



# Opencart Açık Kaynak Kodlu E- Ticaret Web Yazılımının Yerel Bir Muhasebe Programı Olan CetaSoft ve Trendyol Pazaryeri ile Api Entegrasyonu

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Gökhan Gökmen Korkmaz

Proje Danışmanı: Prof.Dr. Merih Palandöken

Ocak 2024

# Opencart Açık Kaynak Kodlu E-Ticaret Web Yazılımının Yerel Bir Muhasebe Programı Olan CetaSoft ve Trendyol Pazaryeri ile Api Entegrasyonu

## ÖZ

Bu makalede, açık kaynaklı e-ticaret web yazılımı OpenCart'ın, yerel muhasebe programı Cetasoft ve popüler pazaryeri Trendyol ile gerçekleştirdiği API entegrasyonuna odaklandık. OpenCart'ın esnek yapısı, Cetasoft'un ön muhasebe çözümü ve Trendyol'un geniş müşteri kitlesi, işletmelere birleşik bir e-ticaret deneyimi sunabilir. Bu entegrasyon sayesinde, finansal süreçlerin düzenli takibi, pazaryeri yönetimi ve müşteri ilişkilerinin güçlendirilmesi mümkün olabilir. Sonuç olarak, bu entegrasyonlar işletme özelinde daha etkili bir rekabet avantajı sağlayabilir ve e-ticaret deneyimini daha entegre hale getirir.

Anahtar Sözcükler: Muhasebe, Entegrasyon, C#, Soap, PHP, Api, RestApi, json, Trendyol, Cetasoft

# Integrating CetaSoft, a local accounting program, and Trendyol Marketplace with API into OpenCart, an open-source e-commerce web software.

## Abstract

In this article, we focused on the API integration of the open-source e-commerce web software OpenCart with the local accounting program Cetasoft and the popular marketplace Trendyol. The flexible structure of OpenCart, Cetasoft's pre-accounting solution, and Trendyol's extensive customer base can provide businesses with a unified e-commerce experience. Through this integration, regular monitoring of financial processes, marketplace management, and strengthening customer relationships can be achieved. Consequently, these integrations can provide a more effective competitive advantage at the business level and make the e-commerce experience more integrated.

Keywords: Accounting, Integration, C#, Soap, PHP, Api, RestApi, json, Trendyol, Cetasoft

*Ođlum Bora'ya ithafen...*

# Teşekkür

Proje çalışmamda bana verdikleri destekten ötürü Ege El Sanatları Kuy. San. Tic. Ltd. Şti. muhasebe çalışanlarına teşekkür ederim.

# İçindekiler

Opencart Açık Kaynak Kodlu E-Ticaret Web Yazılımının Yerel Bir Muhasebe Programı Olan CetaSoft ve Trendyol Pazaryeri ile Api Entegrasyonu .....	i
Opencart Açık Kaynak Kodlu E-Ticaret Web Yazılımının Yerel Bir Muhasebe Programı Olan CetaSoft ve Trendyol Pazaryeri ile Api Entegrasyonu .....	ii
ÖZ .....	ii
Integrating CetaSoft, a local accounting program, and Trendyol Marketplace with API into OpenCart, an open-source e-commerce web software. ....	iii
Abstract .....	iii
Teşekkür.....	v
Bölüm 1 .....	1
1. Giriş .....	1
2. Cetasoft Web Servisleri .....	2
2.1 Web Servis Sınıfı (wsOfis): .....	2
2.2 Veritabanı İşlemleri .....	2
2.3 Müşteri Yönetimi .....	3
2.3.1 AddCustomer .....	3
2.3.2 EditCustomer .....	4
2.3.3 GetCustomer .....	5
2.4 Sipariş İşlemleri.....	6
2.4.1 AddOrder.....	6
2.4.2 AddOrderedItem .....	7
2.4.3 GetOrderStatus .....	8
2.5 Ürün Yönetimi.....	8
2.5.1 GetAllProducts .....	9
2.5.2 GetProductsByCategory.....	10

2.5.3	GetProductsInStock .....	11
2.5.4	GetProductByStockCode .....	12
2.6	Güvenlik Kontrolü.....	12
2.7	Diğer Fonksiyonlar .....	12
3	Trendyol Servisleri .....	14
3.1	updateProducts .....	14
3.2	updatePriceAndInventory .....	14
3.3	getShipmentPackages .....	15
4	Web Sitesi Servisleri (Opencart) .....	16
4.1	Trendyol .....	16
4.1.1	trendyol/updateProducts.....	16
4.1.2	trendyol/getOrderPackages .....	16
4.1.3	trendyol/createOrder .....	16
4.2	Cetasoft.....	17
4.2.1	cetasoft/getAllProducts .....	17
4.2.2	cetasoft/addOrder .....	17
4.2.3	cetasoft/addOrderedItem .....	17
4.2.4	cetasoft/getCustomerId .....	17
5	Akış Diyagramı.....	18
	Kaynaklar .....	19
	SOAP Servisleri .....	19
	RestApi Servisleri.....	19
	PhpCURL .....	19
	EKLER .....	20
	CetaSoft Web Servis Kodları .....	21

Opencart Web Servis Kodları .....	48
Cetasoft.....	48
Controller.....	48
Model.....	52
Trendyol .....	53
Controller.....	53
Model.....	76



# Bölüm 1

## 1. Giriş

E-ticaret dünyası her geçen gün gelişirken, işletmeler rekabet avantajını sürdürebilmek için çeşitli entegrasyonları değerlendirmektedir. OpenCart, açık kaynaklı yapısı ve geniş eklenti desteği ile öne çıkan bir e-ticaret web yazılımıdır. Bu makalede, OpenCart'ın yerel muhasebe programı Cetasoft ve popüler pazaryeri Trendyol ile gerçekleştireceğimiz API entegrasyonu üzerinde duracağız.

OpenCart, kullanıcı dostu arayüzü, esnek yapısı ve geniş eklenti desteği ile birçok işletmenin tercihi haline gelmiştir. Açık kaynak kodlu olması, geliştiricilerin ihtiyaçlara özel özelleştirmeler yapmasını kolaylaştırırken, kullanıcılarına da basit ve etkili bir e-ticaret deneyimi sunmaktadır.

Cetasoft, ön muhasebe yönetimi konusunda güçlü bir çözümdür. İşletmelerin finansal süreçlerini düzenlemek ve takip etmek için tasarlanmış olan bu muhasebe programı, OpenCart ile entegre olarak çalışabilir. Bu entegrasyon, işletmelerin satış verilerini anlık olarak muhasebe sistemine aktarmalarını sağlar, böylece finansal yönetim daha etkili ve hızlı bir şekilde gerçekleştirilebilir.

Trendyol, Türkiye'nin önde gelen e-ticaret platformlarından biridir ve geniş müşteri kitlesi ile dikkat çeker. Trendyol Pazaryeri'nde yer almak, işletmelere potansiyel müşterilere ulaşma ve satışlarını artırma fırsatı sunar. Ancak, bu büyük pazaryerine entegre olmak için etkili bir API entegrasyonu gereklidir.

OpenCart'ın Cetasoft ve Trendyol ile API entegrasyonu, işletmelere birçok avantaj sunar. Bu entegrasyon ile:

Satış verileri anlık olarak muhasebe sistemine aktarılır, finansal yönetim kolaylaşır.

Trendyol Pazaryeri'nde ürünlerin yönetimi ve stok takibi daha etkili hale gelir.

Müşteri siparişleri ve ödemeleri otomatik olarak muhasebe sistemine kaydedilir.

OpenCart'ın Cetasoft ve Trendyol ile gerçekleştirdiği API entegrasyonu, işletmelere entegre bir e-ticaret deneyimi sunar. Finansal süreçlerin düzenli takibi, pazaryeri yönetimi ve müşteri ilişkilerinin güçlendirilmesi, işletmelerin rekabet avantajını artırmalarına yardımcı olur.

Bu entegrasyonlar, işletmelerin müşteri memnuniyetini artırarak daha fazla satış yapmalarına ve süreçlerini daha verimli yönetmelerine olanak tanır. OpenCart, Cetasoft ve Trendyol entegrasyonu, firma özelinde e-ticaretin dinamik dünyasında başarıya giden yolda önemli bir adımdır.

## 2. Cetasoft Web Servisleri

### wsOfis

Aşağıdaki işlemler desteklenmektedir. Tam bir tanımlama için lütfen [Hizmet Tanımını](#) gözden geçirin.

- [AddCustomer](#)  
Yeni bir müşteri kaydı oluşturur.
- [AddOrder](#)  
Yeni sipariş oluşturur.
- [AddOrderedItem](#)  
Oluşturulmuş siparişe ürün ekler.
- [AdvancedSearch](#)  
Cins, Özellik, Marka, Model, Beden, Renk, Cinsiyet alanlarında içeren araması yapar. Boş string geçilen alanları arama kriterine dahil etmez.
- [EditCustomer](#)  
Mevcut bir müşterinin bilgilerinde değişiklik yapar. Boş string geçilen alanları değiştirmeden bırakır.
- [GetAllProducts](#)  
Tüm ürünlerin bilgilerini alır.
- [GetBankAccounts](#)  
Banka hesaplarını ve kodlarını alır.
- [GetCategoryNames](#)  
Kategori isimlerini alır. iKategoriNo parametresine 1'den 5'e kadar bir sayı girilmelidir.
- [GetCustomer](#)  
İstenen müşterinin bilgilerini alır.
- [GetCustomerByName](#)  
Entegre edilen firma carisini getirir. Siparişte iMustesid bilgisi için kullanılır.
- [GetOrderStatus](#)  
Siparişin durumu hakkında bilgi alır.
- [GetOwnerOfWebServices](#)  
Web servislerinin altında çalıştığı kullanıcı adı.
- [GetProductByStockCode](#)  
Stok kodu verilen ürünün bilgilerini alır.
- [GetProductsByCategory](#)  
Belirtilen kategorideki ürünlerin bilgilerini alır. Boş string geçilen kategori adları filtreleme işlemine dahil edilmez.
- [GetProductsInStock](#)  
Stokta olan ürünlerin bilgilerini alır.
- [GetVersion](#)  
CetaSoft Ofis Web Servis versiyonunu döndürür.
- [SimpleSearch](#)  
Cins alanında içeren araması yapar.
- [SimpleSearchFullMatch](#)  
Cins alanında tam eşleşme araması yapar.

### 2.1 Web Servis Sınıfı (wsOfis):

Bu sınıf, bir web servisi olarak işlev gören temel yapıyı oluşturur.

WebService özneliğiyle işaretlenmiştir, bu da bu sınıfın bir web servisi olduğunu belirtir.

Web servisi, farklı HTTP isteklerini karşılamak için WebMethod özneliğiyle işaretlenmiş bir dizi metod sağlar.

### 2.2 Veritabanı İşlemleri

Kod, bir Microsoft Access veritabanına (*strDbConnStr*) bağlanır ve bu veritabanıyla etkileşimde bulunur.

Veritabanı bağlantısı ve sorguları için OleDb kullanılır.

## 2.3 Müşteri Yönetimi

Müşteri eklemek, düzenlemek ve bilgilerini almak için metotlar sağlanmıştır (*AddCustomer*, *EditCustomer*, *GetCustomer*).

Bu metotlar, müşteri bilgilerini almak veya veritabanına eklemek için gerekli olan parametreleri alır ve ilgili veritabanı işlemlerini gerçekleştirir.

### 2.3.1 AddCustomer

Yeni bir müşteri kaydı oluşturur.

```
public int AddCustomer(string strKullaniciAdi, string strSifre, string strAd, string strSoyad, string strIsTel, string strEvTel, string strFaks, string strCepTel, string strFaturaAdres, string strFaturaSehir, string strTeslimAdres, string strTeslimSehir, string strEmail, string strVatandaslikNo, string strVergiDairesi, string strVergiNo)
```

#### **Giriş Parametreleri :**

strAd	: En çok 15 karakter. Boş bırakılamaz.
strSoyad	: En çok 20 karakter. Boş bırakılamaz.
strIsTel	: En çok 20 karakter.
strEvTel	: En çok 20 karakter.
strFaks	: En çok 20 karakter.
strCepTel	: En çok 20 karakter.
strFaturaAdres	: En çok 255 karakter. Boş bırakılamaz.
strFaturaSehir	: En çok 25 karakter. Boş bırakılamaz.
strTeslimAdres	: En çok 255 karakter. Boş bırakılamaz.
strTeslimSehir	: En çok 25 karakter. Boş bırakılamaz.
strEmail	: En çok 255 karakter.
strVatandaslikNo	: En çok 15 karakter.
strVergiDairesi	: En çok 25 karakter.
strVergiNo	: En çok 15 karakter.

**Geri Dönüş Değeri :** Müşteriye atanan, tamsayı (int) türünden id değerini döndürür. Bu müşteriyle ilgili yapılacak diğer işlemlerde, bu id değeri diğer servislerin giriş parametresi olarak kullanılır.

**Ürettiği Özel İstisnalar :**

"strAd parametresi boş bırakılamaz."

"strSoyad parametresi boş bırakılamaz."

"strFaturaAdres parametresi boş bırakılamaz."

"strFaturaSehir parametresi boş bırakılamaz."

"strTeslimAdres parametresi boş bırakılamaz."

"strTeslimSehir parametresi boş bırakılamaz."

### 2.3.2 EditCustomer

Mevcut bir müşterinin bilgilerinde değişiklik yapar.Boş string geçilen alanları değiştirmeden bırakır.

```
public void EditCustomer(string strKullaniciAdi, string strSifre, int iMusteriId, string strAd, string strSoyad, string strIsTel, string strEvTel, string strFaks, string strCepTel, string strFaturaAdres, string strFaturaSehir, string strTeslimAdres, string strTeslimSehir, string strEmail, string strVatandaslikNo, string strVergiDairesi, string strVergiNo)
```

**Giriş Parametreleri :**

iMusteriId : Bilgilerinde değişiklik yapılmak istenen müşterinin AddCustomer servisi tarafından atanan id değeri.

strAd : En çok 15 karakter.

strSoyad : En çok 20 karakter.

strIstel : En çok 20 karakter.

strEvTel : En çok 20 karakter.

strFaks : En çok 20 karakter.

strCepTel : En çok 20 karakter.

strFaturaAdres : En çok 255 karakter.

strFaturaSehir : En çok 25 karakter.

strTeslimAdres : En çok 255 karakter.  
strTeslimSehir : En çok 25 karakter.  
strEmail : En çok 255 karakter.  
strVatandaslikNo : En çok 15 karakter.  
strVergiDairesi : En çok 25 karakter.  
strVergiNo : En çok 15 karakter.

**Geri Dönüş Değeri :**

Yok.

**Ürettiği Özel İstisnalar :**

"Değiştirilecek bilgi belirtilmemiş."

Tüm giriş parametreleri boş bırakılmışsa servis, müşteri bilgilerinde herhangi bir değişiklik yapmaz ve bu istisnayı üretir.

"Bu numarayla kayıtlı müşteri yok."

iMusteriId parametresine verdiğiniz değerle kayıtlı bir müşteri mevcut değilse bu mesajı içeren bir istisna oluşur.

### 2.3.3 GetCustomer

İstenen müşterinin bilgilerini alır.

```
public DataSet GetCustomer(string strKullaniciAdi, string strSifre, int iMusteriId)
```

**Giriş Parametreleri :**

iMusteriId : Bilgileri alınmak istenen müşterinin AddCustomer servisi tarafından atanan tamsayı (int) türünden id değeri.

**Geri Dönüş Değeri :** Aşağıda tanımı verilen alanları içeren bir tablo barındıran DataSet nesnesini döndürür. Eğer verilen Müşteri id'si bulunamazsa veri içermeyen bir dataset döndürür.

```
<xs:element name="Ad" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Soyad" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="IsTel" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="EvTel" type="xs:string" minOccurs="0"/>
<xs:element name="Faks" type="xs:string" minOccurs="0"/>
<xs:element name="CepTel" type="xs:string" minOccurs="0"/>
<xs:element name="FaturaAdres" type="xs:string" minOccurs="0"/>
<xs:element name="FaturaSehir" type="xs:string" minOccurs="0"/>
<xs:element name="TeslimAdres" type="xs:string" minOccurs="0"/>
<xs:element name="TeslimSehir" type="xs:string" minOccurs="0"/>
<xs:element name="Email" type="xs:string" minOccurs="0"/>
<xs:element name="VatandaslikNo" type="xs:string" minOccurs="0"/>
<xs:element name="VergiDairesi" type="xs:string" minOccurs="0"/>
<xs:element name="VergiNo" type="xs:string" minOccurs="0"/>
```

#### **Ürettiği Özel İstisnalar :**

Yok.

## **2.4 Sipariş İşlemleri**

Sipariş oluşturmak, sipariş durumunu almak, ürün eklemek gibi işlevler (AddOrder, AddOrderedItem, GetOrderStatus) sunulmuştur.

Bu metotlar, müşteriye ait siparişleri işlemek ve veritabanına kaydetmek için kullanılır.

### **2.4.1 AddOrder**

Yeni sipariş oluşturur.

```
public int AddOrder(string strKullaniciAdi, string strSifre, int iMusteriId, double
dTahsilatTutari, int iTahsilatTuru, int iTahsilatKuru, int iBankaHesapId, string
strKargoTercihi, string strNot)
```

#### **Giriş Parametreleri :**

iMusteriId : Siparişi veren müşterinin AddCustomer servisi tarafından atanan id değeri.

dTahsilatTutari : Tahsil edilen tutar.

iTahsilatTuru : Nakit için 0, Kredi kartı için 1, Havale / EFT için 2  
verilmelidir.

iTahsilatKuru : TL için 0, USD için 1, EURO için 2 olmalıdır.

iBankaHesapId : Tahsilatın yapıldığı bankanın id değeri.

strKargoTercihi : En çok 50 karakter.

strNot : En çok 255 karakter.

**Geri Dönüş Değeri :** Siparişe atanan, tamsayı (int) türünden id değerini döndürür. Bu değer, bu siparişe ilgili başka işlemler yaparken ilgili servislere giriş parametresi olarak verilir.

Ürettiği Özel İstisnalar :

"iTahsilatKuru, TL için 0, USD için 1, EURO için 2 olmalı."

"iTahsilatTuru, nakit için 0, kredi kartı için 1, havavale/EFT için 2 olmalı."

"Bu numarayla kayıtlı müşteri yok."

"Bu numarayla kayıtlı banka hesabı yok."

## 2.4.2 AddOrderedItem

Oluşturulmuş siparişe ürün ekler.

```
public void AddOrderedItem(string strKullaniciAdi, string strSifre, int iSiparisId,  
string strStokKodu, double dMiktar, double dTutar, int iTutarKuru)
```

### Giriş Parametreleri :

iSiparisId : Ürün eklenmek istenen siparişin AddOrder servisi tarafından atanan tamsayı (int) türünden id değeri.

strStokKodu : En çok 20 karakter. Boş bırakılamaz.

dMiktar : Sipariş miktarı. Stoğun kartında tanımlı ölçü birimi cinsinden.

dTutar : Satış tutarı.

iTutarKuru : TL için 0, USD için 1, EURO için 2 olmalıdır.

### Geri Dönüş Değeri :

Yok.

### Ürettiği Özel İstisnalar :

"iTutarKuru, TL için 0, USD için 1, EURO için 2 olmalı."

"Bu numarayla kayıtlı sipariş yok."

"Bu numarayla kayıtlı sipariş yok."

"Bu numarayla kayıtlı stok yok."

### 2.4.3 GetOrderStatus

Siparişin durumu hakkında bilgi alır.

```
public DataSet GetOrderStatus(string strKullaniciAdi, string strSifre, int iSiparisId)
```

#### Giriş Parametreleri :

iSiparisId : Durumu öğrenilmek istenen siparişin AddOrder servisi tarafından atanan tamsayı (int) türünden id değeri.

#### Geri Dönüş Değeri :

Aşağıda tanımı verilen alanları içeren bir tablo barındıran DataSet nesnesini döndürür. Eğer verilen siparis id'sini bulamazsa veri içermeyen bir DataSet döndürür.

```
<xs:element name="SiparisId" type="xs:int" minOccurs="0"/>
```

```
<xs:element name="SiparisZamani" type="xs:dateTime" minOccurs="0"/>
```

```
<xs:element name="SevkTarihi" type="xs:dateTime" minOccurs="0"/>
```

```
<xs:element name="SevkKArgoFirmasi" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="SevkKargoSubesi" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="GonderiKodu" type="xs:string" minOccurs="0"/>
```

### Ürettiği Özel İstisnalar :

Yok.

## 2.5 Ürün Yönetimi

Ürünleri hepsini almak, kategoriye göre almak, stokta olan ürünleri almak, stok koduna göre aramak gibi işlevler (GetAllProducts, GetProductsByCategory, GetProductsInStock, GetProductByStockCode) sağlanmıştır.

Bu metotlar, ürün bilgilerini almak için veritabanı sorgularını kullanır.



## 2.5.1 GetAllProducts

Tüm ürünlerin bilgilerini alır.

```
public DataSet GetAllProducts(string strKullaniciAdi, string strSifre)
```

### **Giriş Parametreleri :**

Özel giriş parametresi yok.

### **Geri Dönüş Değeri :**

Bir DataSet nesnesi döndürür. Eğer istenen kritere uyan hiç ürün bulunmazsa veri içermeyen bir DataSet nesnesi döndürür.

```
<xs:element name="StokKodu" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Cins" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Ozellik" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Model" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Marka" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Beden" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Renk" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Cinsiyet" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Kategori1" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Kategori2" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Kategori3" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Kategori4" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="Kategori5" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="AlisFiyati" type="xs:double" minOccurs="0"/>
```

```
<xs:element name="AlisFiyatiKuru" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="AlisFiyatiKDV" type="xs:string" minOccurs="0"/>
```

```
<xs:element name="SatisFiyati" type="xs:double" minOccurs="0"/>
```

```
<xs:element name="SatisFiyatiKuru" type="xs:string" minOccurs="0"/>
```

<xs:element name="SatisFiyatiKDV" type="xs:string" minOccurs="0"/>

<xs:element name="KDVOrani" type="xs:double" minOccurs="0"/>

<xs:element name="OTVOrani" type="xs:double" minOccurs="0"/>

<xs:element name="Miktar" type="xs:double" minOccurs="0"/>

<xs:element name="Birim" type="xs:string" minOccurs="0"/>

### **Ürettiği Özel İstisnalar :**

Yok.

## **2.5.2 GetProductsByCategory**

Belirtilen kategorideki ürünlerin bilgilerini alır. Boş string geçilen kategori adları filtreleme işlemine dahil edilmez.

```
public DataSet GetProductsByCategory(string strKullaniciAdi, string strSifre, string strKategori1, string strKategori2, string strKategori3, string strKategori4, string strKategori5)
```

### **Giriş Parametreleri :**

strKategori1 : 1 nolu stok grubunda aranmasını istediğiniz kriter.

strKategori2 : 2 nolu stok grubunda aranmasını istediğiniz kriter.

strKategori3 : 3 nolu stok grubunda aranmasını istediğiniz kriter.

strKategori4 : 4 nolu stok grubunda aranmasını istediğiniz kriter.

strKategori5 : 5 nolu stok grubunda aranmasını istediğiniz kriter.

Ürünün dönüş DataSet'ine dahil edilmesi için giriş parametrelerine verilen kriterlerin ilgili alanlarla tam eşleşmesini gerektirir.

### **Geri Dönüş Değeri :**

Aşağıda tanımlı alanları içeren bir tablo barındıran DataSet nesnesini döndürür. Eğer istenen kriterlere uyan hiç ürün bulunmazsa veri içermeyen bir DataSet nesnesi döndürür.

<xs:element name="StokKodu" type="xs:string" minOccurs="0"/>

<xs:element name="Cins" type="xs:string" minOccurs="0"/>

<xs:element name="Ozellik" type="xs:string" minOccurs="0"/>

```
<xs:element name="Model" type="xs:string" minOccurs="0"/>
<xs:element name="Marka" type="xs:string" minOccurs="0"/>
<xs:element name="Beden" type="xs:string" minOccurs="0"/>
<xs:element name="Renk" type="xs:string" minOccurs="0"/>
<xs:element name="Cinsiyet" type="xs:string" minOccurs="0"/>
<xs:element name="Kategori1" type="xs:string" minOccurs="0"/>
<xs:element name="Kategori2" type="xs:string" minOccurs="0"/>
<xs:element name="Kategori3" type="xs:string" minOccurs="0"/>
<xs:element name="Kategori4" type="xs:string" minOccurs="0"/>
<xs:element name="Kategori5" type="xs:string" minOccurs="0"/>
<xs:element name="AlisFiyati" type="xs:double" minOccurs="0"/>
<xs:element name="AlisFiyatiKuru" type="xs:string" minOccurs="0"/>
<xs:element name="AlisFiyatiKDV" type="xs:string" minOccurs="0"/>
<xs:element name="SatisFiyati" type="xs:double" minOccurs="0"/>
<xs:element name="SatisFiyatiKuru" type="xs:string" minOccurs="0"/>
<xs:element name="SatisFiyatiKDV" type="xs:string" minOccurs="0"/>
<xs:element name="KDVOrani" type="xs:double" minOccurs="0"/>
<xs:element name="OTVOrani" type="xs:double" minOccurs="0"/>
<xs:element name="Miktar" type="xs:double" minOccurs="0"/>
<xs:element name="Birim" type="xs:string" minOccurs="0"/>
```

### **Ürettiği Özel İstisnalar :**

Yok.

### **2.5.3 GetProductsInStock**

Stokta olan ürünlerin bilgilerini alır.

```
public DataSet GetProductsInStock(string strKullaniciAdi, string strSifre)
```

### **Giriş Parametreleri :**

Özel giriş parametresi yok.

### **Geri Dönüş Değeri :**

GetProductsByCategory servisinde tanımı verilen tabloyu barındıran DataSet nesnesini döndürür. Eğer istenen kritere uyan hiç ürün bulunmazsa veri içermeyen bir DataSet nesnesi döndürür.

### **Ürettiği Özel İstisnalar :**

Yok.

## **2.5.4 GetProductByStockCode**

Stok kodu verilen ürünün bilgilerini alır.

```
public DataSet GetProductByStockCode(string strKullaniciAdi, string strSifre, string strStokKodu)
```

### **Giriş Parametreleri :**

strStokKodu : Bilgileri alınmak istenen stoğun kodu.

### **Geri Dönüş Değeri :**

GetProductsByCategory servisinde tanımı verilen tabloyu barındıran DataSet nesnesini döndürür. Eğer istenen kritere uyan hiç ürün bulunmazsa veri içermeyen bir DataSet nesnesi döndürür.

### **Ürettiği Özel İstisnalar :**

Yok.

## **2.6 Güvenlik Kontrolü**

Kod, kullanıcı kimlik doğrulama için metot sağlar (CheckLoginInfo). Bu metot, kullanıcıların yetkilendirilmesini kontrol eder.

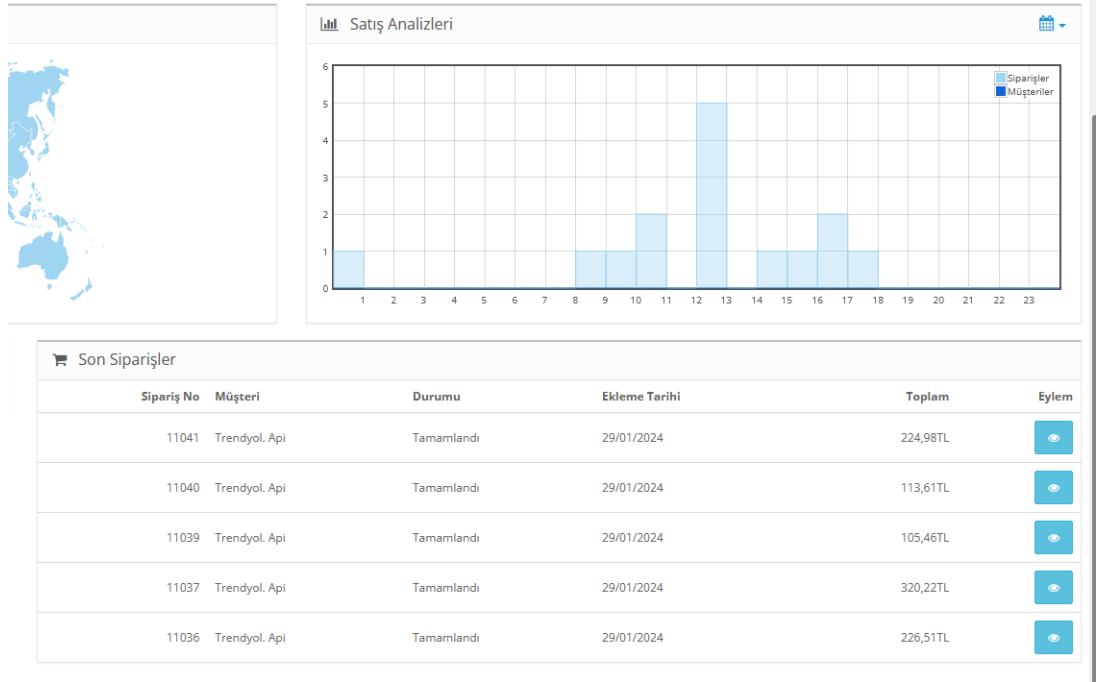
## **2.7 Diğer Fonksiyonlar**

Kategori isimlerini almak, ürün aramak için farklı metotlar sağlanmıştır.

Kod, istemcilerin isteklerine cevap vermek için çeşitli veri yapılarını (DataSet, DataTable) kullanır.

Bu şekilde, verilen kod bu ön muhasebe yönetim sistemini desteklemek üzere bir web servisi sağlar. Müşteri bilgilerini yönetmek, siparişler oluşturmak, ürünleri aramak ve stok durumunu kontrol etmek gibi çeşitli işlevleri barındırır. Güvenlik ve lisans kontrolü gibi temel önlemleri de içerir.

## 3 Trendyol Servisleri



### 3.1 updateProducts

Bu method ile Trendyol mağazanızda createProduct V2 servisiyle oluşturduğunuz ürünleri güncelleyebilirsiniz.

Bu servis üzerinden sadece ürün bilgileri güncellenmektedir. Stok ve fiyat değerlerini güncellemek için updatePriceAndInventory servisini kullanmanız gerekmektedir.

Yeni kategori ve kategori özellik değerleri eklenebileceği sebebiyle ürün güncellemelerinizden önce kullandığınız kategori ve kategori özellik değerlerinin güncel olup olmadığını getCategoryTree ve getCategoryAttributes servislerinden kontrol etmenizi öneririz.

Güncelleme isteğinden sonra "İçerik kontrol bekleniyor." statüsüne geçen ürünleriniz satışa açık durumda olabilir. Sipariş almak istemediğiniz durumlarda stok ve fiyat bilginizi sıfırlamanız gerekmektedir.

Her bir istek içerisinde gönderilebilecek maksimum item sayısı 1.000'dir.

Onaylı ürünlerde productMainId değeri güncellenmemektedir.

### 3.2 updatePriceAndInventory

Trendyol'a aktarılan ve onaylanan ürünlerin fiyat ve stok bilgileri eş zamana yakın güncellenir. Stok ve fiyat bilgilerini istek içerisinde ayrı ayrı gönderebilirsiniz.

Stok-fiyat gncelleme iřlemlerinde request body ierisinde deęiřiklik yapmadan aynı iřteęi tekrar atmanız halinde, sizlere hata mesajı dnecektir. Hata mesajı olarak "15 dakika boyunca aynı iřteęi tekrarlı olarak atamazsınız!" greceksiniz. Sadece deęiřen stok-fiyatlarınızı iřtek atacak řekilde sistemlerinizi dzeltmeniz gerekmektedir.

Quantity alanında gnderdięiniz stok , satılabilir stok bilgisidir. Satılabilir stok bilgisi sipariř alındıęında ya da tarafınızdan yeniden stok gnderildięinde gncellenir.

Stok-fiyat update iřlemlerinde maksimum 1000 item(sku) gncellemesi yapabilirsiniz.

rnleriniz iin maksimum 20 Bin adet stok ekleyebilirsiniz.

### 3.3 getShipmentPackages

Trendyol sistemine iletteęiniz rnler ile mřteriler tarafından verilen ve deme kontrolnde olan her sipariřin bilgisini bu method yardımıyla alabilirsiniz. Sistem tarafından deme kontrolnden sonra otomatik paketlenerek sipariř paketleri oluřturulur.

# 4 Web Sitesi Servisleri (Opencart)

## 4.1 Trendyol

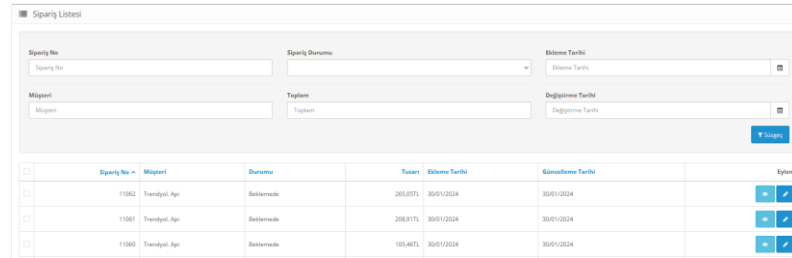
### 4.1.1 trendyol/updateProducts

Trendyol servislerinden updatePriceAndInventory fonksiyonuna gönderdiği parametreler ile ürünlerin fiyat ve stok bilgilerini günceller.

### 4.1.2 trendyol/getOrderPackages

Trendyol servislerinden getShipmentPackages fonksiyonuna gönderdiği parametreler ile trendyoldaki sipariş listesini çeker. Dönüş değerlerini altta belirtilen createOrder fonksiyonuna parametre olarak gönderir. createOrder fonksiyonundan olumlu cevap alması üzerine siparişi cetasoft servislerine aktararak muhasebeye işlenmesini sağlar.

```
← → ↻ https://www.suspus.com.tr/index.php?route=api/trendyol/getOrderPackages
Yer imlerini içe aktar... İlk Adım Dermakozmetik Admin Dermatoloji Admin AntiAging Kontrol Pan... Rip isimleri ve Aç
object(stdClass)#208 (2) { ["success"]=> string(47) "Başarılı: Sipariş başarıyla değiştirildi!" ["order_id"]=> int(11060) }
```



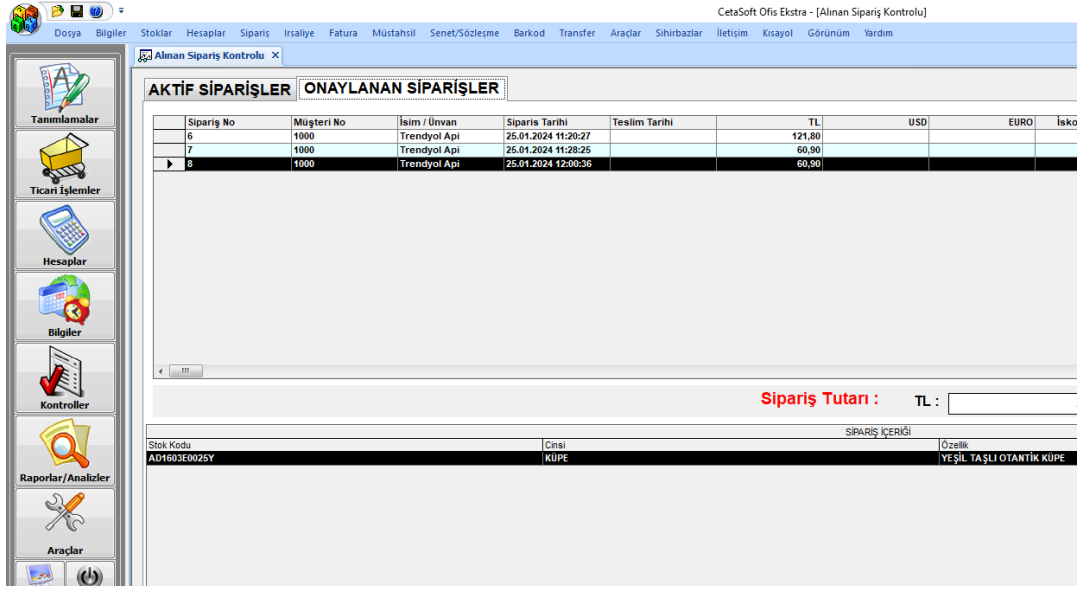
Sipariş No	Miçipari	Durumu	Tutar	Eklenme Tarihi	Güncellenme Tarihi	Eylem
11062	Trendyol Apr	Beklemede	285,03TL	30/01/2024	30/01/2024	[Eylem]
11061	Trendyol Apr	Beklemede	208,91TL	30/01/2024	30/01/2024	[Eylem]
11060	Trendyol Apr	Beklemede	105,46TL	30/01/2024	30/01/2024	[Eylem]

### 4.1.3 trendyol/createOrder

getOrderPackages fonksiyonundan dönen değerleri giriş parametresi olarak kullanarak site üzerinde siparişi oluşturur.



## 4.2 Cetasoft



### 4.2.1 cetasoft/getAllProducts

Cetasoft üzerinde bulunan GetAllProducts metodunu çağırır. Tüm ürünleri siteye akırarır. Var olanları günceller, olmayanları ekler.

### 4.2.2 cetasoft/addOrder

4.1.2 de bulunan trendyol/getOrderPackages tarafından 4.1.3 te bulunan createOrder fonksiyonunun çalıştırılmasından sonra alınan olumlu cevaba istinaden çalıştırılarak muhasebe programına siparişi aktarır.

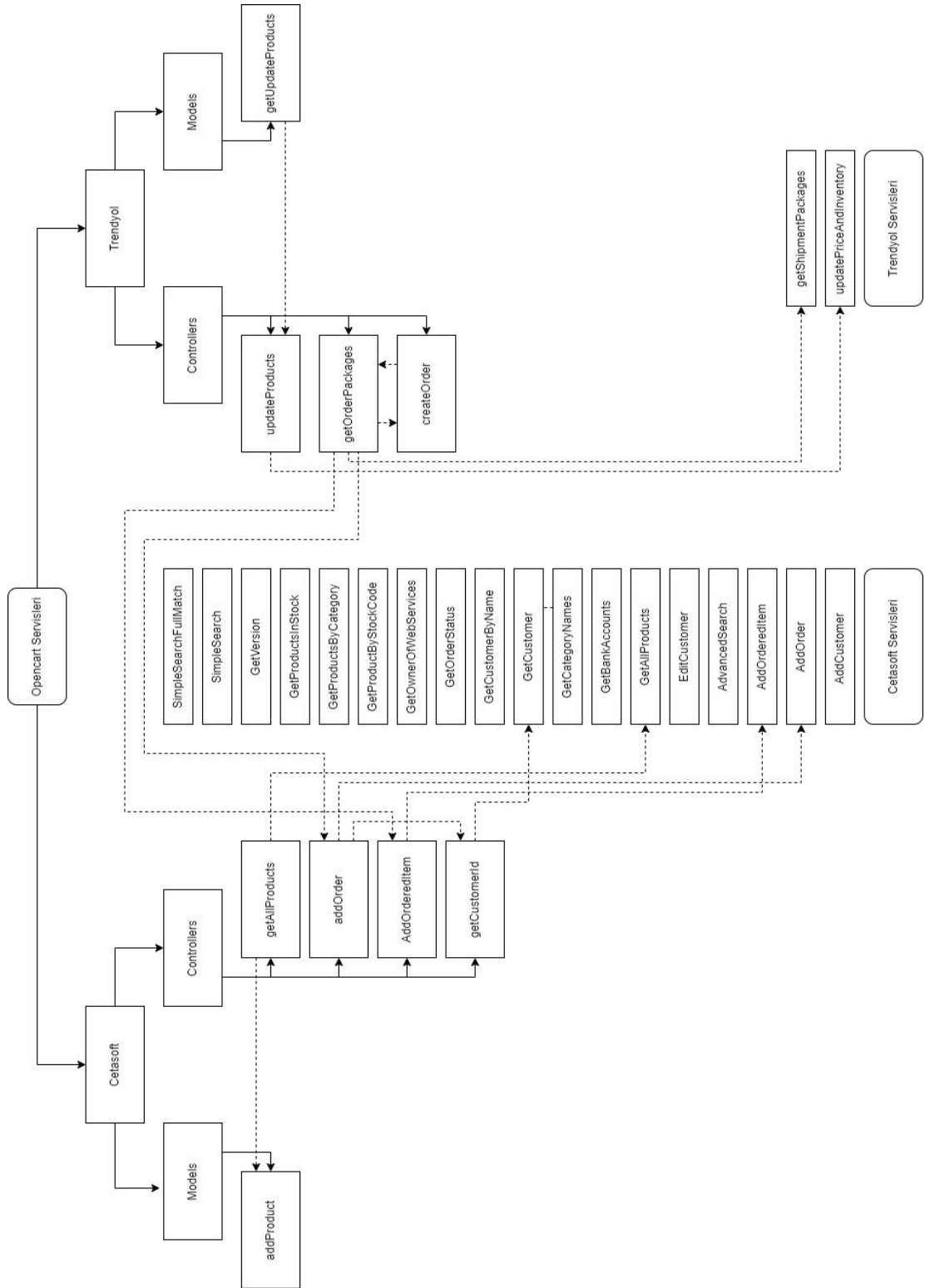
### 4.2.3 cetasoft/addOrderedItem

4.1.2 de bulunan trendyol/getOrderPackages tarafından Trendyoldan gelen ürünleri 4.2.2 addOrder fonksiyonundan dönen sipariş ID'li siparişe ekler ve siparişi sonlandırır.

### 4.2.4 cetasoft/getCustomerId

Adı verilen müşterinin müşteri ID'sini döner.

# 5 Akış Diyagramı



# Kaynaklar

## SOAP Servisleri

ASP.NET Soap:

<https://www.mshowto.org/asp-net-ile-web-servis-gelistirme.html>

<https://jast.hho.msu.edu.tr/index.php/JAST/article/download/144/132>

PHP Soap:

<https://www.php.net/soap>

## RestApi Servisleri

Trendyol Web Servisleri:

<https://developers.trendyol.com/docs/intro>

## PhpCURL

<https://www.php.net/manual/en/book.curl.php>

# EKLER

# CetaSoft Web Servis Kodları

```
using System;

using System.Collections.Generic;

using System.Web;

using System.Web.Services;

using System.Data;

using System.Data.OleDb;

using Microsoft.Win32;

using System.IO;

using System.Text;

namespace WebServisOfis

{

    /// <summary>

    /// Summary description for wsOfis

    /// </summary>

    [WebService(Namespace = "http://fabrika.egeelsanatlari.com/")]

    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]

    [System.ComponentModel.ToolboxItem(false)]

    public class wsOfis : System.Web.Services.WebService

    {

        static private string strDosyaAdi;

        static private string strDbConnStr;
```

```

public wsOfis()
{
    StreamReader sr = new StreamReader(Server.MapPath("~/database.path"));
    strDosyaAdi = sr.ReadLine();
    sr.Close();

    strDbConnStr = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" +
    strDosyaAdi + ";Persist Security Info=True;Jet OLEDB:Database Password=*****";
}

```

```

private Boolean CheckLoginInfo(string strKullaniciAdi, string strSifre)

```

```

{
    string strInvalidUserOrPass = "Kullanıcı adı veya şifre hatalı.";

    if (strKullaniciAdi.ToUpper() != "WEBUSER")
        throw new Exception(strInvalidUserOrPass);

    OleDbConnection oleCon = new OleDbConnection(strDbConnStr);
    oleCon.Open();

    OleDbCommand oleCom = new OleDbCommand("SELECT password FROM
    security WHERE username = 'WEBUSER'", oleCon);

    OleDbDataReader oleReader =
    oleCom.ExecuteReader(CommandBehavior.SingleRow);

    if (oleReader.HasRows == false) throw new Exception(strInvalidUserOrPass);

    oleReader.Read();
}

```

```

byte[] bytDecodedBytes = Convert.FromBase64String(oleReader.GetString(0));
string strDecodedText = Encoding.UTF8.GetString(bytDecodedBytes);

if (strSifre != strDecodedText)
throw new Exception(strInvalidUserOrPass);

oleReader.Close();

if (oleCon.State != ConnectionState.Closed) oleCon.Close();

return true;
} // private Boolean CheckLoginInfo(string strKullaniciAdi, string strSifre)

private string BuildSQLExpForAllProducts()
{
return "SELECT stokkodu AS StokKodu," +
"cinsi AS Cins," +
"ozellik AS Ozellik," +
"model AS Model," +
"marka AS Marka," +
"beden AS Beden," +
"renk AS Renk," +
"cinsiyet AS Cinsiyet," +
"grup AS Kategori1," +
"grup2 AS Kategori2," +
"grup3 AS Kategori3," +

```

```

"grup4 AS Kategori4," +
"grup5 AS Kategori5," +
"al AS AlisFiyati," +
"alkur AS AlisFiyatiKuru," +
"kdvdurumalis AS AlisFiyatiKDV," +
"sat AS SatisFiyati," +
"satkur AS SatisFiyatiKuru," +
"kdvdurum AS SatisFiyatiKDV," +
"kdv AS KDVOrani," +
"otvoran AS OTVOrani," +
"stok AS Miktar," +
"olcu AS Birim " +
"FROM kodlar WHERE tur <> 'D'";
} // private string BuildSQLExpForAllProducts()

```

```
private DataSet ExecQuery(string strQuery)
```

```
{
```

```
OleDbConnection oleCon = null;
```

```
oleCon = new OleDbConnection(strDbConnStr);
```

```
oleCon.Open();
```

```
OleDbDataAdapter oleDA = new OleDbDataAdapter(strQuery, oleCon);
```

```
DataSet dsTmp = new DataSet();
```



```

oleDA.Fill(dsTmp);

if (oleCon.State != ConnectionState.Closed) oleCon.Close();

return dsTmp;

} // private DataSet ExecQuery(string strQuery)

[WebMethod(Description="Yeni bir müşteri kaydı oluşturur.")]
public int AddCustomer(string strKullaniciAdi, string strSifre,
string strAd, string strSoyad,
string strIsTel, string strEvTel, string strFaks,
string strCepTel, string strFaturaAdres, string strFaturaSehir,
string strTeslimAdres, string strTeslimSehir, string strEmail,
string strVatandaslikNo, string strVergiDairesi, string strVergiNo)
{
OleDbConnection oleCon = null;
int iResult;

CheckLoginInfo(strKullaniciAdi, strSifre);

if (strAd == "") throw new Exception("strAd parametresi boş bırakılamaz.");
if (strSoyad == "") throw new Exception("strSoyad parametresi boş bırakılamaz.");
if (strFaturaAdres == "") throw new Exception("strFaturaAdres parametresi boş bırakılamaz.");
if (strFaturaSehir == "") throw new Exception("strFaturaSehir parametresi boş bırakılamaz.");
if (strTeslimAdres == "") throw new Exception("strTeslimAdres parametresi boş bırakılamaz.");

```

```
if (strTeslimSehir == "") throw new Exception("strTeslimSehir parametresi boş bırakılamaz.");
```

```
string strQuery =
```

```
"INSERT INTO adres (hat1, hat2, hat3, hat4, hat5, hat6, hat7, hat8, cari, TLbakiye, USDbakiye, EURObakiye, STRLbakiye, JPYbakiye, SFRbakiye, ADbakiye, DKbakiye, KDbakiye, bakiyegun, adi, soyadi, birlestir, tel1, evtel1, fax, cep, faturaadres, faturasehir, adres2, sehir, email, tckimlik, vd, vn)" +
```

```
" VALUES ('0', '0', '0', '0', '0', '0', '0', '0', '1', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 30, " +
```

```
"" + strAd + ", " + // adi
```

```
"" + strSoyad + ", " + // soyadi
```

```
"" + strAd + " " + strSoyad + ", " + // birlestir
```

```
(strIsTel == "" ? "NULL, " : "" + strIsTel + ", ") + // tel1
```

```
(strEvTel == "" ? "NULL, " : "" + strEvTel + ", ") + // evtel1
```

```
(strFaks == "" ? "NULL, " : "" + strFaks + ", ") + // fax
```

```
(strCepTel == "" ? "NULL, " : "" + strCepTel + ", ") + // cep
```

```
"" + strFaturaAdres + ", " + // faturaadres
```

```
"" + strFaturaSehir + ", " + // faturasehir
```

```
"" + strTeslimAdres + ", " + // adres2
```

```
"" + strTeslimSehir + ", " + // sehir
```

```
(strEmail == "" ? "NULL, " : "" + strEmail + ", ") + // email
```

```
(strVatandaslikNo == "" ? "NULL, " : "" + strVatandaslikNo + ", ") + // tckimlik
```

```
(strVergiDairesi == "" ? "NULL, " : "" + strVergiDairesi + ", ") + // vd
```

```
(strVergiNo == "" ? "NULL" : "" + strVergiNo + "")) + // vn
```

```
");
```

```
oleCon = new OleDbConnection(strDbConnStr);
```

```

oleCon.Open();

OleDbCommand oleCom = new OleDbCommand(strQuery, oleCon);

oleCom.ExecuteNonQuery();

// access veritabanı motoru @@IDENTITY'yi desteklemiyor.
// oleCom = new OleDbCommand("SELECT @@IDENTITY",oleCon);

oleCom.CommandText = "SELECT MAX(otonumara) FROM adres";

iResult = (int)oleCom.ExecuteScalar();

oleCom.CommandText = "UPDATE adres SET numara=" + iResult + " WHERE
otonumara=" + iResult;

oleCom.ExecuteNonQuery();

if (oleCon.State != ConnectionState.Closed) oleCon.Close();

return iResult;

} // public long AddNewCustomer(string strKullaniciAdi, string strSifre,

[WebMethod(Description = "Mevcut bir müşterinin bilgilerinde değişiklik yapar.
Boş string geçilen alanları değiştirmeden bırakır.")]

public void EditCustomer(string strKullaniciAdi, string strSifre,

int iMusteriId,

string strAd, string strSoyad,

string strIsTel, string strEvTel, string strFaks,

string strCepTel, string strFaturaAdres, string strFaturaSehir,

string strTeslimAdres, string strTeslimSehir, string strEmail,

```

```

string strVatandaslikNo, string strVergiDairesi, string strVergiNo)
{
OleDbConnection oleCon = null;

CheckLoginInfo(strKullaniciAdi, strSifre);

if (strAd == "" && strSoyad == "" && strIsTel == "" && strEvTel == "" &&
strFaks == "" &&

strCepTel == "" && strFaturaAdres == "" && strFaturaSehir == "" &&
strTeslimAdres == "" &&

strTeslimSehir == "" && strEmail == "" && strVatandaslikNo == "" &&
strVergiDairesi == "" &&

strVergiNo == "") throw new Exception("Değiştirilecek bilgi belirtilmemiş.");

string strQuery = "SELECT " +
"adi," + // 0
"soyadi," + // 1
"tell," + // 2
"evtel1," + // 3
"fax," + // 4
"cep," + // 5
"faturaadres," + // 6
"faturasehir," + // 7
"adres2," + // 8
"sehir," + // 9
"email," + // 10
"tckimlik," + // 11

```

```

"vd," +      // 12
"vn " +      // 13

"FROM adres WHERE numara = " + iMusteriId.ToString() + """;

oleCon = new OleDbConnection(strDbConnStr);

oleCon.Open();

OleDbCommand oleCom = new OleDbCommand(strQuery, oleCon);

OleDbDataReader oleReader =
oleCom.ExecuteReader(CommandBehavior.SingleRow);

if (oleReader.HasRows == false) throw new Exception("Bu numarayla kayıtlı
müşteri yok.");

oleReader.Read();

strQuery = "UPDATE adres SET ";

if (strAd != "") strQuery += "adi = " + strAd + ", ";

else strQuery += "adi = " + (oleReader.IsDBNull(0) ? "NULL, " : "" +
oleReader.GetString(0) + ", ");

if (strSoyad != "") strQuery += "soyadi = " + strSoyad + ", ";

else strQuery += "soyadi = " + (oleReader.IsDBNull(1) ? "NULL, " : "" +
oleReader.GetString(1) + ", ");

if (strAd != "" && strSoyad != "") strQuery += "birlestir = " + strAd + " " +
strSoyad + ", ";

```

```
else if (strAd != "" && strSoyad == "") strQuery += "birlestir = " + strAd + " " +  
(oleReader.IsDBNull(1) ? "" : oleReader.GetString(1)) + ", ";
```

```
else if (strAd == "" && strSoyad != "") strQuery += "birlestir = " +  
(oleReader.IsDBNull(0) ? "" : oleReader.GetString(0)) + " " + strSoyad + ", ";
```

```
if (strIsTel != "") strQuery += "tel1 = " + strIsTel + ", ";
```

```
else strQuery += "tel1 = " + (oleReader.IsDBNull(2) ? "NULL, " : "" +  
oleReader.GetString(2) + ", " );
```

```
if (strEvTel != "") strQuery += "evtel1 = " + strEvTel + ", ";
```

```
else strQuery += "evtel1 = " + (oleReader.IsDBNull(3) ? "NULL, " : "" +  
oleReader.GetString(3) + ", " );
```

```
if (strFaks != "") strQuery += "fax = " + strFaks + ", ";
```

```
else strQuery += "fax = " + (oleReader.IsDBNull(4) ? "NULL, " : "" +  
oleReader.GetString(4) + ", " );
```

```
if (strCepTel != "") strQuery += "cep = " + strCepTel + ", ";
```

```
else strQuery += "cep = " + (oleReader.IsDBNull(5) ? "NULL, " : "" +  
oleReader.GetString(5) + ", " );
```

```
if (strFaturaAdres != "") strQuery += "faturaadres = " + strFaturaAdres + ", ";
```

```
else strQuery += "faturaadres = " + (oleReader.IsDBNull(6) ? "NULL, " : "" +  
oleReader.GetString(6) + ", " );
```

```
if (strFaturaSehir != "") strQuery += "faturasehir = " + strFaturaSehir + ", ";
```

```
else strQuery += "faturasehir = " + (oleReader.IsDBNull(7) ? "NULL, " : "" +  
oleReader.GetString(7) + ", " );
```

```

if (strTeslimAdres != "") strQuery += "adres2 = " + strTeslimAdres + ", ";

else strQuery += "adres2 = " + (oleReader.IsDBNull(8) ? "NULL, " : "" +
oleReader.GetString(8) + ", ");

if (strTeslimSehir != "") strQuery += "sehir = " + strTeslimSehir + ", ";

else strQuery += "sehir = " + (oleReader.IsDBNull(9) ? "NULL, " : "" +
oleReader.GetString(9) + ", ");

if (strEmail != "") strQuery += "email = " + strEmail + ", ";

else strQuery += "email = " + (oleReader.IsDBNull(10) ? "NULL, " : "" +
oleReader.GetString(10) + ", ");

if (strVatandaslikNo != "") strQuery += "tckimlik = " + strVatandaslikNo + ", ";

else strQuery += "tckimlik = " + (oleReader.IsDBNull(11) ? "NULL, " : "" +
oleReader.GetString(11) + ", ");

if (strVergiDairesi != "") strQuery += "vd = " + strVergiDairesi + ", ";

else strQuery += "vd = " + (oleReader.IsDBNull(12) ? "NULL, " : "" +
oleReader.GetString(12) + ", ");

if (strVergiNo != "") strQuery += "vn = " + strVergiNo + " ";

else strQuery += "vn = " + (oleReader.IsDBNull(13) ? "NULL " : "" +
oleReader.GetString(13) + " ");

strQuery += "WHERE otonumara = " + iMusteriId.ToString();

oleReader.Close();

```

```

oleCom.CommandText = strQuery;

oleCom.ExecuteNonQuery();

if (oleCon.State != ConnectionState.Closed) oleCon.Close();

} // public void EditCustomer(string strKullaniciAdi, string strSifre,

[WebMethod(Description = "İstenen müşterinin bilgilerini alır.")]

public DataSet GetCustomer(string strKullaniciAdi, string strSifre, int iMusteriId)

{

string strQuery = "SELECT " +

"adi AS Ad, " + // 0

"soyadi AS Soyad, " + // 1

"tel1 AS IsTel, " + // 2

"evtel1 AS EvTel ," + // 3

"fax AS Faks, " + // 4

"cep AS CepTel, " + // 5

"faturaadres AS FaturaAdres, " + // 6

"faturasehir AS FaturaSehir, " + // 7

"adres2 AS TeslimAdres, " + // 8

"sehir AS TeslimSehir, " + // 9

"email AS Email, " + // 10

"tckimlik AS VatandaslikNo, " + // 11

"vd AS VergiDairesi, " + // 12

"vn AS VergiNo " + // 13

"FROM adres WHERE numara = '" + iMusteriId.ToString() + "'";

```



```
CheckLoginInfo(strKullaniciAdi, strSifre);
```

```
return ExecQuery(strQuery);
```

```
}
```

```
[WebMethod(Description = "Entegre edilen firma carisini getirir. Siparişte  
iMustesiId bilgisi için kullanılır.")]
```

```
public DataSet GetCustomerByName(string strKullaniciAdi, string strSifre, string  
strFirma)
```

```
{
```

```
string strQuery = "SELECT " +
```

```
"numara AS iMusteriId " + // 0
```

```
"FROM adres WHERE firma = '" + strFirma + "'";
```

```
CheckLoginInfo(strKullaniciAdi, strSifre);
```

```
return ExecQuery(strQuery);
```

```
}
```

```
[WebMethod(Description = "Banka hesaplarını ve kodlarını alır.")]
```

```
public DataSet GetBankAccounts(string strKullaniciAdi, string strSifre)
```

```
{
```

```
string strQuery = "SELECT " +
```

```
"id AS Id, " +
```

```
"banka AS BankaAdi, " +  
"sube AS Sube, " +  
"kur AS HesapKuru, " +  
"hesapno AS HesapNo " +  
"FROM banka hesap";
```

```
CheckLoginInfo(strKullaniciAdi, strSifre);
```

```
return ExecQuery(strQuery);
```

```
}
```

```
[WebMethod(Description="Yeni sipariş oluşturur.")]
```

```
public int AddOrder(string strKullaniciAdi, string strSifre,
```

```
int iMusteriId,
```

```
double dTahsilatTutari, int iTahsilatTuru, int iTahsilatKuru,
```

```
int iBankaHesapId,
```

```
string strKargoTercihi, string strNot)
```

```
{
```

```
OleDbConnection oleCon = null,
```

```
oleCon2 = null;
```

```
int iResult;
```

```
CheckLoginInfo(strKullaniciAdi, strSifre);
```

```
string strQuery = "SELECT " +
```

```
"adi," + // 0
"soyadi," + // 1
"tell," + // 2
"evtell," + // 3
"fax," + // 4
"cep," + // 5
"faturaadres," + // 6
"faturasehir," + // 7
"adres2," + // 8
"sehir," + // 9
"email," + // 10
"tckimlik," + // 11
"vd," + // 12
"vn " + // 13
"FROM adres WHERE numara =" + iMusteriId.ToString() + "";
```

```
if (iTahsilatKuru < 0 || iTahsilatKuru > 2) throw new Exception("iTahsilatKuru, TL için 0, USD için 1, EURO için 2 olmalı.");
```

```
if (iTahsilatTuru < 0 || iTahsilatTuru > 2) throw new Exception("iTahsilatTuru, nakit için 0, kredi kartı için 1, Havale/EFT için 2 olmalı.");
```

```
oleCon = new OleDbConnection(strDbConnStr);
```

```
oleCon2 = new OleDbConnection(strDbConnStr);
```

```
oleCon.Open();
```

```
OleDbCommand oleCom = new OleDbCommand(strQuery, oleCon);
```

```

OleDbDataReader oleReader =
oleCom.ExecuteReader(CommandBehavior.SingleRow);

if (oleReader.HasRows == false) throw new Exception("Bu numarayla kayıtlı
müşteri yok.");

oleReader.Read();

oleCon2.Open();

OleDbCommand oleCom2 = new OleDbCommand("SELECT id FROM bankahesap
WHERE id=" + iBankaHesapId, oleCon);

OleDbDataReader oleReader2 =
oleCom2.ExecuteReader(CommandBehavior.SingleRow);

if (oleReader2.HasRows == false) throw new Exception("Bu numarayla kayıtlı
banka hesabı yok.");

oleReader2.Close();

strQuery =

"INSERT INTO siparis (siparisno, adi, siparistarihi, aciklama, MusteriId,
TahsilatTutari, TahsilatKuru, TahsilatTuru, BankaHesapId, KargoTercihi)" +

" VALUES (-100, " +

"" + oleReader.GetString(0) + " " + oleReader.GetString(1) + ", " +

"NOW, " +

(strNot == "" ? "NULL, " : "" + strNot + ", ") +

iMusteriId.ToString() + ", " +

"" + ((float)dTahsilatTutari) + "" + ", " +

iTahsilatKuru.ToString() + ", " +

iTahsilatTuru.ToString() + ", " +

iBankaHesapId.ToString() + ", " +

(strKargoTercihi == "" ? "NULL " : "" + strKargoTercihi + " ") +

```

```

");
oleReader.Close();

oleCom.CommandText = strQuery;
oleCom.ExecuteNonQuery();

// access veritabanı motoru @@IDENTITY'yi desteklemiyor.
// oleCom = new OleDbCommand("SELECT @@IDENTITY",oleCon);
oleCom.CommandText = "SELECT MAX(otonumara) FROM siparis";
iResult = (int)oleCom.ExecuteScalar();

oleCom.CommandText = "UPDATE siparis SET siparisno=" + iResult + " WHERE
otonumara=" + iResult;

oleCom.ExecuteNonQuery();

if(oleCon.State != ConnectionState.Closed) oleCon.Close();
if(oleCon2.State != ConnectionState.Closed) oleCon2.Close();
return iResult;
} // public long AddOrder(string strKullaniciAdi, string strSifre,

[WebMethod(Description = "Oluşturulmuş siparişe ürün ekler.")]
public void AddOrderedItem(string strKullaniciAdi, string strSifre,
int iSiparisId, string strStokKodu,
double dMiktar, double dTutar, int iTutarKuru)
{
string strQuery;

```

```
OleDbConnection oleCon = null,
oleCon2 = null;

CheckLoginInfo(strKullaniciAdi, strSifre);

if (iTutarKuru < 0 || iTutarKuru > 2) throw new Exception("iTutarKuru, TL için 0,
USD için 1, EURO için 2 olmalı.");

oleCon = new OleDbConnection(strDbConnStr);
oleCon2 = new OleDbConnection(strDbConnStr);

oleCon.Open();

OleDbCommand oleCom = new OleDbCommand("SELECT adi, tutarYTL,
tutarUSD, tutarEURO FROM siparis WHERE otonumara = " +
iSiparisId.ToString(), oleCon);

OleDbDataReader oleReader =
oleCom.ExecuteReader(CommandBehavior.SingleRow);

if (oleReader.HasRows == false) throw new Exception("Bu numarayla kayıtlı sipariş
yok.");

oleReader.Read();

oleCon2.Open();

OleDbCommand oleCom2 = new OleDbCommand("SELECT tur, cinsi, ozellik, olcu
FROM kodlar WHERE stokkodu = " + strStokKodu + "'", oleCon);

OleDbDataReader oleReader2 =
oleCom2.ExecuteReader(CommandBehavior.SingleRow);

if (oleReader2.HasRows == false) throw new Exception("Bu numarayla kayıtlı stok
yok.");

oleReader2.Read();
```

```

strQuery =
"INSERT INTO siparistakip (adi, tur, cinsi, ozellik, olcu, siparisno, stokkodu, adet,
birim, satistoplam, alsatkur)" +
" VALUES (";
strQuery += (oleReader.IsDBNull(0) ? "NULL, " : "" + oleReader.GetString(0) + ",
");
strQuery += (oleReader2.IsDBNull(0) ? "NULL, " : "" + oleReader2.GetString(0) +
", ");
strQuery += (oleReader2.IsDBNull(1) ? "NULL, " : "" + oleReader2.GetString(1) +
", ");
strQuery += (oleReader2.IsDBNull(2) ? "NULL, " : "" + oleReader2.GetString(2) +
", ");
strQuery += (oleReader2.IsDBNull(3) ? "NULL, " : "" + oleReader2.GetString(3) +
", ");
strQuery +=
iSiparisId.ToString() + ", " +
"" + strStokKodu + ", " +
dMiktar.ToString() + ", " +
"" + ((float)(dTutar / dMiktar)) + "" + ", " +
"" + ((float)dTutar) + "" + ", ";
if (iTutarKuru == 0) strQuery += "TL)";
else if (iTutarKuru == 1) strQuery += "USD)";
else if (iTutarKuru == 2) strQuery += "EURO)";
oleReader2.Close();
oleCom2.CommandText = strQuery;

```

```

oleCom2.ExecuteNonQuery();

strQuery = "UPDATE siparis SET ";
if (iTutarKuru == 0) strQuery += "tutarYTL = " +
((float)(oleReader.IsDBNull(1) ? dTutar : oleReader.GetDouble(1) + dTutar)) + """;
else if (iTutarKuru == 1) strQuery += "tutarUSD = " +
"" + ((float)(oleReader.IsDBNull(2) ? dTutar : oleReader.GetDouble(2) + dTutar)) +
""";
else if (iTutarKuru == 2) strQuery += "tutarEURO = " +
"" + ((float)(oleReader.IsDBNull(3) ? dTutar : oleReader.GetDouble(3) + dTutar)) +
""";
strQuery += " WHERE siparisno = " + iSiparisId.ToString();

oleCom2.CommandText = strQuery;
oleCom2.ExecuteNonQuery();
oleReader.Close();

if (oleCon.State != ConnectionState.Closed) oleCon.Close();
if (oleCon2.State != ConnectionState.Closed) oleCon2.Close();
} // public void AddOrderedItem(string strKullaniciAdi, string strSifre,

[WebMethod(Description = "Siparişin durumu hakkında bilgi alır.")]
public DataSet GetOrderStatus(string strKullaniciAdi, string strSifre, int iSiparisId)
{
string strQuery = "SELECT " +
"otonumara AS SiparisId, " +

```



```

"siparistarihi AS SiparisZamani, " +
"SevkTarihi, SevkKargoFirmasi, SevkKargoSubesi, GonderiKodu " +
"FROM siparis WHERE siparisno = " + iSiparisId.ToString();

CheckLoginInfo(strKullaniciAdi, strSifre);

return ExecQuery(strQuery);

} // public DataSet GetOrderStatus(string strKullaniciAdi, string strSifre, long
longSiparisNo)

[WebMethod(Description = "Belirtilen kategorideki ürünlerin bilgilerini alır. Boş
string geçilen kategori adları filtreleme işlemine dahil edilmez.")]

public DataSet GetProductsByCategory(string strKullaniciAdi, string strSifre,
string strKategori1, string strKategori2, string strKategori3,
string strKategori4, string strKategori5)
{
string strQuery;

CheckLoginInfo(strKullaniciAdi, strSifre);

strQuery = BuildSQLExpForAllProducts();

if (strKategori1 != "") strQuery += " AND grup = '" + strKategori1 + "'";
if (strKategori2 != "") strQuery += " AND grup2 = '" + strKategori2 + "'";
if (strKategori3 != "") strQuery += " AND grup3 = '" + strKategori3 + "'";
if (strKategori4 != "") strQuery += " AND grup4 = '" + strKategori4 + "'";

```

```

if (strKategori5 != "") strQuery += " AND grup5 = " + strKategori5 + "";

return ExecQuery(strQuery);

} // public DataSet GetProductByCategory(string strKullaniciAdi, string strSifre,

[WebMethod(Description = "Cins, Özellik, Marka, Model, Beden, Renk, Cinsiyet
alanlarında içeren araması yapar. Boş string geçilen alanları arama kriterine dahil
etmez.")])

public DataSet AdvancedSearch(string strKullaniciAdi, string strSifre,
string strCins, string strOzellik, string strMarka, string strModel,
string strBeden, string strRenk, string strCinsiyet)
{
string strQuery;

CheckLoginInfo(strKullaniciAdi, strSifre);

strQuery = BuildSQLExpForAllProducts();

if (strCins != "") strQuery += " AND cinsi LIKE '%" + strCins + "%'";
if (strOzellik != "") strQuery += " AND ozellik LIKE '%" + strOzellik + "%'";
if (strMarka != "") strQuery += " AND marka LIKE '%" + strMarka + "%'";
if (strModel != "") strQuery += " AND model LIKE '%" + strModel + "%'";
if (strBeden != "") strQuery += " AND beden LIKE '%" + strBeden + "%'";
if (strRenk != "") strQuery += " AND renk LIKE '%" + strRenk + "%'";
if (strCinsiyet != "") strQuery += " AND cinsiyet LIKE '%" + strCinsiyet + "%'";

```

```

return ExecQuery(strQuery);

} // public DataSet AdvancedSearch(string strKullaniciAdi, string strSifre,

[WebMethod(Description = "Cins alanında tam eşleşme araması yapar.")]

public DataSet SimpleSearchFullMatch(string strKullaniciAdi, string strSifre, string
strCins)

{

string strQuery;

CheckLoginInfo(strKullaniciAdi, strSifre);

strQuery = BuildSQLExpForAllProducts();
strQuery += " AND cinsi = '" + strCins + "'";

return ExecQuery(strQuery);

} // public DataSet SimpleSearchFullMatch(string strKullaniciAdi, string strSifre,
string strCins)

[WebMethod(Description = "Cins alanında içeren araması yapar.")]

public DataSet SimpleSearch(string strKullaniciAdi, string strSifre, string strCins)

{

string strQuery;

CheckLoginInfo(strKullaniciAdi, strSifre);

strQuery = BuildSQLExpForAllProducts();

```

```

strQuery += " AND cinsi LIKE '%" + strCins + "%";

return ExecQuery(strQuery);

} // public DataSet SimpleSearch(string strKullaniciAdi, string strSifre, string
strCins)

[WebMethod(Description = "Stok kodu verilen ürünün bilgilerini alır.")]

public DataSet GetProductByStockCode(string strKullaniciAdi, string strSifre, string
strStokKodu)

{

string strQuery;

CheckLoginInfo(strKullaniciAdi, strSifre);

strQuery = BuildSQLExpForAllProducts();

strQuery += " AND stokkodu = '" + strStokKodu + "'";

return ExecQuery(strQuery);

} // public DataSet GetProductByStockCode(string strKullaniciAdi, string strSifre,
string strStokKodu)

[WebMethod(Description = "Stokta olan ürünlerin bilgilerini alır.")]

public DataSet GetProductsInStock(string strKullaniciAdi, string strSifre)

{

string strQuery;

CheckLoginInfo(strKullaniciAdi, strSifre);

```

```

strQuery = BuildSQLExpForAllProducts();

strQuery += " AND stok > 0";

return ExecQuery(strQuery);

} // public DataSet GetProductsInStock(string strKullaniciAdi, string strSifre)

[WebMethod(Description = "Tüm ürünlerin bilgilerini alır.")]

public DataSet GetAllProducts(string strKullaniciAdi, string strSifre)

{

string strQuery;

CheckLoginInfo(strKullaniciAdi, strSifre);

strQuery = BuildSQLExpForAllProducts();

return ExecQuery(strQuery);

} // public DataSet GetAllProducts(string strKullaniciAdi, string strSifre)

[WebMethod(Description="Kategori isimlerini alır. iKategoriNo parametresine 1'den
5'e kadar bir sayı girilmelidir.")]

public DataSet GetCategoryNames(string strKullaniciAdi, string strSifre, int
iKategoriNo)

{

string strFieldName, strFieldAlias;

string strQuery;

```

```

CheckLoginInfo(strKullaniciAdi, strSifre);

switch (iKategoriNo)
{
case 1: strFieldName = "grup"; strFieldAlias = "Kategori1";
break;

case 2: strFieldName = "grup2"; strFieldAlias = "Kategori2";
break;

case 3: strFieldName = "grup3"; strFieldAlias = "Kategori3";
break;

case 4: strFieldName = "grup4"; strFieldAlias = "Kategori4";
break;

case 5: strFieldName = "grup5"; strFieldAlias = "Kategori5";
break;

default:

throw new Exception("iKategoriNo parametresine, 1 ile 5 arasında bir değer
girilmelidir.");

}

strQuery = "SELECT DISTINCT " + strFieldName + " AS " + strFieldAlias + "
FROM kodlar WHERE "+strFieldName+" IS NOT NULL";

return ExecQuery(strQuery);

} // public DataSet GetCategoryNames(int iKategoriNo)

[WebMethod(Description = "Web servislerinin altında çalıştığı kullanıcı adı.")]
public string GetOwnerOfWebServices()

```

```
{  
return Environment.UserName;  
} // public string GetOwnerOfWebServices()  
  
[WebMethod(Description = "CetaSoft Ofis Web Servis versiyonunu döndürür.")]  
public string GetVersion()  
{  
return "1.0";  
} // public string GetVersion()  
  
} // public class wsOfis : System.Web.Services.WebService  
  
} // namespace WebServisOfis  
  
//Müşteriye atanan, tamsayı (int) türünden id değerini döndürür. Bu müşteriyle ilgili yapılacak diğer işlemlerde, bu id değeri diğer servislerin giriş parametresi olarak kullanılır.
```

# Opencart Web Servis Kodlari

## Cetasoft

### Controller

```
<?php
class ControllerApiCetasoft extends Controller {
private $username = "WEBUSER";
private $password = "0000";
private $uriWSDL =
"http://fabrika.egeelsanatlari.com/WebServisOfis/wsOfis.asmx?wsdl";
private $options = array(
"trace" => true,
"exception" => true
);
private $client;
public function __construct($registry) {
parent::__construct($registry);
$this->client = new SoapClient($this->uriWSDL, $this->options);
if(!$this->client){
echo "<table width='100%' height='100%'>\n<tr>\n<td valign='middle'
align='center'>\n";
echo "<table width='30%' height='30%'>\n<tr>\n<td align='center'
bgcolor='#CCCCCC'>";
echo "Baglantı Kurulamadı";
echo "</td>\n</tr>\n</table>\n</td>\n</tr>\n</table>";
}
exit;
```



```

}
}

public function getAllProducts(){

try {

$this->load->model("api/cetasoft");

$allProductsParams = array("strKullaniciAdi" => (string)$this->username, "strSifre"
=> (string)$this->password);

$allProducts = $this->client->GetAllProducts($allProductsParams);

//var_dump($allProducts);

$allProductsXmlString = $allProducts->GetAllProductsResult->any;

$allProductsXmlParse = new SimpleXMLElement($allProductsXmlString);

//var_dump($allProductsXmlParse);

foreach($allProductsXmlParse->NewDataSet->Table as $product){

$results[] = $this->model_api_cetasoft->addProduct($product) . ' ID\'li Ürün
Eklendi. - ' . $product->Ozellik;

}

$this->response->addHeader('Content-Type: application/json');

$this->response->setOutput(json_encode($results));

} catch (SoapFault $fault) {

// Hata durumunda istisnayı göster

echo "SOAP Hatası: " . $fault->getMessage() . "\n";

echo "Request:\n" . $this->client->__getLastRequest() . "\n";

echo "Response:\n" . $this->client->__getLastResponse() . "\n";

}

}
}

```

```

public function addOrder($data = array("firma" => "TRENDYOL",
"cargoProviderName" => "Trendyol Express Marketplace")){

try {

if(isset($this->request->post["params"])){

$data = $this->request->post["params"];

}

$customerId = $this->getCustomerId(array("firma" => (string)$data["firma"]));

$addOrderParams = array("strKullaniciAdi" => (string)$this->username, "strSifre"
=> (string)$this->password, "iMusteriId" => (int)$customerId, "dTahsilatTutari" =>
(float)0, "iTahsilatTuru" => (int)2, "iTahsilatKuru" => (int)0, "iBankaHesapId" =>
(int)1, "strKargoTercihi" => (string)$data["cargoProviderName"], "strNot" =>
(string)");

//var_dump($addOrderParams);

$addOrder = $this->client->AddOrder($addOrderParams);

echo (int)$addOrder->AddOrderResult;

} catch (SoapFault $fault){

// Hata durumunda istisnayı göster

echo "SOAP Hatası: " . $fault->getMessage() . "\n";

echo "Request:\n" . $this->client->__getLastRequest() . "\n";

echo "Response:\n" . $this->client->__getLastResponse() . "\n";

}

}

public function AddOrderedItem(){

if(isset($this->request->post["products"]) && isset($this->request-
>post["params"])){

$params = $this->request->post["params"];

```

```

try {
foreach($this->request->post["products"] as $product){
$SiSiparisId = $params["iSiparisId"];
$strStokKodu = $product["gtin"];
$dMiktar = $product["qty"];
$dTutar = $product["qty"] * $product["price"];
$iTutarKuru = 0;
$AddOrderedItemparams = array("strKullaniciAdi" => (string)$this->username,
"strSifre" => (string)$this->password, "iSiparisId" => (int)$iSiparisId,
"strStokKodu" => (string)$strStokKodu, "dMiktar" => (float)$dMiktar, "dTutar" =>
(float)$dTutar, "iTutarKuru" => (int)$iTutarKuru);
$AddOrderedItem = $this->client->AddOrderedItem($AddOrderedItemparams);
}
} catch (SoapFault $fault) {
// Hata durumunda istisnayı göster
echo "SOAP Hatası: " . $fault->getMessage() . "\n";
echo "Request:\n" . $this->client->__getLastRequest() . "\n";
echo "Response:\n" . $this->client->__getLastResponse() . "\n";
}
}
}

private function getCustomerId($data = array()){
try {
$customerIdParams = array("strKullaniciAdi" => $this->username, "strSifre" =>
$this->password, "strFirma" => $data['firma']);
$customerId = $this->client->GetCustomerByName($customerIdParams);
}
}
}

```

```

$customerIdXmlString = $customerId->GetCustomerByNameResult->any;
$customerIdXmlParse = new SimpleXMLElement($customerIdXmlString);
$customer = $customerIdXmlParse->NewDataSet->Table;

return $customer->iMusteriId;

} catch (SoapFault $fault) {

// Hata durumunda istisnayı göster

echo "SOAP Hatası: " . $fault->getMessage() . "\n";

echo "Request:\n" . $this->client->__getLastRequest() . "\n";

echo "Response:\n" . $this->client->__getLastResponse() . "\n";

}

}

```

```

public function info(){

phpinfo();

$currentDateTime = date('d.m.Y H:i:s');

echo "<p align='center'>Şu anki tarih ve saat: " . $currentDateTime . "</p>";

}

}

```

## Model

```
<?php
```

```

class ModelApiCetasoft extends Model {

public function addProduct($data) {

$this->db->query("REPLACE INTO " . DB_PREFIX . "in_product SET ID =
NULL, StokKodu = " . $this->db->escape($data->StokKodu) . ", Cins = " . $this-
>db->escape($data->Cins) . ", Ozellik = " . $this->db->escape($data->Ozellik) . ",
Model = " . $this->db->escape($data->Model) . ", Renk = " . $this->db-
>escape($data->Renk) . ", Kategori1 = " . $this->db->escape($data->Kategori1) . ",

```

```

Kategori2 = "" . $this->db->escape($data->Kategori2) . "", AlisFiyati = "" .
(float)$this->db->escape($data->AlisFiyati) . "", AlisFiyatiKuru = "" . $data-
>AlisFiyatiKuru . "", AlisFiyatiKDV = "" . (int)$data->AlisFiyatiKDV . "", SatisFiyati
= "" . (float)$data->SatisFiyati . "", SatisFiyatiKuru = "" . $data->SatisFiyatiKuru . "",
SatisFiyatiKDV = "" . (int)$this->db->escape($data->SatisFiyatiKDV) . "",
KDVOrani = "" . (int)$data->KDVOrani . "", Miktar = "" . (int)$data->Miktar . "",
Birim = "" . $data->Birim . "");

```

```

$product_id = $this->db->getLastId();

```

```

return $product_id;

```

```

}

```

```

}

```

## Trendyol

### Controller

```

<?php

```

```

class ControllerApiTrendyol extends Controller {

```

```

    private $supplierid = '123456';

```

```

    private $apiusername = 'apiusername';

```

```

    private $apiuserpassword = 'apiuserpassword';

```

```

    public function setCategoryTree(){

```

```

        $this->load->model('api/trendyol');

```

```

        $this->model_api_trendyol->delTrendyolCategories();

```

```

        $categories = $this->getCategoryTree();

```

```

        $this->serializeCategory($categories);

```

```

    }

```

```

public function setShipmentProviders(){

$data["url"] = "https://api.trendyol.com/sapigw/shipment-providers";

$data['headers'] = array(

'Content-Type:application/json',

'Authorization: Basic '. base64_encode("{ $this->apiusername }:{ $this->apiuserpassword }")

);

$results = json_decode($this->curlget($data), true);

//var_dump($result);

$this->load->model('api/trendyol');

$this->model_api_trendyol->delShipmentProviders();

foreach($results as $shipmentProvider){

$this->model_api_trendyol->setShipmentProviders($shipmentProvider);

}

}

public function categoryAttributesList(){

$categoryId = $this->request->get['categoryId'];

$result = $this->getCategoryAttributes($categoryId);

$this->response->addHeader('Content-Type: application/json');

$this->response->setOutput($result);

}

public function updateProducts2(){

```

```

$data['url'] = "https://api.trendyol.com/sapigw/suppliers/{ $this->supplierid}/products/price-and-inventory";

$data['headers'] = array(

'Content-Type:application/json',

'Authorization: Basic '. base64_encode("{ $this->apiusername}:{ $this->apiuserpassword}")

);

$this->load->model("catalog/product");

$products = $this->model_catalog_product->getProducts();

//var_dump($products);

$totalproducts = count($products);

$step = 20;

for($a=0;$a<=$totalproducts;$a+=20){

if(($a + $step) > $totalproducts){

$step = $totalproducts - $a;

}

foreach(array_slice($products, $a, $step) as $product){

if($product['special']){

$specialprice = (float)round($product['special'] * 1.20, 0)-0.1;

//$specialprice = round((float)$product['price'] * 1.10, 2);

} else {

$specialprice = (float)round($product['price'] * 1.20, 0)-0.1;

}

$data['products']['items'][] = [

"barcode" => $product['ean'],

"quantity" => $product['status'] != 0 ? $product['quantity'] : 0,

```

```

"listPrice" => (float)round($product['price'] * 1.20, 0)-0.1,
"salePrice" => (float)round($product['price'] * 1.20, 0)-0.1
];
}

$data['json'] = json_encode($data['products']);

//var_dump($data['json']);

$result = $this->curl($data);

var_dump($result);

unset($data['products']);

unset($data['json']);

sleep(2);

}

}

public function updateProducts(){

$data['url'] = "https://api.trendyol.com/sapigw/suppliers/{ $this->supplierid}/products/price-and-inventory";

$data['headers'] = array(

'Content-Type:application/json',

'Authorization: Basic '. base64_encode("{ $this->apiusername}:{ $this->apiuserpassword}")

);

$this->load->model("api/trendyol");

$products = $this->model_api_trendyol->getUpdateProducts();

if(count($products) > 0){

```



```

foreach($products as $product){
    $data['products']['items'][] = [
        "barcode" => $product['ean'],
        "quantity" => $product['status'] != 0 ? $product['quantity'] : 0,
        "listPrice" => round((float)$product['price'] * 1.20, 0)-0.1,
        "salePrice" => round((float)$product['price'] * 1.20, 0)-0.1
    ];
}

$data['json'] = json_encode($data['products']);

//var_dump($data['json']);

$results = $this->curl($data);

//var_dump($results);

$result = json_decode($results);

//var_dump($result);

if($result->batchRequestId != ""){
    $this->model_api_trendyol->updateProductChanges();
}

$this->response->addHeader('Content-Type: application/json');

$this->response->setOutput($results);

} else {

$result = array(

'Status' => 'Güncelenecek Ürün Yok'

);

$results = json_encode($result);

$this->response->addHeader('Content-Type: application/json');

```

```

$this->response->setOutput($results);

}

}

public function getOrderPackages(){

$data['url']="https://api.trendyol.com/sapigw/suppliers/{ $this->supplierid}/orders?status=Created";

//$data['url']="https://api.trendyol.com/sapigw/suppliers/{ $this->supplierid}/orders?stratDate=" . strtotime(date('Y-m-d'));

$data['headers'] = array(

'Content-Type:application/json',

'Authorization: Basic '. base64_encode("{ $this->apiusername }:{ $this->apiuserpassword }")

);

$data['query'] = [

"startDate" => strtotime(date('Y-m-d'),"-1 days"),

"endDate" => strtotime(date('Y-m-d'))

];

$results = json_decode($this->curlget($data));

//$this->response->addHeader('Content-Type: application/json');

//$this->response->setOutput($results);

//var_dump($results);

if(count($results->totalElements) != 0){

$this->load->model('localisation/zone');

foreach($results->content as $content){

$post['key'] = $this->key;

$post['orderNumber'] = $content->orderNumber;

```

```

//=====
=====

$post['customerFirstName'] = "Trendyol";

$post['customerLastName'] = "Api";

//=====
=====

$post['invoiceFirstName'] = $content->invoiceAddress->firstName;

$post['invoiceLastName'] = $content->invoiceAddress->lastName;

$post['invoiceAddress1'] = substr($content->invoiceAddress->address1,0,120);

$post['invoiceAddress2'] = substr($content->invoiceAddress->address2,0,120);

$invoiceCityID = $this->model_localisation_zone->getZoneByName(trim($content->invoiceAddress->city));

$post['invoiceCityID'] = $invoiceCityID['zone_id'];

$post['invoiceDisrict'] = $content->invoiceAddress->district;

//=====
=====

$post['shipmentFirstName'] = $content->shipmentAddress->firstName;

$post['shipmentLastName'] = $content->shipmentAddress->lastName;

$post['shipmentAddress1'] = substr($content->invoiceAddress->address1,0,120);

$post['shipmentAddress2'] = substr($content->invoiceAddress->address2,0,120);

$shipmentCityID = $this->model_localisation_zone->getZoneByName(trim($content->shipmentAddress->city));

$post['shipmentCityID'] = $shipmentCityID['zone_id'];

$post['shipmentDisrict'] = $content->shipmentAddress->district;

//=====
=====

$update['url'] = "https://api.trendyol.com/sapigw/suppliers/{ $this->supplierid}/shipment-packages/{ $content->id}";

//var_dump($update);

```

```
//=====
```

```
foreach($content->lines as $line){  
    $update['quantity'] = $line->quantity;  
    $update['barcode'] = $line->barcode;  
    $update['price'] = $line->price;  
    $putp['lines']= [  
    [  
        "lineId" => $line->id,  
        "quantity" => (int)$line->quantity  
    ]  
    ];  
    $putp['params'] = [];  
    $putp['status'] = "Picking";  
    $put['lines']= [  
    [  
        "lineId" => $line->id,  
        "quantity" => (int)$line->quantity  
    ]  
    ];  
    $put['params'] = [  
        "invoiceNumber"=>"Trendyol".strtotime(date('Y-m-d H:i:s'))  
    ];  
    $put['status'] = "Invoiced";  
    $post['products'][] = array(  
        "gtin" => $update['barcode'],
```

```

"qty" => $update['quantity'],
"price" => $update['price']
);
$jsonsp[] = json_encode($putp);
//$this->response->addHeader('Content-Type: application/json');
//$this->response->setOutput($jsonsp);
$jsons[] = json_encode($put);
//var_dump($post);

//=====
===
}

//=====
=====

$cho =
curl_init("https://www.suspus.com.tr/index.php?route=api/trendyol/createorder");
curl_setopt($cho, CURLOPT_RETURNTRANSFER, true);
curl_setopt($cho, CURLOPT_POST, true);
curl_setopt($cho, CURLOPT_POSTFIELDS, http_build_query($post));
$response = json_decode(curl_exec($cho));
var_dump($response);
//$response = curl_exec($cho);
//echo $response;
curl_close($cho);

//=====
=====

if($response->success){
//Muhasebeye Aktar

```

```
$cPost['params'] = array('firma'=>'TRENDYOL','cargoProviderName' => $content->cargoProviderName);
```

```
$cho =  
curl_init("https://www.suspus.com.tr/index.php?route=api/cetasoft/addorder");
```

```
curl_setopt($cho, CURLOPT_RETURNTRANSFER, true);
```

```
curl_setopt($cho, CURLOPT_POST, true);
```

```
curl_setopt($cho, CURLOPT_POSTFIELDS,http_build_query($cPost));
```

```
$post['params']['iSiparisId'] = curl_exec($cho);
```

```
$cho =  
curl_init("https://www.suspus.com.tr/index.php?route=api/cetasoft/addorderitem")  
;
```

```
curl_setopt($cho, CURLOPT_RETURNTRANSFER, true);
```

```
curl_setopt($cho, CURLOPT_POST, true);
```

```
curl_setopt($cho, CURLOPT_POSTFIELDS,http_build_query($post));
```

```
curl_exec($cho);
```

```
file_put_contents('post_log.txt', print_r($post, true), FILE_APPEND);
```

```
foreach($jsonsp as $jsonp){
```

```
$ch = curl_init($update['url']);
```

```
curl_setopt($ch, CURLOPT_HTTPHEADER, $data['headers']);
```

```
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
```

```
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "PUT");
```

```
curl_setopt($ch, CURLOPT_POSTFIELDS,$jsonp);
```

```
$response = curl_exec($ch);
```

```
//var_dump($response);
```

```
curl_close($ch);
```

```

}

foreach($jsons as $json){

    $ch = curl_init($update['url']);

    curl_setopt($ch, CURLOPT_HTTPHEADER, $data['headers']);

    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "PUT");

    curl_setopt($ch, CURLOPT_POSTFIELDS,$json);

    $response = curl_exec($ch);

    //var_dump($response);

    curl_close($ch);

}

}

unset($post);

unset($jsonsp);

unset($jsons);

unset($put);

unset($putp);

}

}

}

public function createOrder(){

    $json = array();

    if(isset($this->request->post['key'])){

```

```
$loginparam = [  
    'url' => 'https://www.suspus.com.tr/index.php?route=api/login',  
    'key' => $this->request->post['key']  
];  
  
$loginresult = json_decode($this->logincurl($loginparam));  
  
//var_dump($loginresult);  
  
if(isset($loginresult->success) && isset($loginresult->token)){  
    $customerparam = [  
        'url' =>  
        'https://www.suspus.com.tr/index.php?route=api/customer&token='.$loginresult->token,  
        'store_id' => '0',  
        'currency' => 'TRY',  
        'customer' => 'Trendyol Api',  
        'customer_id' => '2',  
        'customer_group_id' => '1',  
        'customer_group' => 'Default',  
        'custom_field[1]' => "",  
        'firstname' => 'Trendyol.',  
        'lastname' => 'Api',  
        'email' => 'gokmen.korkmaz@gmail.com',  
        'telephone' => '+90 (212) 331 32 50',  
        'fax' => "",  
        'custom_field[4]' => 'Trendyol',  
        'custom_field[2]' => 'Maslak',  
        'custom_field[3]' => '123456789'
```



```

];

$customerresult = json_decode($this->ordercurl($customerparam));

//var_dump($customerresult);

if(!isset($customerresult->success)){

$json['error'] = 'Hata:02 - Cari tanınmadı!';

$this->response->addHeader('Content-Type: application/json');

$this->response->setOutput(json_encode($json));

} else {

if(!isset($this->request->post['products'])){

$json['error'] = 'Hata:03 - Ürün bekleniyor!';

$this->response->addHeader('Content-Type: application/json');

$this->response->setOutput(json_encode($json));

} else {

$this->load->model('api/trendyol');

foreach($this->request->post['products'] as $product){

$product['gtin'] = $product['gtin'];

$product_info = $this->model_api_trendyol->getProductByGtin($product['gtin']);

//var_dump($product_info);

if($product_info){

$productparam = [

'url' =>

'https://www.suspus.com.tr/index.php?route=api/cart/add&token='.$loginresult->token,

'product_id' => $product_info['product_id'],

'quantity' => $product['qty']

];

}

}

}

}

];

```

```

//var_dump($productparam);

$productresult = json_decode($this->ordercurl($productparam));

//var_dump($productresult);

if(!isset($productresult->success)){

    $addtocart_errors[] = $product_info['product_id'] . "({$product['gtin']}) No'lu ürün
    sepete eklenemedi, ürün mevcut değil.";

}

} else {

    $addtocart_errors[] = $product['gtin'] . " barkodlu ürün sepete eklenemedi
    ({$product['qty']} adet), ürün mevcut değil.";

}

}

if(!isset($addtocart_errors)){

    $gettotalparams = [

        'url' =>
        'https://www.suspus.com.tr/index.php?route=api/cart/products&token='.$loginresult-
        >token

    ];

    //var_dump($gettotalparams);

    $gettotalresult = json_decode($this->ordercurl($gettotalparams));

    //var_dump($gettotalresult);

    $carttotal = str_replace("TL", "", $gettotalresult->totals[2]->text);

    $carttotal = str_replace(".", "", $carttotal);

    $carttotal = str_replace(",", ".", $carttotal);

    $carttotal = (float)$carttotal;

    //echo $carttotal;

    $addressparam = [

```

```

'url' =>
'https://www.suspus.com.tr/index.php?route=api/payment/address&token='.$loginresult->token,

'firstname' => $this->request->post['invoiceFirstName'],

'lastname' => $this->request->post['invoiceLastName'],

'company' => "",

'address_1' => strlen($this->request->post['invoiceAddress1']) > 127 ? substr($this->request->post['invoiceAddress1'],0,100) : $this->request->post['invoiceAddress1'],

'address_2' => strlen($this->request->post['invoiceAddress2']) > 127 ? substr($this->request->post['invoiceAddress2'],0,100) : $this->request->post['invoiceAddress2'],

'city' => $this->request->post['invoiceDisrict'],

'postcode' => "",

'country_id' => '215',

'zone_id' => $this->request->post['invoiceCityID']

];

//var_dump($addressparam);

$addressresult = json_decode($this->ordercurl($addressparam));

//var_dump($addressresult);

$addressparam = [

'url' =>
'https://www.suspus.com.tr/index.php?route=api/shipping/address&token='.$loginresult->token,

'firstname' => $this->request->post['shipmentFirstName'],

'lastname' => $this->request->post['shipmentLastName'],

'company' => "",

'address_1' => strlen($this->request->post['shipmentAddress1']) > 127 ? substr($this->request->post['shipmentAddress1'],0,100) : $this->request->post['shipmentAddress1'],

```

```

'address_2' => strlen($this->request->post['shipmentAddress2']) > 127 ? substr($this-
>request->post['shipmentAddress2'],0,100) : $this->request-
>post['shipmentAddress2'],

'city' => $this->request->post['shipmentDisrict'],

'postcode' => "",

'country_id' => '215',

'zone_id' => $this->request->post['shipmentCityID']

];

//var_dump($addressparam);

$addressresult = json_decode($this->ordercurl($addressparam));

//var_dump($addressresult);

$methodparam = [

'url' =>
'https://www.suspus.com.tr/index.php?route=api/payment/methods&token='.$loginre
sult->token

];

$methodresult = json_decode($this->ordercurl($methodparam));

//var_dump($methodresult);

$methodparam = [

'url' =>
'https://www.suspus.com.tr/index.php?route=api/payment/method&token='.$loginres
ult->token,

'payment_method' => 'bank_transfer'

];

$methodresult = json_decode($this->ordercurl($methodparam));

$methodparam = [

'url' =>
'https://www.suspus.com.tr/index.php?route=api/shipping/methods&token='.$loginre
sult->token

```

```

];

$smethodsresult = json_decode($this->ordercurl($smethodsparam));

//var_dump($smethodsresult);

$smethodparam = [

'url' =>
'https://www.suspus.com.tr/index.php?route=api/shipping/method&token='.$loginresult->token,

'shipping_method' => 'flat.flat'

];

$smethodresult = json_decode($this->ordercurl($smethodparam));

//var_dump($smethodresult);

$extra_comment = strlen($this->request->post['invoiceAddress1']) > 128 ? ' Fatura Adresi Ekstra: ' . $this->request->post['invoiceAddress1'] : '';

$extra_comment .= strlen($this->request->post['invoiceAddress2']) > 128 ? ' Fatura Adresi Ekstra: ' . $this->request->post['invoiceAddress2'] : '';

$extra_comment .= strlen($this->request->post['shipmentAddress1']) > 128 ? ' Kargo Adresi Ekstra: ' . $this->request->post['shipmentAddress1'] : '';

$extra_comment .= strlen($this->request->post['shipmentAddress2']) > 128 ? ' Kargo Adresi Ekstra: ' . $this->request->post['shipmentAddress2'] : '';

$addorderparam = [

'url' =>
'https://www.suspus.com.tr/index.php?route=api/order/add&token='.$loginresult->token,

'order_status_id' => '1',

'comment' => date('d.m.Y - H:i:s') . ' Tarihinde push api servisi ile gerçekleştirilmiştir.' . ' Trendyol Sipariş No : ' . $this->request->post['orderNumber'] . ' Müşteri Bilgisi: ' . $this->request->post['customerFirstName'] . ' ' . $this->request->post['customerLastName'] . ' ' . $extra_comment

];

$orderresult = json_decode($this->ordercurl($addorderparam));

```

```

//var_dump($orderresult);

if(isset($orderresult->success)){

$json = $orderresult;

$this->updateSites();

} else {

$json['error'][] = $orderresult->error;

$json['error'][] = $gettotalresult;

$data['log'] = json_encode($json);

$data['date'] = date('Y-m-d H:i:s');

$data['customer'] = 'Trendyol Api';

$jsondata = array(

"success" => false,

"order_id" => $result

);

//$json = $jsondata;

}

$this->response->addHeader('Content-Type: application/json');

$this->response->setOutput(json_encode($json));

}

}

}

}

}

}

}

```

```

public function updateSites(){
    $urls = array(
        /*'https://www.suspus.com.tr/index.php?route=api/modanisa/updateProductsStock',*/
        'https://www.suspus.com.tr/index.php?route=api/trendyol/updateProducts'
    );
    foreach($urls as $url){
        $data['url'] = $url;
        $this->curlget2($data);
    }
}

private function getBrandResult($brand){
    $data["url"] = "https://api.trendyol.com/sapigw//brands/by-name?name={$brand}";
    $data['headers'] = array(
        'Content-Type:application/json',
        'Authorization: Basic '. base64_encode("{$this->apiusername}:{$this->apiuserpassword}")
    );
    $result = $this->curlget($data);
    return $result;
}

private function serializeCategory($data = array()){
    $this->load->model('api/trendyol');
    foreach($data as $category){
        $results = array(

```

```

'id' => $category['id'],
'name' => $category['name'],
'parentId' => $category['parentId'] == NULL ? 0:$category['parentId']
);
$this->model_api_trendyol->setTrendyolCategories($results);
$this->serializeCategory($category['subCategories']);
}
}

```

```

private function getCategoryTree(){
$data["url"] = "https://api.trendyol.com/sapigw/product-categories";
$data["headers"] = array(
'Content-Type:application/json',
'Authorization: Basic '. base64_encode("{ $this->apiusername }:{ $this->apiuserpassword }")
);
$result = json_decode($this->curlget($data), true);
//$this->response->addHeader('Content-Type: application/json');
//$this->response->setOutput($result);
return $result['categories'];
}

```

```

private function getCategoryAttributes($categoryId){
$data["url"] = "https://api.trendyol.com/sapigw/product-categories/1241/attributes";
$data["headers"] = array(

```



```

'Content-Type:application/json',

'Authorization: Basic '. base64_encode("{ $this->apiusername }:{ $this-
>apiuserpassword}")

);

$result = $this->curlget($data);

return $result;

}

```

```

private function curlget($data = array()){

$ch = curl_init($data['url']);

curl_setopt($ch, CURLOPT_HTTPHEADER, $data['headers']);

curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

curl_setopt($ch, CURLOPT_HTTPGET, true);

$curlresult = curl_exec($ch);

//$curlresult2 = curl_getinfo($ch);

curl_close($ch);

return $curlresult;

}

```

```

private function curlget2($data = array()){

$ch = curl_init($data['url']);

curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

curl_setopt($ch, CURLOPT_HTTPGET, true);

$curlresult = curl_exec($ch);

//$curlresult2 = curl_getinfo($ch);

curl_close($ch);

```

```

return $curlresult;
}

private function logincurl($data = array()){
$cf = dirname(__FILE__) . '/cookietrd.txt';
$url = $data['url'];
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_COOKIEJAR, $cf);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
$curlresult = curl_exec($ch);
curl_close($ch);
return $curlresult;
}

private function ordercurl($data = array()){
$cf = dirname(__FILE__) . '/cookietrd.txt';
$ch = curl_init($data['url']);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_COOKIEFILE, $cf);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
$curlresult = curl_exec($ch);

```

```
curl_close($ch);  
return $curlresult;  
}
```

```
private function curlput($data = array()){  
    $ch = curl_init($data['url']);  
    curl_setopt($ch, CURLOPT_HTTPHEADER, $data['headers']);  
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "PUT");  
    curl_setopt($ch, CURLOPT_POSTFIELDS, $data['json']);  
    curl_setopt($ch, CURLOPT_TIMEOUT, 5);  
    $curlresult = curl_exec($ch);  
    curl_close($ch);  
    return $curlresult;  
}
```

```
private function curl($data = array()){  
    $ch = curl_init($data['url']);  
    curl_setopt($ch, CURLOPT_HTTPHEADER, $data['headers']);  
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
    curl_setopt($ch, CURLOPT_POST, true);  
    curl_setopt($ch, CURLOPT_POSTFIELDS, $data['json']);  
    curl_setopt($ch, CURLOPT_TIMEOUT, 5);  
    $curlresult = curl_exec($ch);  
    curl_close($ch);
```

```
return $curlresult;
}
}
```

## Model

```
<?php
```

```
class ModelApiTrendyol extends Model {
```

```
public function setTrendyolCategories($data = array()){
```

```
$sql = "INSERT INTO " . DB_PREFIX . "trendyol_categories(id, name, parent_id)
VALUES({$data['id']}, '{$data['name']}', {$data['parentId']})";
```

```
$this->db->query($sql);
```

```
}
```

```
public function setShipmentProviders($data = array()){
```

```
$sql = "INSERT INTO " . DB_PREFIX . "trendyol_shipment(id, name, code,
tax_number) VALUES({$data['id']}, '{$data['name']}', '{$data['code']}',
'{$data['taxNumber']}')";
```

```
$this->db->query($sql);
```

```
}
```

```
public function delTrendyolCategories(){
```

```
$sql = "TRUNCATE TABLE " . DB_PREFIX . "trendyol_categories";
```

```
$this->db->query($sql);
```

```
}
```

```
public function delShipmentProviders(){
```

```
$sql = "TRUNCATE TABLE " . DB_PREFIX . "trendyol_shipment";
```

```

$this->db->query($sql);

}

public function getProductByGtin($gtin){

$sql = "SELECT * FROM " . DB_PREFIX . "product p WHERE p.ean = " . $gtin .
""";

$query = $this->db->query($sql);

return $query->row;

}

public function getUpdateProducts(){

$sql = "SELECT p.* FROM " . DB_PREFIX . "product p JOIN " . DB_PREFIX
."product_changes pc ON p.product_id=pc.product_id WHERE pc.status2=0";

$query = $this->db->query($sql);

return $query->rows;

}

public function updateProductChanges(){

$query = $this->db->query("UPDATE " . DB_PREFIX ."product_changes SET
status2=1 WHERE status2=0");

}

}

```