edilebilmelidir. Aksi takdirde mükerrer fiş üretilmiş olacaktır. Bunu kontrol edebilmek için Özel Alan kullanılmaktadır. Flawmore yazılım, DeveloperID olarak FLAWMORE_YAZILIM, özel alan olarak ise kendi belirlediği bir formatta, 64 karakter string yazabilecektir. Örnek: BRD_THK_2017_05. Daha sonrasında API' deki metodlar aracılığıyla BRD_THK_2017_05 özel alanlı fiş var mı diye kontrol edilecek, var ise silinecektir. Bu sayede mükerrer fiş sorununun önüne geçilebilecektir.

5.17 Fiş Kesinleştirme Tarihi Kavramı

GMS.NET yazılımında, muhasebe kullanıcıları, belirli zamanlarda (Örn: Geçici beyannameleri tamamlayıp gönderdikten sonra) fişlerin değiştirilmesini engellemek isteyebilirler. Bu durumda GMS.NET yazılımından Fiş Kesinleştirme tarihi girerler. Örneğin, fiş kesinleştirme tarihi olarak 31.03.2017 belirlendiğinde, 31.03.2017 ve öncesi tarihli fişleri silmek, değiştirmek, bu tarih öncesine fiş eklemek mümkün olmayacaktır. GMS.NET API aracılığıyla bu değeri öğrenmek mümkündür. 1. Sınıf firmalarda FisKesinlestirmeTarihiniGetir metodu ile bu değer öğrenilebilmekte iken 2. Sınıf firmalarda bu konu ile ilgili aşağıdaki 3 metoddan herhangi birisi kullanılabilir.

Konu: GMS.NET API Dokümanı

1. Sınıf firmalarda kesinleştirme tarihinin öğrenilmesi:

```
DateTime fkt = new DateTime(1, 1, 1);  
bool ret = api.FisKesinlestirmeTarihiniGetir(ref mesaj, ref fkt);
```

2. Sınıf firmalarda fiş kesinleştirme tarihi gelir ve gider kesinleştirme tarihi olarak ayrı ayrı öğrenebileceği gibi ikisi birden gerektiğinde tek metodla öğrenilebilir.

```
DateTime fkt = new DateTime(1, 1, 1);  
bool ret = api.FisKesinlestirmeTarihiniGetirGider(ref mesaj, ref fkt);
```

```
DateTime fkt = new DateTime(1, 1, 1);  
ret = api.FisKesinlestirmeTarihiniGetirGelir(ref mesaj, ref fkt);
```

```
DateTime fktGelir = new DateTime(1, 1, 1);  
DateTime fktGider = new DateTime(1, 1, 1);  
ret = api.FisKesinlestirmeTarihiniGetirGelirGider(ref mesaj, ref fktGelir, ref fktGider);
```

5.18 Standart Fiş Ekleme

Fiş başlık kısmını ilgilendiren tarih, fiş tipi, DeveloperID, fiş başlık açıklaması alanları belirlenir. Fiş satırları bir datasete doldurulur. FisKaydet metodu ile standart fiş eklenir. Alan ve lookup değerleri için örnek projeye bakılabilir. Ekleniecek fişin fiş kesinleştirme tarihinden önce olmaması gereklidir.

```
bool ret = api.FisKaydet(ref mesaj, ref fisNo, fisBaslikID, fisAciklama, fisTipi, fisTarih, developerID, dsFisSatir);
```

5.19 Özel Fiş Ekleme

Fiş başlık kısmını ilgilendiren tarih, fiş tipi, DeveloperID, fiş başlık açıklaması alanları belirlenir. Fiş başlık ve fiş satırları bazında özel alan bilgisi atanmalıdır. Fiş satırları bir datasete doldurulur. OzelFisKaydet metodu ile özel fiş eklenir. Alan ve lookup değerleri için örnek projeye bakılabilir. Ekleniecek fişin fiş kesinleştirme tarihinden önce olmaması gereklidir.

```
bool ret = ret = api.OzelFisKaydet(ref mesaj, ref fisNo, fisBaslikID, fisAciklama, fisTipi, fisTarih, ozelAlan, developerID, dsFisSatir);
```

5.20 Özel Fiş Kontrol

Belirli bir özel alana sahip olan fişin bulunup bulunmadığı kontrol edilmek istenebilir. Bu durumda OzelFisVarMiKontrol metodu çağrılır. Sorgulanmak istenen ozelAlan 64 uzunluğunda string değişkendir. Örnek: BRD_THK_2017_05. Eğer bu değere sahip fiş satırları sistemde mevcut ise fisZatenVar bool değişkeni true dönecektir. Bu durumda fiş yeniden üretilmeyebilir veya fiş silinip yeniden üretilebilir.

```
bool ret = api.OzelFisVarMiKontrol(ref mesaj, ref fisZatenVar, ref olanFisID, ref olanFisTarih, ref olanFisNo, ref olanFisVersiyon, ozelAlan, developerID);
```

5.21 Özel Alan Bilgili Fiş Satırlarının Silinmesi

Belirli bir özel alana sahip olan fişin ve/veya fiş satırlarının silinmesi için kullanılır. Fiş satırlarının tamamı silinmiş ise fiş de otomatik olarak silinecektir ancak fiş, GMS.NET tarafında başla fişlerle birleştirilmiş ise, sadece belirtilen özel alana sahip fiş satırları silinecektir. Bu durumda fisDuruyor değişkeni true dönecektir. Bu durum az rastlanan bir durumdur ve bir sorun teşkil etmemektedir. Örnek projede metodun uygulaması görülebilir.

```
bool ret = api.EntegreFisSatirSil(ref mesaj, ref fisDuruyor, silinecekFisID, ozelAlan, developerID);
```

Konu: GMS.NET API Dokümanı

5.22 İki Tarih Arası Fiş Silinmesi

3. Parti programdan GMS.NET' e fiş kaydı senaryosunda sorun çıktığı zaman belirli tarih aralığında hatalı aktarılan fişlerin silinip yeniden kaydedilmesi gerekliliğinde kullanılır. Örneğin Ocak ayında Flawmore yazılım tarafından aktarılan tüm fişlerin silinebilmesi için değişkenler aşağıdaki gibi ayarlanmalı ve FisSilByDeveloperID metodu çağrılmalıdır.

```
DateTime bslTarih = new DateTime(2017, 1, 1);  
DateTime bitTarih = new DateTime(2017, 1, 31);  
string developerID = "FLAWMORE_YAZILIM";  
bool ret = api.FisSilByDeveloperID(ref mesaj, developerID, bslTarih, bitTarih);
```

Konu: GMS.NET API Dokümanı

6- 2. SINIF FİRMALARLA İLGİLİ İŞLEMLER**6.1 OKH Listesi**

Ortalama kar haddi listesini getirir.

```
DataSet dsOkh = new DataSet();  
bool ret = api.OkhKodGetir(ref mesaj, ref dsOkh);
```

6.2 Yeni Hesap Ekleme

Hesap planına yeni bir hesap eklemek için kullanılır. Eklenecek olan yeni hesabın sistem kurallarına uygun olması ile ilgili testler otomatik olarak yapılmaktadır.

```
string hesapKodu = "02"; //16 karakter  
string hesapAdi = "B Malı"; //40 karakter  
decimal kdVOran = -1; // -1 belirtilmemiş 0 : 0 , 1, 8 , 18 vb yaz  
string birim = "Adet";  
int hesapTipi = (int)GmsNetImsHesapTip.Serbest;  
string okhKodu = "";  
ret = api.HesapEkleSinif2(ref mesaj, hesapKodu, hesapAdi, kdVOran, hesapTipi, birim, okhKodu);
```

6.3 Hesap Bilgi Güncelleme

İstenen hesabın çeşitli bilgilerinin güncellenmesi için kullanılır. Verilen koddaki hesap bulunur, verilen bilgilerle güncellenir.

```
string hesapKodu = "02"; //16 karakter  
string hesapAdi = "B Malı Yeni"; //40 karakter  
decimal kdVOran = -1; // -1 belirtilmemiş 0, 1, 8 , 18 vb oranlar  
string birim = "Kg";  
int hesapTipi = (int)GmsNetImsHesapTip.Serbest;  
string okhKodu = "";  
ret = api.HesapGuncelleSinif2(ref mesaj, hesapKodu, hesapAdi, kdVOran, hesapTipi, birim, okhKodu);
```

6.4 Hesap Silme

İstenen hesabın hesap planından silinmesi için kullanılır.

```
string hesapKodu = "02"; //16 karakter  
bool ret = api.HesapSilSinif2(ref mesaj, hesapKodu);
```

6.5 Standart Gelir/ Gider Kaydı Ekleme

2. Sınıf firmalarda kayıtlar satır bazındadır. CreateDsFisSatirSinif2 metodu kullanılarak bir gelir/gider kayıt satırı oluşturulur, bilgiler doldurulur, FisKaydetSinif2 metodu kullanılarak kaydedilir. Alan ve lookup değerleri için örnek projeye bakılabilir. Eklenecek gelir/ gider kaydının kesinleştirme tarihinden önce olmaması gereklidir.

```
bool ret = api.FisKaydetSinif2(ref mesaj, developerID, dsGG);
```

6.6 Özel Gelir/ Gider Kaydı Ekleme

Standart gelir/ gider kaydına ek olarak özel alan bilgisi ayarlanır. 1. Sınıf firmalardakine benzer şekilde OzelFisVarMiKontrolSinif2 metodu ile belirli özel alan bilgisi içeren gelir/ gider kaydının olup olmadığı kontrol edilebilir. Eğer böyle bir satır var ise FisSilSinif2 metodu ile ilgili gelir/ gider kaydı silinebilir.

```
bool ret = api.OzelFisKaydetSinif2(ref mesaj, ozelAlan, developerID, dsGG);
```

Konu: GMS.NET API Dokümanı

6.7 İki Tarih Arası Gelir/ Gider Kaydı Silinmesi

3. Parti programdan GMS.NET' e gelir/ gider kaydı senaryosunda sorun çıktığı zaman belirli tarih aralığında hatalı aktarılan gelir/ gider kayıtlarının silinip yeniden kaydedilmesi gerektiğinde kullanılır. Örneğin Ocak ayında Flawmore yazılım tarafından aktarılan tüm gelir/ gider kayıtlarının silinebilmesi için değişkenler aşağıdaki gibi ayarlanmalı ve FisSilByDeveloperIDSınıf2 metodu çağrılmalıdır.

```
DateTime bslTarih = new DateTime(2017, 1, 1);  
DateTime bitTarih = new DateTime(2017, 1, 31);  
bool ret = api.FisSilByDeveloperIDSınıf2(ref mesaj, developerID, bslTarih, bitTarih);
```

7- EKLER VE LİNKLER**7.1 Örnek c# GMS.NET API Kullanım projesi**

<http://www.mikrokom.com/docs/aktarim/MikroKomAPITest.zip> adresinden indirilebilir.

7.2 Örnek Yedek Dosyası

7.1' de belirtilen örnek projenin kullanmış olduğu firma ve dönemleri içinde bulunduran örnek yedek dosyasıdır.
http://www.mikrokom.com/docs/aktarim/Yedek_2012_05_15_092718.zip adresinden indirilebilir. Yedek dosyalarının yüklenebilmesi için GMS.NET' in lisanslı olması gerekmektedir.